UMass at TREC 2003: HARD and QA

Nasreen AbdulJaleel, Andres Corrada-Emmanuel, Qi Li,
Xiaoyong Liu, Courtney Wade, and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst

The Center for Intelligent Information Retrieval (CIIR) at UMass Amherst participated in two tracks for TREC 2003: High Accuracy Retrieval from Documents (HARD) and Question Answering (QA).

- In the HARD track, we developed document metadata to correspond to query metadata requirements; implemented clarification forms based on query expansion, passage retrieval, and clustering; and retrieved variable length passages deemed most likely to be relevant. This work is discussed at length in Section 1.
- In the QA track, we focused on retrieving passages that were likely to contain the answer to the question.

1 HARD track

1.1 Overview

The goal of the High Accuracy Retrieval from Documents track was to explore techniques for improving the accuracy of the top-ranked documents in response to a query. We participated in all three aspects of the problem:

- We mapped query metadata values to document metadata values that we assigned. We then adjusted the ranking of documents depending on whether their metadata matched the query metadata.
- We generated clarification forms to tease more information out of the searcher. We tried several types of clarification forms, including providing a list of keywords that might appear in relevant documents, a list of top-ranking clusters that might contain relevant documents, and a list of passages that might appear in relevant documents.
- We explored passage-level retrieval of documents to see if we could pinpoint the relevant portions of documents.

In the final analysis, all runs using metadata or clarification forms failed to outperform our best baseline run. We interpret this as an indictment of the track and of our effort. As with most new TREC tracks, the HARD track was slow to get started, had problems being clearly defined, and had poor training data. In addition, several engineering bottlenecks delayed our initial work and prevented us from moving as rapidly as we had originally intended.

The rest of this section discusses what we did. We first describe the baseline runs that we generated for comparison. The same section presents the mechanism behind our passage retrieval runs. In Section 1.3 we discuss the types of clarification forms that we used and, in Section 1.4, how we used the responses. We outline how we used query and document metadata in Section 1.5 and how it was incorporated into the ranking in Section 1.6. We discuss our results in Section 1.8.

1.2 Baseline Runs

1.2.1 Standard Document Retrieval Baseline

Two of our document retrieval baseline runs (ciirtbas and ciirtdbas) used relevance modeling (method 2) [10] as implemented in Lemur.[1] This relevance model was built using the top 75 terms from the top 50 documents. Ciirtbas uses only the topic title as the query and ciirtdbas uses the topic title and description.

1.2.2 Passage-Inspired Document Retrieval Baseline

The two other baseline runs (ciirtpsgbas and ciirtdpsgbas) re-ranked documents from the standard document retrieval baselines based on the best passage from each document.

Passage retrieval was performed using passage language models [1]. The passages were ranked according to query-likelihood. $P(q|P) = \prod_{i=1}^{n} P(q_i|P)$ where P is a passage. The results were smoothed using interpolation with a collection model and a Dirichlet prior.

Each document was divided into passages of 150 words, with an overlap of 75 words. The best passage for every document in the initial ranked list (ciirtbas and ciirtdbas) was determined. These "top" passages were then ranked by their log-likelihood. In the final list(ciirtpsgbas or ciirtdpsgbas), each passage was replaced with its corresponding document.

1.2.3 Passage Retrieval Baseline

Our final baseline (ciirtp) was a ranked list of passages. Passages were retrieved using the method described in the previous section. However for this run, the top 10 passages from each document were considered, as opposed to only one in the previous case.

1.3 Clarification Forms

1.3.1 Top-term clarification form

The system first employs a simple language modeling approach [13] to retrieve the top 1000 documents, and then constructs a relevance model [10] from those documents. The top 30 terms selected by the relevance model were used for the top-term clarification form. A snapshot of the form is shown in Figure 1. In this form, we ask the LDC annotators to mark the terms that are relevant to the query as "Good", the terms that are non-relevant to the query as "Bad", and leave the terms that they can't judge as "Unknown" which is the default option. The text in the parenthesis after each term is a sample context in which the term appears in the retrieved documents. In addition, we also ask the annotators to suggest if there are any terms other than the ones already shown in the form that they think might occur in the relevant documents.

