Simple Questions for Interactive Information Retrieval

Giridhar Kumaran and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
140 Governors Drive
Amherst, MA 01003, USA
{giridhar,allan}@cs.umass.edu

ABSTRACT

We explore simple questions that can be used for interactive information retrieval. We develop four techniques that have the potential to be used in an interactive setting. The techniques are designed to be easy to use and understand, and provide good improvements in performance with minimal effort from the user. We test the automatic versions of the techniques in two environments known to be difficult, and report significant improvements in performance as measured by MAP and GMAP over pseudo-relevance feedback. Our successful testing of one of the techniques in an interactive setting encourages the pursuit of more similar techniques to improve information retrieval with a new approach.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**–Query formulation,Search process

General Terms: Performance, Experimentation

Keywords: User interaction, feedback, information retrieval

1. INTRODUCTION

Information retrieval (IR) systems play a crucial role as an intermediary between an user and information. They need to provide a convenient interface to convey information needs, transform those needs to a suitable query, and return relevant information from the corpus. However, given variables such as the way users express information needs, internal representation of the data in the index, number of relevant documents in the target corpus and so on the role of the information retrieval system becomes more complex. Since there are obvious limits to what can be achieved by the IR system automatically, it is widely acknowledged that certain types of queries requires some form of IR system-user interaction. This naturally brings up more issues ranging from the best design for an user interface, how to use the information obtained and how to involve the user.

Recent empirical studies [8] suggest that even user interaction can achieve only a limited (though significant) improvement. In this work we explore what we believe will contribute to a minimal set of interaction strategies to achieve that improvement. We address the question of what to ask the user to improve retrieval results. We are guided by the belief that such *information-gathering* strategies should be light-weight, i.e. we view user participation as a resource that must be judiciously used. In such a scenario strategies such as presenting a user with a list of documents and asking her to mark entire documents, passages or terms as relevant/non-relevant is taboo. In other words, the goal is to identify a set of simple questions that can be quickly and (possibly) effortlessly answered by the user.

In Section 2 we will start by motivating and presenting a set of simple automatic techniques and evaluate them in Section 4. The techniques are designed such that they can be seamlessly adapted into interactive versions. We will show in Section 5 the effect on performance when the user is brought into the loop in one technique.

2. SIMPLE TECHNIQUES

Understanding the reasons why current IR systems fail [4] is key to determining what additional information we need to seek from the user. These reasons could be technical failures like stemming, wrong emphasis on certain query terms, emphasis on only certain aspects or just vocabulary mismatch between queries and documents. The techniques we present are not targeted at any one particular problem. We will show that they mitigate appreciably the problems associated with some of the reasons mentioned. The techniques also take advantage of structured query languages. Structured query languages offered a more powerful way to express queries owing to the fact that additional information about the information need can be easily incorporated. This inclusion is made possible by the use of phrases, synonyms, date ranges, term absence indication and so on.

We explore four techniques to improve retrieval performance. For each, we illustrate with example queries that allow application of the technique, and describe how we address the associated problem.

2.1 Edit Distance

2.1.1 The problem

Topic 356: Identify documents discussing the use of *estrogen* by postmenopausal women in Britain.

Failure analysis of this low-scoring query (Section 3) re-

vealed that in a number of relevant documents, *estrogen* was spelt as *oestrogen*.

Topic 356: How many deaths are attributed to having taken tainted L-*tryptophan* dietary supplements?

A similar analysis revealed that some relevant documents spelt *tryptophan* as *tryptophane*. The stemmer failed to conflate these two terms to the same stem. Thus, differences in spelling between the query and the documents in a collection are an obvious plausible reason for retrieval to fail. In addition to spelling variations like those mentioned, another problem observed was cultural differences such as those between Britain and the United States. An example of this is *legalization* and *legalisation*.

2.1.2 Implementation

To address this problem, we focus on string edit distance as a simple type of spelling correction. The edit distance [11] between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. Each word that has alternate spellings is replaced with a synonym operator combining all synonyms and the query is run. For example, `#syn( estrogen oestrogen )`. For the automatic experiments in this paper, we consider any word that has an edit distance of one as an alternate spelling. Because this is a quite simple question for a person to answer, we anticipate including larger edit distances as possibilities when formulating clarifying questions. Larger edit distances are likely to result in some very unlikely matches (e.g., Britax for Britain), so before presenting options to a user, we will need to incorporate additional heuristics or learning approaches to reduce the candidates to a set that is plausible.

