Using Controlled Query Generation To Evaluate Blind Relevance Feedback Algorithms

Chris Jordan
Dalhousie University
6050 University Ave.
Halifax, NS, Canada
cjordan@cs.dal.ca

Carolyn Watters
Dalhousie University
6050 University Ave.
Halifax, NS, Canada
watters@cs.dal.ca

Qigang Gao
Dalhousie University
6050 University Ave.
Halifax, NS, Canada
qggao@cs.dal.ca

ABSTRACT
Currently in document retrieval there are many algorithms each with different strengths and weakness. There is some difficulty, however, in evaluating the impact of the test query set on retrieval results. The traditional evaluation process, the Cranfield evaluation paradigm, which uses a corpus and a set of user queries, focuses on making the queries as realistic as possible. Unfortunately such query sets lack the fine grained control necessary to test algorithm properties. We present an approach called Controlled Query Generation (CQG) that creates query sets from documents in the corpus in a way that regulates the theoretic information quality of each query. This allows us to generate reproducible and well defined sets of queries of varying length and term specificity. Imposing this level of control over the query sets used for testing retrieval algorithms enables the rigorous simulation of different query environments to identify specific algorithm properties before introducing user queries. In this work, we demonstrate the usefulness of CQG by generating three different query environments to investigate characteristics of two blind relevance feedback approaches.

Categories and Subject Descriptors
H.3 [Information Storage and Retrieval]: Systems and Software—Performance evaluation (efficiency and effectiveness); H.3 [Information Storage and Retrieval]: Information Search and Retrieval—Query formulation, Relevance feedback, Search process

General Terms
Algorithms, Experimentation, Performance, Theory

Keywords
Controlled Query Generation, Algorithm Evaluation, Control Environment, Blind Relevance Feedback

1. INTRODUCTION
There are many algorithms available for document retrieval ranging from the classical retrieval algorithms [1] of Boolean, vector space, and probabilistic modeling to the more recent language modeling [20] approaches. While the mechanics behind each algorithm are well understood, less is understood about the effect of query selection on their retrieval performance. Currently there is no clear topology performing retrieval algorithm across the range of query types; some algorithms perform better on some types of queries than others. Understanding the relationships between query properties and algorithm performance can help guide the research and development of new and improved retrieval algorithms.

A typical approach to evaluating retrieval algorithms follows what is known as the Cranfield paradigm [24]. The performance of an algorithm is measured using a set of queries over a collection of documents or corpus. The relevant set of documents (qrels) for a query is used to calculate retrieval performance scores such as precision and recall. Popular test collections include TREC [22] and CACM [8] which have predefined query sets and qrels created by users or evaluators. While these evaluations reveal to what degree an algorithm satisfies those queries, each query has unique characteristics and the composition of the corresponding qrels is not necessarily reproducible. As such, it is difficult to match specific algorithm properties with specific query properties. Overall, there is a lack of control and consistency in queries based on user requests with respect to the characteristics of the query and the definition of document relevancy.

In this paper we propose a method for evaluating retrieval algorithms that provides for greater control over the query set. We introduce this additional control using a synthetic query generation method, Controlled Query Generation (CQG), to generate sets of queries from source document sets which then also doubles as the corresponding qrels. A query environment is a collection of synthetic queries generated in the same manner over different source document sets in a corpus.

CQG creates queries that are reproducible and theoretically consistent and, more importantly, has the capability to generate different query environments for the same corpus. By carefully controlling the process in which the queries are generated from the source documents, query environments with very specific characteristics can be developed. For example, longer vs shorter queries and queries with highly discriminating terms vs queries with less discriminating terms. Exposing retrieval algorithms to such query environments
allows us to study algorithm properties in a rigorous fashion not possible with user defined queries. While the traditional style of evaluation will determine if users would be satisfied with the results of a retrieval algorithm, an evaluation using CQG will evaluate how an algorithm reacts to query characteristics. These are complementary evaluations.

