Evaluating High Accuracy Retrieval Techniques

Chirag Shah W. Bruce Croft

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{chirag,croft}@cs.umass.edu

ABSTRACT

Although information retrieval research has always been concerned with improving the effectiveness of search, in some applications, such as information analysis, a more specific requirement exists for *high accuracy* retrieval. This means that achieving high precision in the top document ranks is paramount. In this paper we present work aimed at achieving high accuracy in ad-hoc document retrieval by incorporating approaches from question answering (QA). We focus on getting the first relevant result as high as possible in the ranked list and argue that traditional precision and recall are not appropriate measures for evaluating this task. We instead use the mean reciprocal rank (MRR) of the first relevant result. We evaluate three different methods for modifying queries to achieve high accuracy. The experiments done on TREC data provide support for the approach of using MRR and incorporating QA techniques for getting high accuracy in ad-hoc retrieval task.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)* ; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

General Terms

Measurement, Performance, Experimentation

Keywords

High accuracy retrieval, ad-hoc retrieval, question answering

1. INTRODUCTION

When we look at two major research streams in the present information retrieval (IR) community, *i.e.*, ad-hoc retrieval and question answering (QA), we find a well-defined set of methodologies and metrics for measuring the performance. Ad-hoc retrieval typically involves retrieving a set of documents for a given query and ranking them using their relevance to the query. The performance is usually measured with precision and recall or some variations of them [24]. Question answering, on the other hand, involves steps such as analyzing the question for its type [35], creating surface patterns [23], retrieving and ranking passages [3], and identifying the answer. There are also some techniques that both of these streams have investigated such as word sense disambiguation [25, 21], expanding the query using some lexical resource like WordNet[1] [20, 31], etc. However, at the root, these tasks have different goals. Ad-hoc retrieval is mainly about retrieving a set of documents with good precision and recall, whereas QA focuses on getting *one* correct answer or a small set of answers with high accuracy. In this paper we try to link these two by performing ad-hoc retrieval with the goal of QA, *i.e.,* achieving high accuracy with respect to the most relevant results.

The goal of achieving high accuracy (i.e. high precision at the top ranks) is particularly important for some applications. Any system that has a limitation on the bandwidth of the user interface, such as with mobile devices, or where there is a requirement for additional processing on the results, such as in cross-lingual settings, will have a requirement for accuracy. Recognizing the importance of achieving high accuracy in retrieval, TREC[2] introduced a new track called *High Accuracy Retrieval from Documents (HARD)*[3] in 2003. This track focused on achieving high accuracy retrieval using some feedback from the user (*e.g.*, expertise, purpose) or some other meta-data (*e.g.*, genre of the document). The retrieval could be at any level including document, passage, phrase, or words. Our task also deals with the problem of getting high accuracy in retrieval, but with contrast to HARD, we do not make use of any additional information. Also, we use only the document as the unit of retrieval as in conventional ad-hoc retrieval.

In the light of the above issues, we study how QA techniques can help in getting high accuracy for ad-hoc retrieval and propose a different measure for evaluation (MRR) instead of recall and precision. We also have carried out an analysis of ad-hoc retrieval from a QA perspective.

The rest of the paper is organized as follows. In section 2 we analyze the problem of high accuracy retrieval for ad-hoc retrieval. Specifically, we evaluate an ad-hoc retrieval run using MRR and compare the performance of the system with QA system performance in general. We also analyze the performance of queries that perform very badly. Then, in section 3, we present three approaches to boost the accuracy of ad-hoc retrieval. These approaches are inspired by some work in QA domain. The evaluation of these ap-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
Copyright 2004 ACM 1-58113-881-4/04/0007 ...$5.00.

[1] http://www.cogsci.princeton.edu/~wn/
[2] http://trec.nist.gov/
[3] http://ciir.cs.umass.edu/research/hard/

proaches and a discussion of the results are presented in section 4. The results support our hypothesis that using QA-like techniques on ad-doc retrieval can improve high accuracy performance. We conclude the paper with some discussion of future work in section 5.

2. ANALYSIS OF THE PROBLEM

With exponentially increasing digitized text collections on the Web and in other repositories [13], it has become easier to achieve good recall, but the growing concern of the user is to get more accuracy [5]. Search engines typically return hundreds or thousands of results to a user's query. They are likely to contain the information that the user is seeking, but unless the required results are at the top of the ranked list, this information will not be useful. Specifically, if the user is looking for just one or two relevant results similar to answers in a QA system, then the effectiveness of the system depends on how high these results are in the rank list instead of the overall precision or recall. To facilitate the evaluation of a system with this focus, QA systems use mean reciprocal rank (MRR) as the measure [14], which is defined as the inverse of the rank of the retrieved result. The higher it is, the better, with the best case being MRR=1.0 (when the result is at rank 1). Since our task is similar, *i.e.*, getting the first relevant result as high as possible in the rank list, we adopt this approach to evaluating performance. This section investigates the problem of achieving high accuracy with this perspective and analyzes the reasons behind low accuracy.