Simple questions

Is oestrogen a reasonable variant spelling of estrogen?
Is taxes a way that someone might spell faxes?

Parallels could be drawn to spelling-correction features available in most web-search engines. The main difference in our approach is that we find alternate spellings for a term instead of correcting it, and the alternate spellings are identified from the corpus itself. We refer to this technique as **Edit**.

2.2 Identifying phrases

2.2.1 The problem

Topic 320: Fiber optic link around the globe (Flag) will be the *world's longest* undersea fiber optic cable. Who's involved and how extensive is the technology on this system. What problems exist?

Topic 339: What drugs are being used in the treatment of *Alzheimer's Disease* and how successful are they?

In the two examples above the apostrophe is used to form possessives of nouns. In such situations we can expect the two terms to occur as a phrase in the relevant documents too.

Topic 344: What steps have been taken *world-wide* by those bearing the cost of *E-mail* to prevent excesses?

Topic 443: What is the extent of U.S. (government and private) investment in *sub-Saharan* Africa?

In the next two examples hyphens are used in different contexts to indicate compound words. Again, we can expect to see these terms appear either as phrases or mentioned as one word in relevant documents.

Topic 443: Find documents that discuss issues associated with *so-called "orphan drugs"*, that is, drugs that treat diseases affecting relatively few people.

The use of double quotes also implies that the enclosed terms form a phrase.

2.2.2 Implementation

In most IR systems punctuation is discarded while parsing. Interestingly, as in the examples above show, there is utility in pre-processing. We can make use of the punctuation to identify useful phrases in the query automatically. Empirical observations indicated that forming larger phrases with adjacent terms in the case of apostrophe usage helped further. For example, in topic 320, we can process the query to output the following query[1] in the Indri (Section 3) query language.

```
#combine (#od3[2](flag world longest)
#od2(world longest)
#od3(world longest undersea))
```

We refer to this technique as **Phrase**. The techniques Edit and Phrase are collectively referred to as **qPro** (query pre-processing).

An interaction with the user could involves questions such as those given below.

Simple questions

Is it correct that you see margaret thatcher's resignation as a phrase? Is it correct that you see who's involved as a phrase?

2.3 Identifying patterns

In addition to phrases, patterns of terms occur frequently in similar documents. The terms that constitute these patterns can occur either adjacent to each other or within a term window. Identifying these patterns can help improve precision as a sequence of terms offers a better indication of relevance compared to individual terms.

2.3.1 Implementation

For each query we selected the top 100 results from a pseudo-relevance feedback run and broke down the documents into sentences using MXTERMINATOR[10]. The sentences were then parsed to obtain co-occurance data for all pairs of terms. Once this was acquired, patterns are formed by specifying an unordered window equal to the average distance between pairs of terms, for each pair. A weighting function ((Equation **??**) based on the average distance between the pairs of terms and the number of times they co-occured was used to rank the unordered term pairs.

$$weight(t_1, t_2) = avg.dist. * inv.sent.freq.$$

$$avg.dist. = \log((avg(t_1, t_2) + 1.0)$$

$$inv.sent.freq. = \log((\#sent + 1)/((\#sent(t_1, t_2)) + 0.5))$$

To select the patterns to be included in the query, we chose the patterns that occurred in the top five thousand, and whose constituent terms all occurred in the original query.

[1] After stemming and stop-word removal
[2] An ordered window of width 3

We refer to this technique as **Patterns**.

Simple questions

"Would you expect to see leaning and pisa nearby, with terms such as tower and of between them?"

"Would you expect to see three and dam nearby, with terms such as gorges between them?"

2.4 Topic Selection

2.4.1 The problem

Automatic techniques like pseudo-relevance feedback show consistent *overall* improvement over sets of queries. Unfortunately they do not perform well on, or severely hurt, already poorly-performing queries. Overall gains are usually only due to performance improvements in queries that performed reasonably in the first iteration. This is because such techniques are sensitive to the original set of documents retrieved. If the top-ranking documents are non-relevant due to a poorly specified query, or if there are too few relevant documents in the collections, pseudo-feedback techniques perform badly. Our next technique attempts to tackle the latter issue. By *injecting* relevant documents from an external source into the higher ranks of the initial retrieved set, we can provide a better document set for pseudo-feedback-based query expansion. Researchers have previously used the Web as an external resource[6], or a gigantic external corpus[2]. The use of these resources requires an initial retrieval followed by some form of clustering to determine the most appropriate set for inclusion. This procedure does not also lend itself to be used in a *simple* interactive setting.