In this paper, we will describe the theoretical basis for the CQG method and demonstrate its use by evaluating two blind relevance feedback (BRF) approaches, one based on the vector space model and the other on language modeling, over three query environments, defined by the relative entropy between the source document sets and the corpus. The first query environment is composed of single term queries with varying relative entropy contributions. The second environment contains two term queries where one of the query terms is the largest relative entropy contributor. The third environment is composed of multi term queries where the query terms contribute the most to relative entropy. Exposing algorithms to these environments will test how they will react to queries of varying length and term quality.

2. BACKGROUND

The traditional approaches to evaluating document retrieval algorithms includes a corpus, a collection of queries, and corresponding query result evaluations, often called query relevant sets (qrels). Algorithms are evaluated on how well they retrieve the qrels from the corpus for each query. Two general evaluation models dominate: controlled query sets and user query sets. One of the first controlled evaluations was the Cranfield 1 test [13] using a corpus of 18,000 aeronautical research documents. Queries were derived from source documents in the corpus and the qrel for each query was then determined by evaluators assessing each document in the corpus manually.

Many retrieval evaluations have been conducted by using user query sets or in other words allowing users to interact with the retrieval systems and providing the relevance judgments themselves. The problem with user evaluations is that it is very difficult to then compare different algorithmic approaches. As well, there are costs involved in running user evaluations and endemic difficulties in replicating experimental results due to the natural behavior of users. The Cranfield paradigm imposes control by removing the user from the evaluation but does not impose control over the characteristics of the queries and does not scale well.

The use of source documents to create synthetic test queries as in the Cranfield 1 test has drawbacks [21]. Queries derived from source documents do not necessarily represent actual user queries and can be expected to produce more favorable retrieval results. The Cranfield paradigm though does allow evaluations of different algorithms to be compared directly with each other within a specific set of constraints. Other test collections such as TREC [22] and CACM [8] have been developed following this paradigm.

The TREC test sets presents an additional challenge to retrieval systems in having to deal with very large corpora that contain hundreds of thousands of documents. With such large corpora, the qrels cannot be created by assessing every document. A different technique called pooling [24] is used. In pooling, the retrieval results from multiple systems for a given query are collected together and judged for relevancy. A drawback with pooling is that it does not determine the complete membership for a qrel. This issue is only problematic if there are few retrieved documents. Voorhees [24] also explored the issue of how the definition of relevance differs between human assessors and found that while relevance judgments do vary significantly, they seem to have little effect on how different retrieval systems compare to each other.

While these types of evaluations are useful for comparing systems to each other in a controlled environment, they also have a significant shortcoming in their lack of control over the queries. When evaluating any system, not just retrieval systems, it is important to have complete control over the system inputs in order to perform incremental adjustments to them for experimentation that will specifically test system properties. For a retrieval system, queries are the main inputs and typically little control or reproducibility is included in the selection of the query set. Consequently, many evaluations indicate which retrieval algorithm performs better for the given query set but this may not be generalizable and does not indicate strengths and weaknesses vis a vis a viz query structure.

As a result of these traditional approaches to evaluation, there has been very little work done in analyzing the impact of query structure and composition on retrieval. Analysis of search engine query logs has revealed that users generally issues short queries consisting of 2 to 3 terms [19]. Empirical studies have shown though that longer queries outperform shorter queries [2, 11]. Kelly et al [11] also found that queries pertaining to the user’s motivation and background knowledge about the search topic improve performance. While these findings do give insight as to what ideal queries should look like, their composition, properties, and impact on retrieval are still not well understood.

In this work, we use a method, Controlled Query Generation (CQG), which uses source document sets from the corpus to engineer queries with specific characteristics. CQG introduces incremental control over the queries in the query set by deriving them from these source documents rather than relying on users to author them. By using this technique it is possible to conduct retrieval algorithm evaluations that specifically identify the impact of different query characteristics.