2.1 Looking at ad-hoc retrieval from a QA perspective

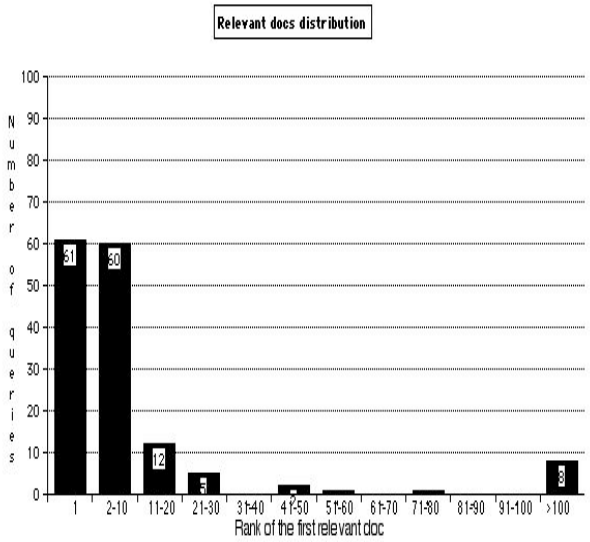

Figure 1: Relevant document distribution with *title* queries - baseline (run#0)

Since we want to do ad-hoc retrieval from the perspective of a QA system, it is important to understand the limitations of the former system and the effectiveness of the latter one. However, it becomes difficult to compare the performance of these two systems given that they have different queries and relevance judgements even for the same test corpus. Nevertheless, we want to define a baseline to understand how to improve its accuracy to meet the standards of a QA system. To do this, we selected TREC's Tipster Vol. I and II as datasets and topics 51-200 (total 150) as queries

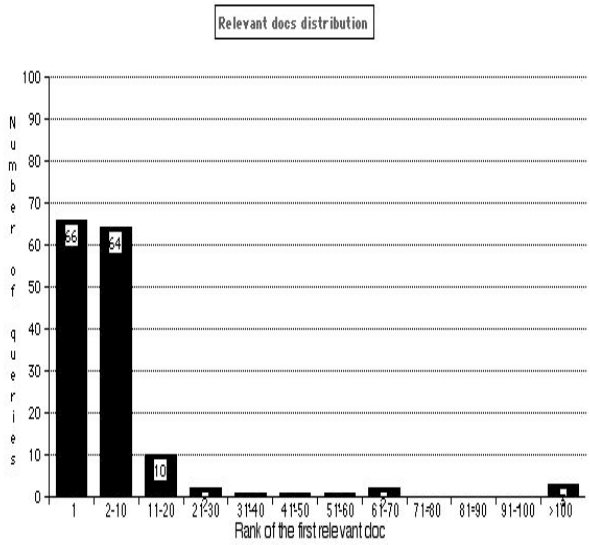

Figure 2: Relevant document distribution with *description* queries - baseline (run#0)

(both *title* and *description*). For all experiments reported in this paper, we have used the language modeling framework for retrieval, as described in [22, 1]. In this approach, we build a language model for each document. The ranking of a document for a query is based on the probability with which the query terms are generated by the document's language model as shown below:

$$P(Q|D) = P(q_1, q_2, .., q_n|D) = \prod_{i=1}^{n} P(q_i|D) \qquad (1)$$

where the last term in the above equation is obtained from the assumption of conditional independence of terms given the document's language model. We used the Lemur toolkit[4] for implementing our retrieval system. The standard stopword list of Lemur and K-stem [10] were used for stopword removal and stemming, respectively. All our runs are performed using both *title* and *description* parts of the queries. We also made use of structured queries [18] as required. The results of our baseline runs along with those of various QA systems of TREC-8 [26], TREC-9 [27], TREC-10 [28], and TREC-11 [29] are given in table 1. In the case of our baselines, the *Correct answers* column indicate the percentage of queries for which the first relevant document was at rank one. Values for TREC systems are given as medians from the performance of various systems presented that year.