2.4.2 Implementation

Our approach is to also use an external corpora - the Usenet News Groups. The advantage of using this corpora is that it is already categorized by topic. The topics are generated by humans, and the assignment of documents happens by default. While this solves the problem of determining how to cluster the results from the external corpora, it still leaves us with the problem of determining which news group to select in response to a query. Our automatic solution to this problem is to select the news group to which a simple majority of the top 100 results belong to, and performed focused retrieval within this group and use the those documents for query expansion.

This technique is referred to as $ePRF_a$ (external pseudo-relevance feedback – automatic). The interactive version of this technique is $ePRF_i$ (external pseudo-relevance feedback – interactive), while the combination is referred to as $ePRF_{a+i}$ (external pseudo-relevance feedback
– automatic+interactive)

Simple questions

The query is **piracy**.

Which of the following groups do you think are related to your query?

- microsoft.public.windowsxp.general
- sci.military.naval
- alt.sailing.asa
- uk.games.video.playstation
- alt.bored

A user interested in topics related to piracy of ships will select the first and fourth groups while someone interested in software piracy would select the second and third groups. Its plausible that the user might be unable to make a decision about a certain group, like alt.bored, in which case she can simply decline to answer. The example also reveals another use of selecting topics - it makes query disambiguation very easy.

3. EXPERIMENTAL SETUP AND BASELINE

We chose the fifty TREC (Text REtrieval Conference) 2005 Robust Track queries for training, and a set of fifty randomly chosen queries from the TREC 2004 Robust Track as our test set. These queries were tested on the AQUAINT collection, and TREC disks 4&5, minus the Congressional Record, respectively. The choice of Robust Track queries was motivated by the fact that these queries are previously know to be *hard*, and the impact of standard pseudo-relevance feedback is less compared to other query sets. Thus these queries are good candidates for testing the impact of our proposed techniques.

3.1 Corpus

The AQUAINT[3] corpus consists of newswire text data in English, drawn from three sources: the Xinhua News Service, the New York Times News Service, and the Associated Press Worldstream News Service. The corpus covers the period from January 1996 to September 2000, inclusive, for the Xinhua text collection, and from June 1998 to September 2000, inclusive, for New York Times and Associated Press. The total number of documents is around one million.

TREC disk 4 consists of newswire text data from the Financial Times Limited (1991, 1992, 1993, 1994) and the Federal Register (1994). Material is available from NIST Standard Reference Data Products. TREC disk 5 consists of material from the Foreign Broadcast Information Service (1996) and the Los Angeles Times (1989, 1990). There are in all around five hundred thousand documents.

3.2 Queries

TREC queries consist of a title, description and narrative section. Each of these sections contains progressively longer descriptions of the information need. Our baseline queries consisted of terms from the title and description portions of the queries. Including the narrative would have made the queries unrealistically long and over-specified.

3.3 Retrieval System and Baseline

As our retrieval system, we used version 2.2 of the open-source Indri[4] system. We used the 418 stopwords included in the stop list used by the InQuery system, and the K-stem stemming algorithm implementation provided as part of Indri.

Our baseline system (QL) is a query-likelihood variant of statistical language modeling. Given a query

$$Q = q_1 q_2 q_3 \dots q_n,$$

and a document

$$D = d_1 d_2 d_3 \dots d_n,$$

[3] `http://www.ldc.upenn.edu`

[4] `http://www.lemurproject.org/indri`

the probability $P(Q|D)$ that the query would be generated by the document is

$$P(Q|D) = \prod_{j=1}^{n} P(q_j|D)$$

with

$$P_{ML}(q_j|D) = \frac{c(q_j; D)}{\sum_{i=1}^{n} c(w_i; D)}$$

where $c(q_i; D)$ represents the number of times that term q_i occurs in document D and ML refers to maximum likelihood.

The pseudo-relevance feedback mechanism is based on relevance models[7]. Relevance modeling build a language model (probability distribution) of the vocabulary that is likely to occur in relevant documents. It does so by looking for the probability that words co-occur with query terms throughout the corpus. Although the formal framework is elaborate and quite powerful, the implementation boils down to a variation on typical pseudo-relevance feedback. That is, the initial query is used to rank documents and the top several documents are assumed to be relevant. The vocabulary of those documents is analyzed to calculate a probability distribution of words that are related to the query—because the words occur in high-ranking documents. The resulting probability distribution is used as an additional component of the query with expanded vocabulary. Relevance models consistently improve retrieval performance over simpler language modeling approaches, and meet or beat other techniques based on automatic query expansion. We used the top 25 documents for feedback, and added 25 terms to the original query.