In the responses we have obtained from LDC on this clarification form, there is an average of 12.2 good terms marked among the 30 terms provided per test topic, and the maximum number of good terms marked for any topic is 22 while the minimum is 3. Table 1 summarizes the term statistics per topic. Only one test topic had suggested terms other than the 30 provided. In our submission runs that made use of this clarification form, we expanded the original query with the good terms (suggested terms are also considered as good terms) and performed retrieval. Retrieval results are discussed in section 1.7.

1.3.2 Other clarification forms (CF)

Besides the top-term clarification form, we also generated other clarification forms, namely cluster-by-size CF, cluster-by-distribution CF, and passage CF.

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

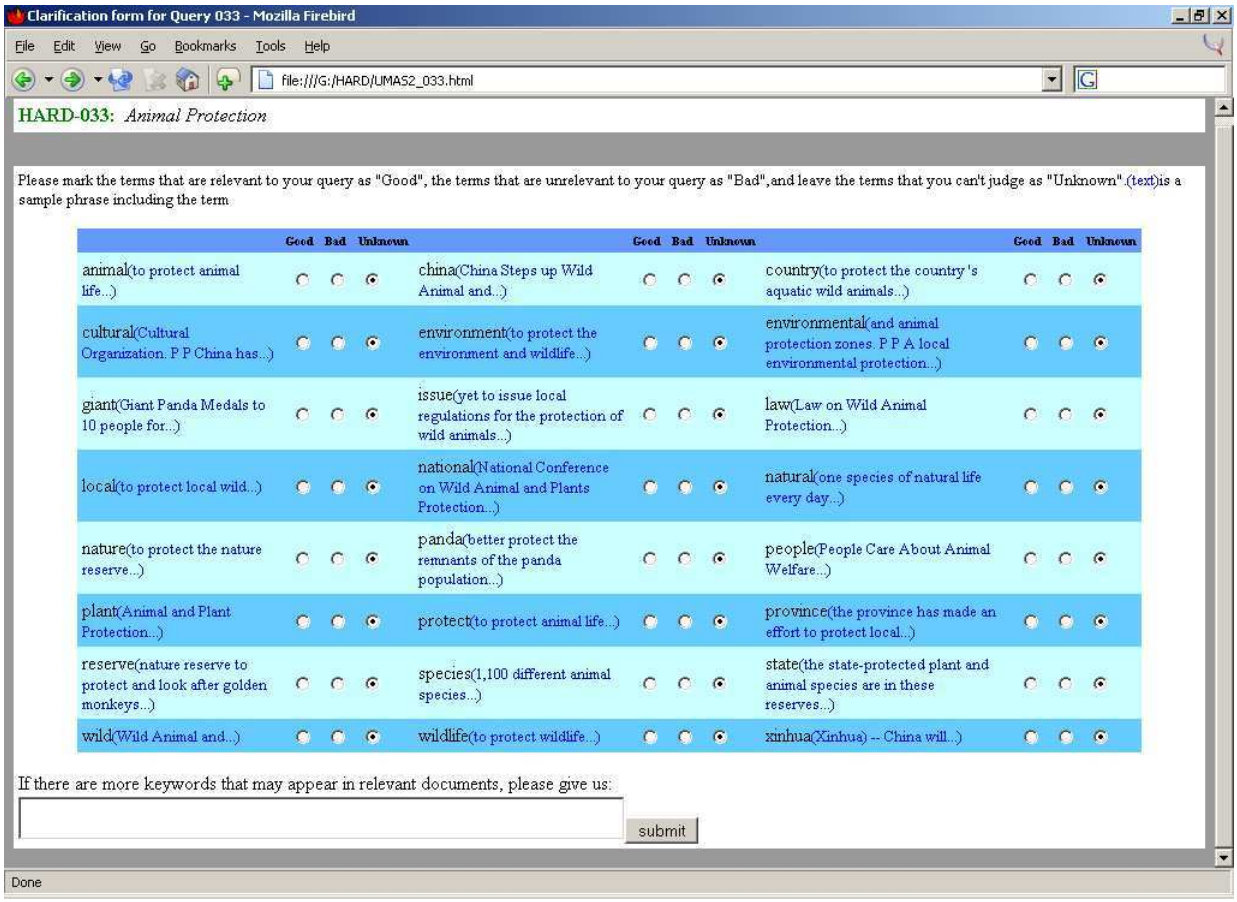

Figure 1: A snapshot of the top-term clarification form for test topic 033.

Table 1: Statistics of the terms that are marked "Good", "Bad", and "Unknown".