2.1 Language Modeling and Relative Entropy

To generate queries in CQG for this work we used relative entropy [5] of the language models [20] for the source document sets and the whole corpus. Language modeling is an approach for calculating a probability mass function based on a sample text, such as a document. A basic approach for estimating a language model is to use the maximum likelihood estimate (MLE), which generates a probability mass function for a document using the ratios of its term occurrences to total term occurrences as shown below in Equation 1.

$$P_{\text{ml}}(t | d_k) = \frac{\text{freq}_{t,d_k}}{\sum \text{freq}_{d_k}} \quad (1)$$

where $t$ is a term in the corpus vocabulary, $d_k$ is a document in the corpus, $\text{freq}_{t,d_k}$ is the frequency with which $t$ occurs in $d_k$, and $\sum \text{freq}_{d_k}$ is the sum of all the term frequencies in $d_k$. MLE assigns zero probabilities to terms that do not appear in the sample text, which is extreme since the absence of a term in a document does not necessarily mean that the document model could never generate it. Consequently, smoothing techniques are generally used to remove
been previous applications of relative entropy to information retrieval and ease of implementation.<Document>

\[ P_{jm}(t|d_k) = (1 - \lambda) P_{ml}(t|d_k) + \lambda P_{ml}(t|\text{corpus}) \quad (2) \]

where \( \lambda \) is a constant between 0 and 1 that regulates how much \( P_{ml}(t|d_k) \) and \( P_{ml}(t|\text{corpus}) \) contribute to the construction of the model. For this work \( \lambda \) is set to be 0.4 to be consistent with previous work done by Croft [6, 14]. \( P_{ml}(t|\text{corpus}) \) is the model for the corpus. This model is estimated with the MLE using all the documents in the corpus concatenated together. As all documents are used in the estimation of this model, no terms are assigned a zero probability and further smoothing is not required.

Relative entropy is a method for comparing two probability mass functions. While not a true distance function, relative entropy can be used for comparing language models, as shown below in Equation 3.

\[ D(P_{d_k}||P_{d_j}) = \sum_{t \in V} P(t|d_k) \log \frac{P(t|d_k)}{P(t|d_j)} \quad (3) \]

where \( P_{d_k} \) and \( P_{d_j} \) are language models for documents \( d_k \) and \( d_j \) respectively. Relative entropy is calculated iteratively over all the terms in the vocabulary, \( V \), used by the corpus.

Language modeling has been used for document retrieval by treating a query as a random event that can be generated by a document model. Documents are ranked based on the likelihood of their model generating the query. There have been previous applications of relative entropy to information retrieval. Specifically, Cronen-Townsend et al used relative entropy between the top ranked documents and the corpus to measure query performance [6]. Zhai and Laflerty applied relative entropy to both document retrieval [12] and relevance feedback [28]. Cai et al developed scoring methods based on relative entropy and other divergence functions to identify terms in the top ranked documents to use for query expansion [4]. The use of relative entropy in this paper is somewhat similar to Cai’s work but differs in that CQG uses source document sets to compare with the corpus rather than the top ranked in retrieved sets.

### 2.2 Relevance Feedback Example

To demonstrate the use of CQG in algorithm evaluation, an evaluation of two blind relevance feedback (BRF) algorithms is conducted. BRF is a technique for improving the query using the top ranked documents from the initial search; this technique works on the assumption that these top ranked documents are relevant to the user’s search. In relevance feedback [17] systems, the user first issues a query and then provides feedback, typically as relevance judgments, on the retrieved set. This feedback is then incorporated into the query to form a new search. Users, however, are reluctant to give feedback during search sessions. Automatic methods for gathering feedback, referred to as implicit feedback, attempt to overcome this lack of user interaction though have only met with marginal success. BRF algorithms, however, exclude user from the feedback process entirely and assume that at least some of the top ranked documents are relevant and can be used to improve the query. Clearly, this assumption does not always hold and noise in the top ranked documents may lead to query drift [15].