It is important to note that this table is not for direct comparison, but just to have an idea of the relative performance of an ad-hoc retrieval and a QA system. We can see that our baselines have a correct document in rank 1 about 40% of the time, which compares favorably to the performance of various QA systems. The average MRR for both baselines is also more than 0.5, which means that on average a relevant document occurred higher than rank 2. However, if we look at the distributions of queries with respect to the rank of their first relevant documents as shown in figures 1 and 2, we can see that there are almost as many queries in rank 2-10 range as there are at rank 1. The average MRR could be increased by moving up relevant documents from lower ranks in the case of poorly performing queries, and by moving some of the big group

[4]http://www-2.cs.cmu.edu/~lemur/

Table 1: Results of various QA systems as presented in TREC over the years. The last two systems are our baselines with ad-hoc retrieval. Please note that the figures given in this table are not here for comparison as they are on different datasets and measured differently. They merely give an idea about the effectiveness of present QA systems and our baselines.

System	Dataset	Number of docs	MRR (median)	Correct answers (median)
TREC-8 (1999)	TREC Disks 4, 5	528,000	For 50-byte responses: 0.2610 For 250-byte response: 0.3830	39.50% 52.50%
TREC-9 (2000)	News from TREC Disks 1-5	979,000	For 50-byte responses: 0.2300 For 250-byte responses: 0.3700	34.00% 48.00%
TREC-10 (2001)	News from TREC Disks 1-5	979,000	For strict evaluation: 0.3600 For lenient evaluation: 0.3600	44.70% 42.90%
TREC-11 (2002)	AQUAINT Corpus	1,033,000	0.4980 (Confidence weighted score)	29.80%
Ad-hoc retrieval with *title* queries	Tipster vol. I, II	741,856	0.5492	40.67%
Ad-hoc retrieval with *description* queries	Tipster vol. I, II	741,856	0.5745	44.00%

of relevant documents at rank 2 to rank 1 . This shows that there is considerable opportunity for improvement.

2.2 Analyzing some bad queries

In order to understand why some queries fail to achieve good performance, we investigated the queries whose MRR values were quite low (less than 0.1). We not only studied the reasons behind the failure of these queries, but also changed the queries *manually* and tested again for retrieval to verify our hypothesis. Table 2 provides these details about some of the queries that we analyzed. Looking at the the results in table 2, we can make the following observations.

1. We could improve the MRR value in all the cases by *improving* the query.
2. There were three problems that we could identify: ambiguous words in the query, mixtures of words of different importance in the query, and query-document information mismatch.
3. Not all the words are equally important in a given query. Often, expanding the query helps, but if we expand the *wrong* words, then it can hurt retrieval performance.
4. Not all kinds of expansion can help. Even if we know which words are ambiguous or good for expansion, we cannot simply add *any* words. Sometimes synonyms help, and sometimes other related words provide the missing information.

It is clear from the above observations that achieving high accuracy will require modifying queries or some words of the queries using techniques such that the queries that are performing badly should be helped, but the others that are doing well should not be hurt. The next section proposes some methods for doing such *selective manipulation* to the queries.

3. OUR APPROACHES FOR MODIFYING QUERIES

As discussed earlier, though we are working in an ad-hoc retrieval-like scenario, *i.e.*, retrieving documents for a given query, our goal is more similar to that of a QA system. We, therefore, looked at some popular QA techniques to see if and how they can fit in to our system. Some of the interesting ideas that we found useful for our work are converting natural language queries to database queries [17], question expansion [8], identifying the role of each word in the question [2], and use of various NLP techniques for question processing [9]. After studying many such techniques from QA domain, we came up with the following three approaches.

Method 1: Giving more weight to the headwords

Even after stop-word removal [34] and stemming [15], we find that not all the words in the query are equally important and we should not treat them evenly. Some QA systems analyze the given question to find the headwords of that sentence [19]. For instance, in the question *What river in the US is known as the Big Muddy?* has *river* as the headword. Identifying the headword helps in focusing the search for the right answer. We adopt the technique described below to find the headword in a given query and giving it more weight than other words of the query.

1. Parse the query for part-of-speech (POS) tagging. We used Supertagger [11] for this.
2. Find the first noun phrase using POS information.
3. Consider the last noun of this noun phrase as the headword.
4. Give this headword more weight (we gave double) than normal words and reconstruct the query.

Method 2: Using clarity scores as weights

The reason for poor retrieval is often the use of ambiguous words in the query [6]. To address this issue, we can use a simple heuristic that the more ambiguous the word is, the less importance should it be assigned. To implement this idea, we used Cronen-Townsend *et al.*'s [7] technique of finding query clarity scores. In their paper the authors show how to predict the query performance by computing the relative entropy between a query language model and the corresponding collection language model. The resulting clarity score measures the coherence of the language usage in documents whose models are likely to generate the query. They used these clarity scores to identify ineffective queries. Here our objective is also to find the *effectiveness* of different words of the query and assign relative weightage to them. The following procedure demonstrates how we implemented this idea.