	Average	Max.	Min.
Good terms	12.2	22	3
Bad terms	5.4	20	0
Unknown terms	6.4	21	0

- *Cluster-by-size* CF. The system first retrieves the top 200 documents using a simple language modeling approach [13]. The group-average hierarchic clustering algorithm [4] is then applied to the retrieved documents set to obtain clusters. The similarity between documents is determined by the cosine similarity between their term vectors, and the threshold is set to 0.6. Clusters are ranked by their size with the largest cluster at the top of the ranked list. Top 15 clusters are provided, and both the headline of the centroid document and the top 10 terms for each cluster are shown in the CF form. The LDC annotators are asked to mark whether each cluster is good, bad, or unknown. If none of clusters are marked good, the annotators are encouraged to suggest some terms that they think might occur in the relevant documents. In the responses we obtained from LDC on this form, on average, there are 5.5 clusters that are marked "good" , 5.1 marked "bad", and 4.4 marked "unknown", out of the total of 15 clusters provided per test topic. Among the 50 test topics, there are two topics for which all 15 provided clusters are judged "good", four topics for which all 15 clusters are judged "bad", and four topics that have only "unknown" clusters. Only one topic had suggested terms other than the ones displayed in the form. Table 2 shows the statistics on clusters per topic.

Table 2. Statistics of the clusters that are marked "Good", "Bad", and "Unknown".

	Average	Max.	Min.
Good clusters	5.5	15	0
Bad clusters	5.1	15	0
Unknown clusters	4.4	15	0

- *Cluster-by-distribution* CF. This CF is generated in a similar fashion to the *cluster-by-size* CF but with a different cluster ranking mechanism. After the clusters are formed by the group-average clustering algorithm, a simple cluster language model is constructed from the clusters and a query language model is constructed using information from the query. A Kullback-Leibler (KL) divergence score is computed between each cluster model and the query model, and is used as the basis for the ranking of clusters. Clusters and related information are displayed in the same format as the *cluster-by-size* CF.

- *Passage* CF. The system uses the relevance model [10] for retrieval of top 1000 documents, and then segments the documents into passages and ranks the passages. The top 10 passages are presented in the clarification form, again to be marked as "good", "bad", or "unknown" based on their relevance to the query.

However, due to the tight schedule of the LDC annotators, only the top-term CF and the cluster-by-size CF were filled out for all test topics. After comparing retrieval performance on the training data using top-term CF and cluster-by-size CF, we selected the top-term CF to be used in our final submission runs.

1.4 Incorporating Clarification Forms

One of our runs (ciirtcftt) used information gathered from the top-term clarification form. We issued the original query with all of the terms marked "good" added, and with all of the additional suggested terms. In our training experiments, we found that this method actually harmed results, although we did not have enough data to determine whether this could be expected to hold in general. We experimented with several methods for using the terms marked "bad" to improve retrieval, but none produced promising results on the training data.

1.5 Metadata

1.5.1 Statistics on the metadata of the test topics

We summarize in the following tables the statistics on the metadata of the 50 test topics. The first column of each table stands for the type of metadata and its occurring values and the second columns gives the number of test topics with each particular metadata value. Most topics have more than one (often long) snippet of related text.

Purpose	Num. topics
Details	29
Answer	13
Background	8
Any	0

Granularity	Num. topics
Sentence	3
Passage	15
Document	20
Any	12

Genre	Num. topics
Reaction	10
International Reaction	2
Overview	19
Administrative	2
Any	17

Familiarity	Num. topics
1	1
2	25
3	16
4	3
5	0
Unknown	5

Figure 2. Statistics on the metadata of the test topics.

1.5.2 Document Metadata—Annotation and Classification

Metadata annotation. A group of five students was hired to provide relevance judgments and metadata values for the top retrieved documents for the training topics. A web interface was developed for this purpose. An html form was generated for each top-ranked document. It displayed the document and the corresponding training query. The annotators were asked to assign values for the following metadata categories:

Metadata Category	Values
Relevance	Relevant, Related, Non-relevant
Time (for purpose)	Historical, Current, Future, Other
Expertise (for familiarity)	Child, Amateur, Novice, Expert
Granularity	Document, Passage, Sentence, Phrase
Genre	Overview, Administrative, I-Reaction, Other

Some of the above document metadata categories are different from the query metadata categories. It was not clear how to assign values to some of the query metadata categories, when starting from a top-ranked document for a query. Two such categories were "Purpose" and "Familiarity". Instead, we chose metadata categories "Time" and "Expertise" as indicators of "Purpose" and "Familiarity", respectively.