Much of the initial work on BRF can be traced back to Salton [3] and TREC [7]. Since then, there has been much work done in exploring how to improve its application. This work includes experimentation with the number of top ranked documents to use [16], the number of terms to extract from those documents [9], and the locality of the terms to extract with respect to the query terms [9, 26]. Results indicate that the performance of BRF algorithms depends on the type of query that has been issued. Sakai attempted to find the optimal BRF settings for classes of queries which led to his work on optimization tables [18].

There has been a lot of work exploring how to improve BRF though it is not completely understood how different query characteristics affect retrieval performance and we believe this is due to the lack of control over the queries in the Cranfield style evaluations that were conducted previously. In this paper we perform evaluations using CQG to show that BRF behaves differently in different query environments. The lack of control in traditional evaluation practices has prevented the identification of query environments that BRF can be expected to provide optimal improvements. CQG allows an algorithm evaluation to be constrained to specific query environments and in using this style of evaluation we have successfully identified specific query conditions in which to use BRF.

### 3. CONTROLLED QUERY GENERATION

Controlled Query Generation (CQG) generates a set of queries based on a source document set. These queries are synthetic in nature and are not necessarily representative of real user queries. The purpose of CQG however is to manufacture queries that will evaluate the impact of specific query characteristics on algorithm performance. To initiate this process, sets of related documents in the corpus must be defined. CQG generates sets of queries based on and targeting each of these sets of related source documents. It identifies the most discriminating terms in each source document set using relative entropy. These terms are then used to manufacture a set of queries, called a query environment, targeting these source documents. The advantage that CQG has over user defined queries is that different query environments can be simulated by altering how the queries are manufactured. Queries are also created iteratively in a manner such that their information content and structure is strictly controlled. This type of query manipulation with incremental adjustments is very difficult to do effectively with user queries.

One of the challenges in using CQG is determining the methods for generating queries for testing. In this work we present three different variations all based on relative entropy of terms in the corpus. The first treats single terms as queries using the relative entropy for each term in the vocabulary of the source document set to select queries with varying discriminative power. While single term queries do not contain a lot of information to describe the information need, it is not uncommon for users to issue them [19].

The second CQG variation generates two term queries. It combines the term that contributes the most to the relative entropy between the source document set and the corpus with other terms in the corpus vocabulary. This query environment is representative of queries where one term is highly discriminating while the other one varies in quality. This method introduces noise into queries in a controlled man-
The TMD set, the next query is this term concatenated to the previously generated query with a term frequency equal to
its raw average from the source documents.

Step 7: Repeat Steps 5–6 until there are no more terms.

Figure 1 illustrates the steps in CQG. This process can create a set of queries for each source document set by exhaustively iterating through the corpus vocabulary. In this work, the cycle is halted when only terms that contribute less than 1% to relative entropy between the source documents and the corpus remain to be selected in step 5. This threshold was determined through experimentation, which found that terms that contribute less than 1% to relative entropy have a negligible impact on query performance.

4. A CQG EXAMPLE EVALUATION

The Reuters~21578 corpus [23] was used to evaluate CQG as all its documents have been manually categorized for a variety of schemes; every document has been evaluated by a human assessor for membership into each categorization scheme. The TOPICS and PLACES scheme in particular can be used to identify sets of related documents to use as source document sets. The TOPICS scheme labels documents based on economic themes and the PLACES scheme labels them based on geographic themes. In this work, a source document set contains documents that have a TOPICS label and PLACES label in common. For example, all documents with TOPICS = “grain” and PLACES = “canada” are considered members of the same source document set. Table 1 shows the distribution of the related document set sizes; there are more smaller sets than larger ones. There were two document sets that were much larger than the rest, “acq usa” and “earn usa” which contained 1787 and 3143 documents respectively, that were considered to be outliers and not used during the evaluation. Furthermore, some documents did not have a TOPICS or PLACES label and these were not used during the evaluation here. As such only 19,812 documents in Reuters~21578 were used. Using the TOPICS and PLACES labels, a total of 1,882 source document sets were defined on the test corpus. A standard index of keywords was created after standard stop words were removed and terms were stemmed using Porters algorithm [1].