For all systems, we report mean average precision (MAP), geometric mean average precision (GMAP), and percentage of queries improved over the QL system. MAP is the most widely used measure in Information Retrieval. While precision is the fraction of the retrieved documents that are relevant, average precision is a single value obtained by averaging the precision values at each new relevant document observed. MAP is the mean of the average precisions of a set of queries. Similarly, GMAP is the geometric mean of the average precisions of a set of queries. Our focus is on the GMAP measure as it is more indicative of performance across an entire set of queries. MAP can be skewed by the presence of a few well-performing queries, and hence is not as good a measure as GMAP from the perspective of measure comprehensive performance.

4. RESULTS

We ran automatic versions of all the techniques described in isolation as well as in combination with other techniques. Tables 1 and 2 summarize our results. To start with, we can observe that collections, PRF improves over the baseline QL system in terms of both MAP and GMAP. This is interesting as it indicates that in these collections PRF not only improves MAP but also GMAP, indicating that the improvements are spread across queries.

4.1 baseline + qPro

Baseline + qPro refers to the queries that include automatic versions of the Edit and Phrase techniques. The impact of these techniques cannot of gauged directly from the MAP and GMAP scores as not all queries are affected by these techniques. Hence, even though the overall effect might be small, effects on the individual queries are significant. We can also observe that both the QL and PRF scores for this query set are better than those of the baseline.

4.2 baseline + qPro + Patterns

We next experimented with a combination of qPro and Patterns. Observation of the scores reveals that both MAP and GMAP are hurt. While this conveys that the Patterns technique might not be useful, inspection of the queries revealed that the technique was identifying good patterns, but included a number of spurious patterns too. We believe that the interactive version of Patterns will ameliorate this problem and perform better.

4.3 baseline + qPro + Patterns + $ePRF_a$

This was the final automatic technique combination we explored. From the tables, we can observe that the external expansion played a significant role in boosting the performance, already lowered by patterns, to the most competitive score, in terms of both MAP and GMAP. This indicates that the $ePRF_a$ technique was well suited for all types of queries.

5. USER IN THE LOOP

All the experiments reported this far are automatic. To test the hypothesis that including the user in the loop, i.e. asking the user to help make choices previously done in a automatic manner does indeed improve results, we experimented with topic selection (Section 2.4). One of the authors acted as a user. The query chosen was the one with automatic query pre-processing, and automatic pattern addition. With this query, we explored whether asking the user to select topics related to the query will score over the automatic external expansion described in Section **??**.

The query was submitted to the Usenet archive and the top Usenet group titles were displayed to the user. The user was asked to choose as many groups (from just reading the name of the group) as he felt would contain information pertinent to the query. Once the groups were selected, a second focused retrieval was performed to retrieve documents from the groups selected. This served as the external corpus for the query, and PRF was performed using this corpus in addition to the original corpus. The results for both collections are shown in tables 4 and 3. For the Robust 04 collection, the $ePRF_a$ performs slightly better than PRF. The utility of user-interaction is revealed in the results for the $ePRF_i$ system which clearly scores over the $ePRF_a$ in both MAP and GMAP. Even better is the performance when we merge the external corpora selected by the automatic and interactive system. The resulting system $ePRF_{a+i}$ performs the best on all measures.

6. RELATED WORK

Improving retrieval performance by automatically or manually reformulating queries [1, 12, 3] has been the focus of much research. Approaches that are based on the assumption that top-ranked documents are relevant to the original query can be rendered ineffective if the queries are poorly specified, or if few relevant documents are returned at the top of the ranked list. Some approaches bring the user into

System		MAP	GMAP	% Queries Improved
Baseline	QL	0.3601	0.2469	
	PRF	0.3803	0.2586	58%
Baseline + qPro	QL	0.364	0.2600	
	PRF	0.386	0.2705	24%
Baseline + qPro + Patterns	QL	0.3313	0.2127	
	PRF	0.3808	0.2317	58%
Baseline + qPro + Patterns + $ePRF_a$		- 0.3877	- 0.2659	56%