Familiarity mapping. Initially we built a familiarity classifier using support vector machines (SVM) [7], to classify documents into one of the four categories: expert, amateur, novice, and child. However, by using ten-fold cross validation on 629 instances, the classifier only yielded an average accuracy of 71%. To insure the availability of sufficient number of documents for retrieval for each familiarity level, we instead developed the following method.

To determine which document should map to which familiarity metadata value, we compute the readability indices and then map the scores. Readability/reading level describes the ease with which a document can be read or understood [3]. While familiarity and reading level are different, we made an assumption that materials that are more difficult to understand would be more appropriate for a user that is more familiar with a topic. We have computed five readability indices for each document, namely, Dale-Chall [2], FOG [5], Holquist [6], Flesch-Kincaid [8, 9], and SMOG (the Simplified Measure of Gobbledygook) [12]. The statistics of document readability scores across the whole collection is given in table 3. By manually checking how well each of the readability indices is able to differentiate various documents of known levels, we selected SMOG as the final measure.

Table 3. Statistics of document readability scores.

	Min.	Max.	Median	Mean	Std. Dev.
SMOG	3	153.00	10.94	11.99	6.39
Holquist	7.12	5509.0	31.77	47.20	105.64
Dale-Chall	3.69	978.26	4.55	5.48	5.99
Flesch-Kincaid	0	7665.9	9.04	15.95	47.23
FOG	0.03	7867.0	13.03	20.35	48.38

From figure 2, we see that a majority of the test topics specify familiarity values 2 and 3. To insure the quality of retrieval, we made a corresponding mapping that allowed most documents to fall under those two categories. The mapping scheme is shown in table 4. For example, if a document has a SMOG readability score of 9, it is mapped to familiarity=2. If a test topic also specifies the same familiarity value, this document will then be included in the pool of 155,372 documents to perform retrieval for that topic. There are a total of 372,219 documents in the collection out of which 128 documents have no familiarity score because they have no actual contents.

Table 4. Mapping between document SMOG score and familiarity metadata value

Familiarity value	1	2	3	4	5
Range of SMOG score	<=7	(7, 10]	(10, 14]	(14, 19]	>19
Num. of docs mapped	35185	155372	143053	61418	20725

Genre classification. We built a genre classifier using support vector machines. There are total of 615 training documents for this metadata from annotation, out of which 373 were annotated as "overview", 172 as "administrative," 31 as "i-reaction," 21 as "reaction," and 17 as "other." We applied ten-fold cross validation and obtained an average accuracy of 90%. However, the number of training instances is small thus the quality of the classifier can not be reliably determined. When we applied this classifier on the 10 NIST-provided training topics, it was found that 89% of the retrieved documents were classified as "overview" and the remaining 11% were classified as "administrative". There were no instances that were classified as "reaction" or "i-reaction." Because of the potential impact that the classification results might have on the final retrieval, we decided not to use it in our final submission. Instead, we resort to more conservative measures discussed in section 1.6.1.

Purpose classification. A purpose classifier was built, again using SVMs, to classify documents into either "background" or "details," because the third value "answer" can be handled together with the granularity metadata. The documents that were tagged as "current" or "future" for metadata Time used in annotation are considered having a Purpose metadata value of "details", and those tagged "historical" for Time map to "background" for Purpose. Out of the 567 annotated documents for this metadata only 6 were given the value "background." A classifier trained on this data is clearly not reliable. Due to time constraints, we did not use this metadata in our submission runs.

1.6 Incorporating Metadata

1.6.1 Genre

As discussed above, attempts at more sophisticated learning-based methods of assigning genres to documents were unsuccessful so our submitted runs rely on ad hoc rules for re-ranking documents. In the runs that used the genre metadata field (ciirtmda and ciirtmdap), documents were ranked using baseline methodology and then re-ranked according to the following rules:

1. If the query metadata field was **overview**, any document from the Federal Register or Congressional Record was moved five places down the ranked list.

2. If the query metadata field was **reaction**, any document from Xinghua English or the Federal Register was moved five places down the ranked list.

3. If the query metadata field was **i-reaction**, any document *not* from Xinghua English was moved five places down the ranked list.