This approach for creating these sets of related documents
has been used before in previous work [10]. While it is artificial in nature, it is sufficient as it does identify documents that are related categorically. It is recognized that some sets generated in this manner may be topically very broad. Using subject matter experts to create these related sets similar to the work in TREC [22] would be ideal as it would help ensure that all documents in a set are strongly related to the work in TREC. An ideal approach would be to use the work in TREC to create these related sets and if every document in the corpus is to be evaluated in order to assess complete membership for each one. In using the categorical approach on Reuters-21578, complete membership to the related document sets is known and we avoid the cost of having to use assessors to define them.

Queries generated by CQG were used to evaluate 3 BRF retrieval algorithms (Blind Relevance Feedback Rocchio using 5 documents (ROC5), Blind Relevance Feedback Rocchio using 10 documents (ROC10), Relevance Modeling (RM)) and two baseline retrieval models (Vector Space Model (VSM) and Song’s Language Modeling (LM)). The two BRF approaches, Rocchio and Relevance modeling, are used to re-rank the retrieved set for the original query. We have assumed that relevant documents would contain at least one of the original query terms. This assumption does not always hold, of course, especially with short queries.

In performing the statistical analysis, we looked at the retrieval performances for queries generated at beginning, middle, and end of the CQG process. Queries generated at these different periods have different compositions. As such, we have devised three query groups, first, middle, and bottom, to group queries from these respective points of generation during CQG. We also conduct the same statistical analysis over the entire set of queries generated by CQG.

The VSM does not use feedback and is a baseline for the two Rocchio BRF algorithms. The ROC5 and ROC10 are Rocchio relevance feedback algorithms [17] using the top 5 and top 10 documents as positive feedback respectively. Song’s approach to language modeling [20] is used as the baseline for relevance modeling. In this work we use the Jelinek–Mercer method for smoothing instead of the Good Turing approach used by Song. Relevance modeling [14] is a BRF approach to language modeling that uses the top ranked documents to construct a probabilistic model for performing the second retrieval. The contribution that each of the top ranked documents makes to this model is directly related to their retrieval score for the initial query. To be consistent with Lavenko's work, the top 50 documents are employed in the construction of the relevance models.

## 5. RESULTS

We observed that retrieval performance tended to be better on smaller target sets than on larger ones. One reason we believe to be the cause of this result is the overlap of vocabulary between documents in a set. It does not necessarily follow that all documents in the same set share a large common vocabulary. With a large set, the vocabulary used in each document may vary greatly thus decreasing the discriminative power of the terms used to distinguish it in the corpus. For evaluation purposes, sets are grouped together by sizes 1 to 10, 11 to 15, 16 to 20, 21 to 50, and 51 to 400. It is natural for the size of the relevant sets to vary in a corpus as not all topics will have equal coverage. Smaller relevant sets are representative of topics that are not well covered by the corpus.

Both Mean Average Precision (MAP) and R–Precision were measured during these trials. MAP provides a measure of how precise the top ranked documents are while R–Precision reports recall as a percentage of the top R documents, where R is the number of relevant documents for a given query.

For these trials, the algorithms are evaluated on the three different query environments discussed previously. For each environment, all possible queries are generated by CQG. Of these queries, 100 are randomly selected from each quadrant. As well, 100 queries are randomly selected from the entire query set. Hence, there are four sets of 100 queries from different points in the CQG process that the algorithms are evaluated on for each environment. The results from trials over the CQG queries are shown in Figure 2 for MAP and Figure 3 for R–Precision. All trials are analyzed using a randomized complete block design ANOVA and a Tukey HSD [25] to determine any significant differences between the algorithms. For each TMD query environment, at least one algorithm was significantly different ($p < 0.0001$). For many of the single term query trials over the small and medium sets, the algorithms were not significantly different. For the large sets though, at least one of the algorithms is significantly different ($p < 0.05$). In the two term query environment, the ANOVAs revealed that at least one algorithm is significant different for each trial ($p < 0.0001$).