Table 2: Analysis of some badly performing *title* queries. We also resolved the problems manually and reran them. The MRR for both original and new runs are given here. The MRR is calculated using the first relevant result.

Topic	Query	Problem and solution	Original MRR	New MRR
59	Weather Related Fatalities	*Fatalities* is not a common word, Adding its synonyms helped. Also, *related* is not a useful word. Therefore, we gave different weights for each word.	0.0286	0.0667
64	Hostage-Taking	*Hostage-taking* may be good from query perspective, but not likely to occur in the documents. Adding some related words helped.	0.0016	0.0833
73	Demographic Shifts across National Boundaries	*Demographic shift* is not a common phrase. Adding some synonyms and weighting each word differently helped.	0.0417	0.0769
75	Automation	Too short and ambiguous. Adding its related words helped.	0.0278	0.5000
85	Official Corruption	Query-collection mismatch. Adding synonyms of *corruption* helped.	0.0200	1.0000
88	Crude Oil Price Trends	*Trends* is not as common as *business*. Therefore, adding *business* and weighting each term differently helped.	0.0025	0.0143
98	Fiber Optics Equipment Manufacturers	*Manufacturer* was not common in this context in the collection. Adding *producers* helped.	0.0087	1.0000
118	International Terrorists	Query-collection mismatch problem. Adding the related word *terrorism* helped.	0.0333	1.0000
120	Economic Impact of International Terrorism	Query-collection mismatch problem. Adding the related word *terrorists* helped.	0.0263	0.2000

1. Find query clarity scores based on the technique given in [7]. We find clarity scores not only for the queries, but also for each term of the query.
2. Construct weighted queries with clarity score of each word as its weight as we want to give more weight to words that have high clarity scores.

Method 3: Using clarity scores to find terms to expand with WordNet

Query-dataset mismatch is another factor that affects the accuracy of the retrieval. This factor essentially arises when the information is not presented the way it is asked in the query. For instance, if the query is about *weather related fatalities* (topic 59 in TREC ad-hoc retrieval task) and the documents have this information represented as something like *deaths by abrupt changes in weather*, then they may not get high ranks. This problem can be solved if we supply additional information, *viz.*, synonyms of *fatalities* in the original query. Many have used lexical resources like WordNet for doing such query expansion [32]. This approach has been shown to increase recall, but not necessarily precision and here our task is to improve accuracy as defined in our goals earlier. Therefore, it is not useful to expand *every* word of the given query even if it improves recall. For selectively expanding words, we again look at the clarity scores. The steps of this method are enumerated below.

1. Find query clarity scores based on [7]. Again, we find clarity scores not only for the entire query, but also for each term of the query.
2. These scores represent how clear a term is. Therefore, we follow this simple heuristic: divide all the terms into the following three categories and perform the appropriate actions.
   (a) Terms with high clarity scores should not be touched. Therefore, they are left in the query.
   (b) Terms with very low clarity scores are likely to be very ambiguous and expanding them is very likely to bring more noisy words. Therefore these words are ignored.
   (c) Expand the terms whose clarity scores are between the two limits of clarity scores[5] using WordNet synonyms.

Using this method we are addressing two problems: getting rid of the words that are so ambiguous that they cannot help in retrieval, and helping those words with not so bad clarity scores by including their WordNet synonyms.

4. EXPERIMENTS AND ANALYSIS

To implement the methods proposed in the previous section, we used TREC's dataset disks 1 and 2 comprising more than 700,000 documents from Tipster collection and taking more than 2 GB of disk space. The corresponding queries were extracted from topics 51-200 making total 150 queries. The experiments were conducted on both *title* queries as well as *description* queries. The following subsections present the results of various experiments along with the analysis.

4.1 Experiments with title queries

The experiments done on *title* queries and their results are given in the table 3 and plotted in figures 3 to 5. The following observations can be made from these results.

[5]We determined this limits empirically and based on some observations.

Table 3: Results of the runs with *title* queries. Average MRR is calculated considering the rank of the first relevant document, whereas average precision is found using standard TREC measures from entire rank list. With each of our runs we also give percentage improvements with respect to the baseline and the p value from two tailed paired t-test with 95% confidence interval. Bold cases show that the results are statistically significant. Up or down arrows indicate better or worse respectively.