Table 1: Robust 04. Performance of the different systems on the Robust 04 dataset in terms of MAP, GMAP, and percentage of queries improved. The percentage improvement for systems other than the baseline is measured with respect to the baseline+PRF system. A system is regarded as better than another for a particular query if the MAP score due to one is higher than that due to the other

System		MAP	GMAP	% Queries Improved
Baseline	QL	0.2257	0.1581	
	PRF	0.2673	0.1610	54%
Baseline + qPro	QL	0.2301	0.1632	
	PRF	0.2677	0.160	8%
Baseline + qPro + Patterns	QL	0.2	0.0979	
	PRF	0.262	0.1167	56%
Baseline + qPro + Patterns + $ePRF_a$		- 0.2766	- 0.1396	52%

Table 2: Robust 05. Performance of the different systems on the Robust 05 dataset in terms of MAP, GMAP, and percentage of queries improved.

		QL	PRF	$ePRF_i$	$ePRF_a$	$ePRF_{a+i}$
Robust 05	MAP	0.2	0.262	0.2841	0.2766	0.3073
	GMAP	0.0979	0.1167	0.1514	0.1396	0.1997

Table 3: Results on the Robust 05 collection. The QL column corresponds to a pre-processed query with patterns added.

		QL	PRF	$ePRF_i$	$ePRF_a$	$ePRF_{a+i}$
Robust 05	MAP	0.3313	0.3808	0.3957	0.3877	0.3969
	GMAP	0.2127	0.2317	0.2751	0.2659	0.2768

Table 4: Results on the Robust 04 collection. The $ePRF_{a+i}$ performs best. It achieves a 20% improvement over PRF in GMAP

the loop by asking her to mark documents at the top of the ranked list as relevant or non-relevant, and use this information for feedback[3]. This could cumbersome to the user. Providing users with an interface to specify the query elaborately and accurately has been tried too. However, such interfaces involve issues ranging from deciding which supplementary information to ask for to the optimal design of the interface.

Some of the techniques we have developed were inspired by the results of the Reliable Information Access [4] workshop report. The outcome of the workshop was an ontology of reasons why current IR systems fail. The Patterns technique draws comparison with the dependency model[9]. However, the patterns generated by the dependency model are query dependent, which the Patterns technique generated corpus-specific patterns.

There hasn't been much work with the goal of improving retrieval performance with minimal participation from the user. We believe our previous work[5] is a step in that direction.

7. CONCLUSIONS AND FUTURE WORK

We believe that in many cases it will not be possible for an IR system to automatically infer the correct meaning of an ambiguous query. The motivation for this work, therefore, is to find questions that can be asked of a searcher and that can improve performance. Furthermore, we are not interested in complex questions or in questions that require significant time from the searcher. Instead, we aim for questions that are short and that can usually be answered "yes" or "no" or by selecting from a very small set of options.

This study has explored a small set of such questions and demonstrated that their use can substantially improve performance on appropriate queries. Although some of our experiments have sidestepped the actual questions, we envision each of the techniques being used interactively. In addition to looking for additional *simple* questions, the next steps of our work involve developing interfaces that shows these questions and allow us to explore their usability as well as their utility. We do not believe this approach will help with all queries—in particular, it is unlikely to provide value for "easy" queries that are already handled well—but if we can improve a substantial number, including perhaps poorly performing queries, it will be worthwhile.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *ACM SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 88–95, 2003.

[2] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR '06: To Appear in Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Informa tion Retrieval*, New York, NY, USA, 2006. ACM Press.

[3] D. Harman. Towards interactive query expansion. In *ACM SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 321–331, 1988.

[4] D. Harman and C. Buckley. Reliable information access final workshop report. 2003.

[5] G. Kumaran and J. Allan. Simple questions to improve pseudorelevance feedback results. In *SIGIR '06: To Appear in Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2006. ACM Press.

[6] K.-L. Kwok, L. Grunfeld, and P. Deng. Improving weak ad-hoc retrieval by web assistance and data fusion. In *AIRS*, pages 17–30, 2005.

[7] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA, 2001. ACM Press.

[8] J. Lin, E. Abels, D. Demner-Fushman, D. W. Oard, P. Wu, and Y. Wu. Menagerie of tracks at maryland: Hard, enterprise, qa, and genomics, oh my! In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[9] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479, New York, NY, USA, 2005. ACM Press.

[10] J. C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[11] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532, 1998.

[12] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *ACM SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.