### 5.1 Significant Differences in MAP

With regards to MAP, we found that for the single term queries, relevance modeling is significantly better than language modeling over the large source document sets except for queries selected from the bottom quadrant; queries from the bottom quadrant are possibly too poor in quality for feedback to be able to improve performance on them. The Rocchio BRF algorithms were only significantly better than VSM for a few trials over the large document sets. For the two term queries we found that relevance modeling is significantly better than language modeling for medium and large sets except for queries selected from the first quadrant, possibly a result of queries from the first quadrant containing

<table>
<thead>
<tr>
<th>Size</th>
<th>Documents per Set</th>
<th>Number of Source Document Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Too</td>
<td>1</td>
<td>831</td>
</tr>
<tr>
<td>Small</td>
<td>2</td>
<td>346</td>
</tr>
<tr>
<td>Very</td>
<td>4</td>
<td>85</td>
</tr>
<tr>
<td>Small</td>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td>Medium</td>
<td>11 to 15</td>
<td>75</td>
</tr>
<tr>
<td>Large</td>
<td>21 to 50</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>51 to 400</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 1: Distribution of source document set sizes.
Figure 2: MAP performance over the three query environments
Figure 3: R-Prec performance over the three query environments
two highly discriminative terms. On the TMD queries we find the opposite in that language modeling is significantly better than relevance modeling for all sizes. VSM is only significantly better than the Rocchio BRF algorithms for small sets.

### 5.2 Significant Differences in R-Precision

We found the results to be similar for R-Precision. For single term queries, we found that relevance modeling is significantly better than language modeling over large sets for queries from the first and middle quadrants. Relevance modeling is significantly better than language modeling over the two term queries for medium and large sets except for queries from the first quadrant. Language modeling however is significantly better for the TMD queries except for small sets. Rocchio BRF is significantly better than VSM for a few of the single query term trials over large document sets. VSM is significantly better than both of the Rocchio BRF algorithms in the TMD query environment for queries selected from the bottom quadrant over all document set sizes and over small and medium sizes for queries selected from the middle quadrant.

### 5.3 Discussion

The results from these trials are not entirely surprising. BRF has received a lot of attention as a simple method for improving the user query. These trials indicate that relevance modeling improves performance for short queries but damages it for the TMD queries. This result makes sense as relevance modeling expands the query based on the top ranked documents. The single and two term queries are apparently capable of retrieving at least a few relevant documents in the top ranks which relevance modeling then builds upon. Single term queries from the bottom quadrant though seem to be too poor to improve using feedback and two term queries from the first quadrant appear to have enough discriminative power negate its positive effects. Conversely, the TMD queries create an environment of very comprehensive queries. These queries are difficult to improve upon and consequently relevance modeling fails to provide significant improvements.

It is interesting that the VSM and Rocchio BRF algorithms do not differ significantly for most trials. We attribute this result to the queries being formed from terms selected by their relative entropy contribution. In other words, the query terms that are used during the CQG process here had the greatest probability in the source document models and the lowest in the corpus model. Such query terms are naturally favored by probabilistic approaches to retrieval. Perhaps if the terms were selected for query generation using a term frequency scoring scheme there would be a greater difference in performance. Exploration of alternative term scoring functions is left to future work.

From these results, it appears that a good heuristic would be to apply BRF techniques whenever a single term query is issued. Even though the difference in performance was not always significant, there is a chance that BRF will improve performance. Single term queries describe information needs very poorly. BRF however is to able to improve these queries given that they had sufficient discriminative power and at least one relevant documents in the top rank.