#	Method	Avg. Prec.	Avg. MRR	Relevant doc on rank 1
0	Baseline	0.1873	0.5492	40.67%
1	Headwords weights=2	0.1737 **(-7.26%, p=0.0012)** ↓	0.5128 **(-6.63%, p=0.0065)** ↓	36.00% **(-11.48%, p=0.0191)** ↓
2	Using clarity scores as weights	0.1867 (-2.51%, p=0.8863)	0.5659 (+3.04%, p=0.1198)	44.67% (+9.84%, p=0.0575)
3	Using clarity scores for finding terms to expand with WordNet	0.1963 (+4.81%, p=0.1664)	0.5541 (+0.89%, p=0.8423)	46.67% (+16.40%, p=0.0717)

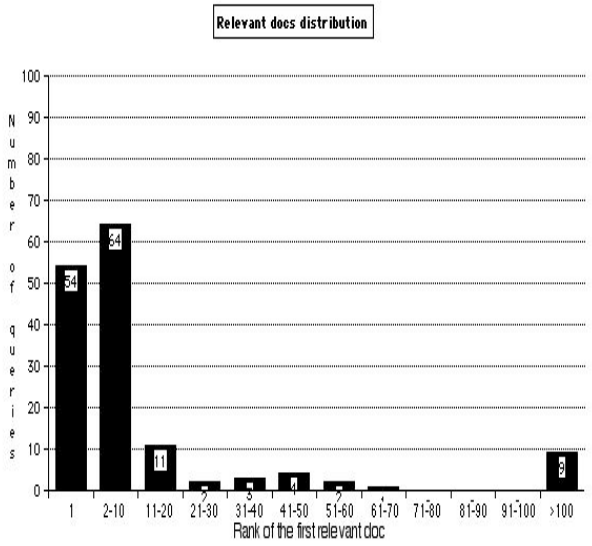

Figure 3: Relevant document distribution with *title* queries - more weight to headwords (run#1)

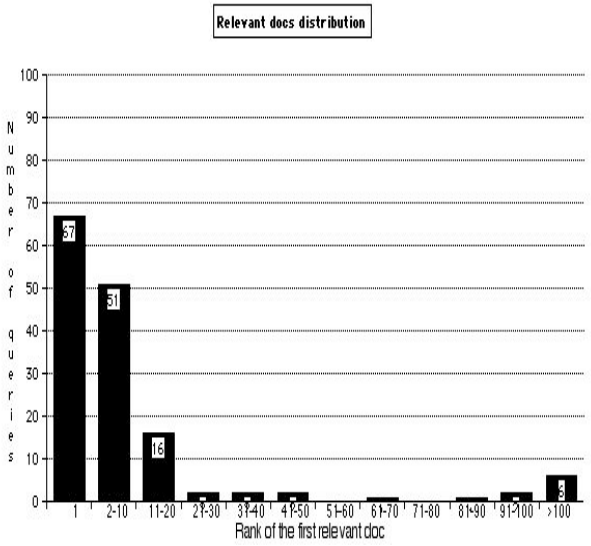

Figure 4: Relevant document distribution with *title* queries - using clarity scores as weights (run#2)

- The first run is when we extracted headwords and gave them more weight than normal words. However, *title* queries are typically about 2-3 words long without proper sentence structure. Therefore, the technique of finding headwords does not perform as effectively as in the QA domain where the questions have proper sentence structure. The idea behind using headwords is to focus on important words in the given query, but in the case of *title* queries, the words are generally keywords and they are *all* likely to be important. Thus, we got worse performance when we tried using headwords for *title* queries.
- In the case of run number two, the average precision value goes down compared to the baseline, but average MRR increases. The increase in percentage of relevant documents on rank 1 also shows that run two is better than the baseline. This indicates that normal precision measure may not be correct for the task that we have, *i.e.*, getting high accuracy in terms of getting the first relevant result high in the rank list.
- The third run, which uses clarity scores and selectively expands words using WordNet, gives the best performance increasing not only average MRR, but also average precision.

4.2 Experiments with description queries

The experiments done on *description* queries and their results are given in table 4 and plotted in figures 6 to 8. The following observations can be made from these results.

- We again observe that run number one has got less average precision than the baseline, but has higher average MRR. The results about percentage of relevant documents at rank 1 also reflect that run number one is better than the baseline. This again supports the fact that in a task like this, precision or recall are not always correct measures to use.
- Run number two gives better average precision and significantly better average MRR.
- Run number three gives the best results with significant improvements in average precision as well as average MRR.

4.3 Overall analysis

We can notice the following points from the results of all the runs.

- Wherever we have more queries getting the first relevant document at rank 1, we have less queries in ranks 2-10. This

Table 4: Results of the runs with *description* queries. Average MRR is calculated considering the rank of the first relevant document, whereas average precision is found using standard TREC measures from entire rank list. With each of our runs we also give percentage improvements with respect to the baseline and the p value from two tailed paired t-test with 95% confidence interval. Bold cases show that the results are statistically significant. Up or down arrows indicate better or worse respectively.

#	Method	Avg. Prec.	Avg. MRR	Relevant doc on rank 1
0	Baseline	0.1766	0.5745	44.00%
1	Headwords weights=2	0.1449 **(-17.95%, p=0.0000)** ↓	0.5841 (+1.67%, p=0.6497)	44.67% (+1.52%, p=0.8356)
2	Using clarity scores as weights	0.1892 (+7.13%, p=0.0933)	0.6302 **(+9.69%,p=0.0097)** ↑	51.33% **(+16.66%, p=0.0213)** ↑
3	Using clarity scores for finding terms to expand with WordNet	0.2350 **(+33.07%, p=0.0000)** ↑	0.6403 **(+11.45%, p=0.0155)** ↑	52.00% **(+18.18%, p=0.0451)** ↑

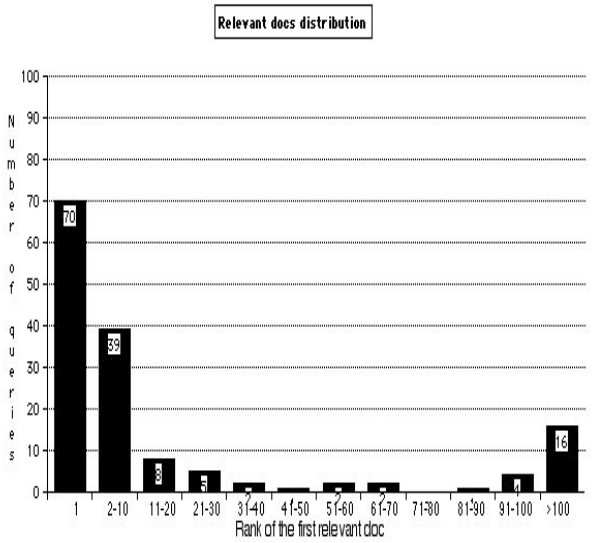

Figure 5: Relevant document distribution with *title* queries - using clarity scores for finding terms to expand with WordNet (run#3)

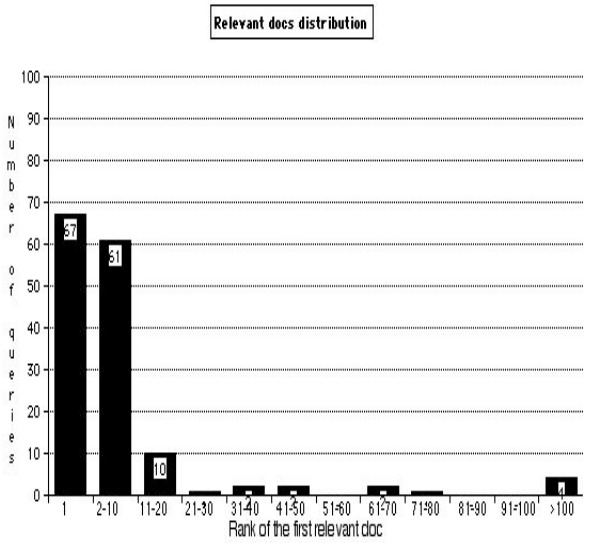

Figure 6: Relevant document distribution with *description* queries - more weight to headwords (run#1)

shows that we can *improve* queries that were already performing reasonably.

- We can see in the runs for *title* queries that as we pushed more queries to rank 1, we also got more queries at ranks higher than 100. This means that while trying to improve the queries, we also hurt some queries. We could get better performance in runs two and three, but they were not significantly better than the baseline.
- Runs for *description* queries did quite well in that, while bringing more queries to top rank, we did not make other queries go down in the rank list. This is mainly because we used techniques from QA domain that assume proper sentence-like structure in the query or question. *Title* queries could not offer such structure, while *description* queries could. As we can see from the results of *description* queries, we got improvements in MRR in all the cases, the second and third runs being significantly better.

5. CONCLUSION AND FUTURE WORK

In this paper we presented a different perspective for looking at high accuracy document retrieval. We argued that traditional measures of ad-hoc retrieval are not appropriate for such high accuracy retrieval task and supported it with extensive experiments. It was clear that when the task is to get the first relevant document as high as possible in the rank list, the query need to be made as precise and expressive as possible. To obtain such *better* queries, we proposed three methods inspired from QA literature. We showed improvements in results in almost all the cases with *title* as well as *description* queries using our methods. In some of the cases we could even get statistically significant improvements.

Although our focus in the presented work was to improve the MRR of the first relevant document only, the proposed techniques also helped in improving overall precision in many cases. This indicates that selectively using some methods from QA domain can help in traditional ad-hoc retrieval.