As for two term queries, it appears that the most discriminating term for a relevant set is significant enough to nullify the positive effects of BRF should the second term be drawn from the first quadrant. In other words, if the second term also has a high discriminative power then this two term query will not need BRF. Terms of less discriminative power however will benefit from the application of BRF. An analysis of typical user queries needs to be performed to determine the discriminative power of terms users generally select for a query. If this analysis reveals that when users issue two term queries that only one of the terms has significant discriminative power then BRF should be prescribed for these types of queries as well.

### 6. CONCLUSION

The Cranfield paradigm of creating test queries from documents in a corpus is a proven approach to evaluating retrieval algorithms however it is limited in that it can only indicated to what extent information needs were met by relying on assessments of relevance. Even though the queries used may be very representative of actual user queries, there is little control over specific query properties which may affect an algorithms performance. To this end, we introduce an approach to query generation called Controlled Query Generation (CQG) that provides greater control thus allowing for query environments with specific properties to be created.

The query sets created in CQG target predefined sets of related source documents. While these queries and source documents are artificial in nature, they afford direct control over the amount of information a query contains. CQG provides a means to generate controlled query environments for use in comparisons between different retrieval algorithms as a complement to real user queries. Through CQG evaluations, it is possible to identify the strengths and weaknesses of an algorithm with respect to the effects of query term quality and query length.

In this work, single term, two term, and Theoretically Most Discriminating (TMD) query environments were created and two BRF approaches were evaluated. The results indicated that relevance modeling is significantly better than language modeling on short queries while it is significantly worse on the TMD queries. The performance of Rocchio BRF approaches and VSM however, did not differ greatly in these environments. The results indicate that for single term queries BRF should be used. For the two term queries tested here, applying BRF does provide an advantage.

Language models for the relevant sets were used here as the basis for the generation of queries. All documents in a source document set are treated as being equally important in the construction of the model. Typically all documents are not equally relevant. Sometimes only part of the document, perhaps a few paragraphs, are relevant to the users query. An advantage of the CQG method is that it can use alternative theoretical approaches to query term selection, such as maximal gain or chi squared. CQG increases the level of control and consistency of the query set used for the evaluation process.

In this sample provided here, a general analysis was conducted where the query sets as a whole were used in the calculation of summarization performance metrics, MAP and R-Precision. Other types of analysis can be conducted on these results such as specifically examining the queries in each set where an algorithm fails to improve performance. More extensive analysis of these various situations may lead...
to a more refined understanding of how BRF based algorithms react to the natural language in a corpus. It is left to future work to explore different means of mining these results from CQG evaluations.

One weakness of CQG is that it requires access to a corpus with consistently definable sets of related documents. Manually defining such sets on a corpus requires a large effort and is influenced by the evaluators. Corpora, like Reuters-21578, that are rigorously categorized are unfortunately rare. The TREC corpora have qrels defined on them though not every document has been evaluated for membership. Even so, these qrels can potentially be used as source document sets and will be the subject of future experimentation using CQG.

CQG evaluations are not intended to replace the need for user evaluations but rather to complement them. It is quite obvious that the queries generated by CQG are often times not representative of actual user queries. These synthetic queries however, can be manufactured in different ways to embody different query properties and consequently are a relatively efficient way to assess the strengths and weaknesses of different retrieval algorithms. CQG evaluations are relatively inexpensive to conduct and should be performed before the more costly user trials. In this manner, algorithms can be modified and optimized for different query environments in a controlled manner. User evaluations then can be used to validate their retrieval effectiveness.

One of the future challenges for CQG is in developing new methods to generate queries. Simply concatenating terms together will not result in a query environment that is optimal for the evaluation of particular characteristics for an algorithm. Much thought must be given to how to control the manufacturing of the queries as this process will ultimately determine the quality of the resulting query environment. By creating a query environment that isolates specific query properties, the impact of these respective properties on retrieval algorithms can be evaluated. Knowing how document retrieval algorithms are effected by different query properties may enable better algorithms and potentially methods for improving users queries to be developed.

7. ACKNOWLEDGMENTS

This research was funded by the Natural Sciences and Engineering Research Council of Canada, NSERC.

8. REFERENCES