The work that we presented here may seem similar to the home-page finding problem [4]. In the nutshell, this problem deals with returning home-pages based on the request given. Since there are not many home-pages for a person or an organization, most of the

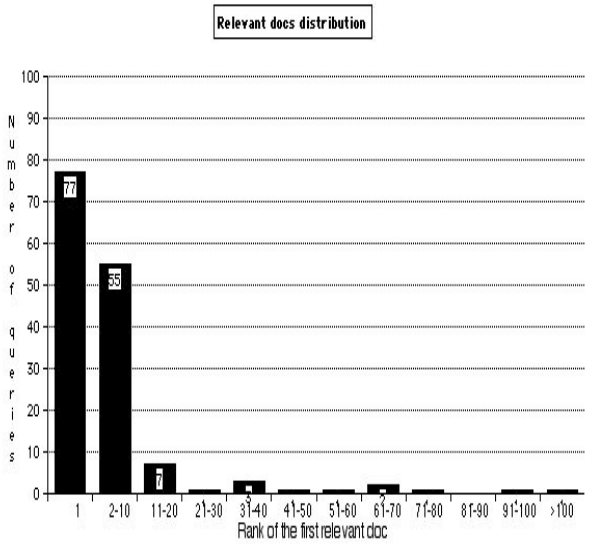

Figure 7: Relevant document distribution with *description* queries - using clarity scores as weights (run#2)

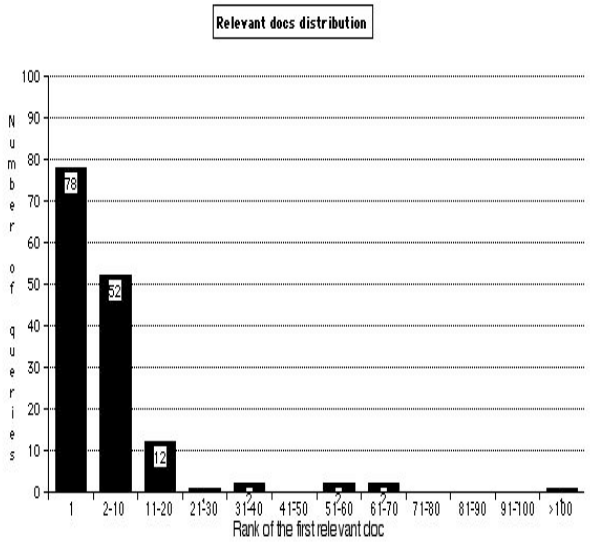

Figure 8: Relevant document distribution with *description* queries - using clarity scores for finding terms to expand with WordNet (run#3)

times the expectation is to receive one or two results. Therefore, this task also requires high accuracy. However, the home-page finding task takes advantage of various additional features like the URL and title of a page, HTML tags, link information, etc. We do not have any such information or domain-specific knowledge at our disposal. Our task is also more general as we make no assumption about type of request or the information to return.

One of the techniques that we proposed involved expanding the query. A good amount of research has been done for query expansion [30, 32, 33]. These techniques have helped a lot to improve recall and, to a certain extent, precision in ad-hoc retrieval. However, instead of helping, these techniques can often reduce effectiveness [16]. While doing experiments, we also realized that expanding *any* query in its entirety is not useful for achieving high accuracy. Therefore, we proposed a technique for *selective expansion*, in which we showed how to use query clarity scores to determine which words to expand and which words to ignore.

As one of our next steps in this research, we carried out experiments with relevance models [12], which does automatic query expansion, to understand how the techniques we have proposed would perform in that environment. We observed that in general, the relevance models give better results compared to normal query likelihood method of retrieval. However, in some of our runs, using relevance models hurt the performance. In particular, we noticed that while bringing some queries up in the rank list, the model also drove some other down in the list. Further investigation of making careful use of relevance models is under progress.

We also plan to develop a formal basis for the use of clarity measures in the expansion process. We would also like to extend our work to some more focused problems like home-page finding or HARD-like tasks. Since these domains are specific and we can either use domain knowledge as in the case of home-page finding, or meta-data or some other form of feedback as in the case of HARD, we hope to achieve even better results with them. As noticed in some of the cases, while some queries got improved, some also got hurt. It is quite likely that a technique that could push the queries from the high rank range to the top rank is not appropriate for those documents further down in the list. Therefore, we may need to combine more than one techniques to deal with this issue. We are also exploring some other techniques from the QA domain that can help us in achieving high accuracy in ad-hoc retrieval.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903 and in part by NSF grant #IIS-9907018. Any opinions, findings and conclusions or recommendations expressed in this material are of the author(s) and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] Adam Berger and John D. Lafferty. Information retrieval as statistical translation. In *Research and Development in Information Retrieval*, pages 222–229, 1999.

[2] Sabine Buchholz and Walter Daelemans. Shapaqa: Shallow parsing for question answering on the world wide web. In *Proceedings Euroconference Recent Advances in Natural Language Processing (RANLP)*, pages 47–51, 2001.

[3] C.L.A. Clarke, G.V. Cormack, D.I.E. Kisman, and T.R. Lynam. Question answering by passage selection (multitext experiments for trec-9). In *Proceedings of Text REtrieval Conference*, 2000.

[4] N. Craswell and D. Hawking. Overview of the TREC-2002 Web Track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC)*, November 2002.

[5] W. Bruce Croft. What do people want from information retrieval? *D-Lib Magazine*, November 1995.

[6] S. Cronen-Townsend and W. B. Croft. Quantifying query ambiguity. In *Proceedings of HLT*, pages 94–98, 2002.

[7] S. Cronen-Townsend, Y. Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of the ACM Conference on Research in Information Retrieval (SIGIR)*, 2002.

[8] Ralph Grishman. Information extraction: Techniques and challenges. In *SCIE*, pages 10–27, 1997.

[9] Ulf Hermjakob and Abdessamad Echihabi. Natural language based reformulation resource and web. In *Proceedings of the Eleventh Text REtrieval Conference (TREC)*, November 2002.

[10] D. A. Hull. Stemming algorithms - a case study for detailed evaluation. *JASIS*, pages 70–84, 1996.

[11] Aravind K. Joshi and B. Srinivas. Disambiguation of super parts of speech (or supertags): Almost parsing. In *COLING-94*, 1994.

[12] V. Lavrenko and W.B.Croft. Relevance-based language models. In *Proceedings on the 24th annual international ACM SIGIR conference*, pages 120–127, 2001.

[13] Steve Lawrence and C. Lee Giles. Searching the Web: General and Scientific Information Access. *IEEE Communications Magazine*, January 1999.

[14] Jimmy Lin and Boris Katz. Question answering techniques for the world wide web. In *Tutorial presentation at EACL*, 2003.

[15] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1-2):22–31, 1968.

[16] Rila Mandala, Takenobu Tokunaga, and Hozumi Tanaka. The use of WordNet in information retrieval. In Sanda Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, pages 31–37. Association for Computational Linguistics, Somerset, New Jersey, 1998.

[17] Wesley W. Chu Frank Meng. Database query formation from natural language using semantic modeling and statistical keyword meaning disambiguation. Technical Report 990003, 16, 1999.

[18] Donald Metzler and W. Bruce Croft. Combining the Language Model and Inference Network Approaches to Retrieval. *Journal of Information Processing and Management*, 2003.

[19] Donald Metzler and W. Bruce Croft. Analysis of statistical question classification for fact-based questions. *Submitted to Journal of Information Retrieval*, 2003.

[20] George A. Miller, Richard Beckwith, Christian Fellbaum, Derek Gross, and Katherine Miller. *Introduction to WordNet: An On-line Lexical Database*. Cognitive Science Laboratory, Princeton University, August 1993.

[21] Pedersen, Ted, and Rebecca Bruce. Distinguishing word senses in untagged text. In *Proceedings of the 2nd Conference on Empirical Methods in NLP (EMNLP-2)*, Providence, August 1997.

[22] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

[23] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of ACL*, 2002.

[24] Gerald Salton, editor. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[25] M. Sanderson. Word Sense Disambiguation and Information Retreival. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieaval*, pages 142–151. Springer-Verlag, 1994.

[26] E. Voorhees. The TREC-8 Question Answering Track Report. In *Proceedings of the Eighth Text REtrieval Conference (TREC)*, November 1999.

[27] E. Voorhees. Overview of the TREC-9 Question Answering Track. In *Proceedings of the Ninth Text REtrieval Conference (TREC)*, November 2000.

[28] E. Voorhees. Overview of the TREC 2001 Question Answering Track. In *Proceedings of the Tenth Text REtrieval Conference (TREC)*, November 2001.

[29] E. Voorhees. Overview of the TREC 2002 Question Answering Track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC)*, November 2002.

[30] E. M. Voorhees. On expanding query vectors with lexically related wordds. Technical report, NIST, 1993.

[31] E. M. Voorhees. Using WordNet to disambiguate word sense for text retrieval. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, 1993.

[32] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, 1994.

[33] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.

[34] Yiming Yang and John Wilbur. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society of Information Science*, 47(5), 1996.

[35] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM Press, 2003.