4. If the query metadata field was **administrative**, any document not from the Congressional Record or Federal Register was moved to the end of the ranked list. Because of the size of the CR and FR document collections, this means that only CR and FR documents were ever returned in the top 1000 documents.

These rules were derived largely from our own experience working with the documents and due to the limited training data, we were generally unable to test their effectiveness.

1.6.2 Familiarity

For all runs incorporating familiarity metadata (ciirtmda and ciirtmdap), we tagged every document with an integer familiarity score in the range 1 to 5 as explained in section 1.5.2. After each document was ranked using the baseline methodology, the rankings were shifted slightly by subtracting a δ value from the rank of each document. We set $\delta = 2 - |(\text{query familiarity score}) - (\text{document familiarity score})|$.

1.7 Granularity

Two of our runs (ciirtmdgp and ciirtmdap) incorporated query granularity by retrieving passages of appropriate size. For queries with Granularity value "Sentence" and " Passage", passages of length 50 words and 150 words, respectively, were retrieved. For the value "Any", ranked lists for passages of size 50 and 150 were merged, after normalization, with the ranked list of documents from ciirtmda. A simple measure of "novelty" was used to reduce overlap between different sized passages from the same document. If a passage or document had more than a 75higher up in the ranked list, it was removed from the ranking. The resulting list was submitted as ciirtmdgp.

The entries in the merged list were reranked using the familiarity and genre information as indicated in the previous section. The "novelty" measure was also applied and the reranked list was submitted as ciirtmdap.

1.7.1 Related Text

Only one of our runs (ciirtrt) used the related text query metadata. Because the related text included relatively large amounts of text (often entire articles), we were concerned that simply appending the related text to the original query might dwarf the query terms. To compensate for this our new query was set to the original query repeated 100 times with the related text appended. Again, we did not have sufficient data to test this model so the number 100 is arbitrary.

1.7.2 Summary of Submitted Runs

We submitted 10 runs in all, including five baseline runs and five other runs that incorporated different combinations of metadata and clarification form data. Table 4 summarizes each of the runs submitted. The first five lines give some information about the baseline runs and the last five lines show what techniques each of the other runs used.

Table 4. Summary of submitted runs.

RUNID	QUERY		METADATA					CLAR. FORM	RETURNS	
	title	descrip.	gran.	purp.	genre	famil.	rltd. text	top terms	docs	psgs.
ciirtbas	*								*	
ciirtdbas	*	*							*	
ciirtpsgbas	*								*	
ciirtdpsgbas	*	*							*	
ciirtp	*									*
ciirtmda	*				*	*			*	
ciirtmdap	*		*		*	*			*	*
ciirtmdgp	*		*						*	*
ciirtrt	*						*		*	
ciirtcftt	*							*	*	

1.8 Results

1.8.1 Document-Level Evaluation

Table 5 shows document-level evaluation results for all ten runs submitted. The top line shows the results for ciirtbas, our best-performing baseline run. Boldface numbers indicate runs that were significantly different (using the Student's t-test at the .05 significance level). Each significance test was performed against the ciirtbas entry in the same column. None of our runs performed significantly better than our ciirtbas baseline. One run, ciirtmda, had better hard average precision and hard precision at 20 documents than the baseline. However, the improvement was not statistically significant.

For 20 of the 48 queries, adding the "good" terms from the clarification form to the original query improved average precision and for 19 of the queries precision at 20 documents retrieved improved. Using genre and familiarity improved the average precision of 16 queries but only improved precision at 20 documents retrieved for two of the queries. We should note here that 4 of the 48 queries had genre of "any" and familiarity "unknown" so they stood no chance of improving when techniques to incorporate genre and familiarity metadata were incorporated. Using the related text metadata to augment the query resulted in better average precision for 16 queries and better precision at 20 documents retrieved for 11 queries. This means that the use of the chosen metadata and our clarification form did help improve retrieval in some cases but not in general.

Advance discussion on the track mailing list led us to believe that only Congressional Record and Federal Register could be considered relevant using HARD relevance evaluation criteria on queries with genre equal to "Administrative." To take advantage of this known fact we designed our system only to retrieve those two types of documents for "Administrative" queries. Interestingly, however, for the two "Administrative" queries in the testing set (HARD-069 and HARD-176), there were several documents from the New York Times and Xinghua English collections that were marked relevant. As a result, our system did worse on both of these queries than it did using the baseline method.

Table 5. Document-level evaluation of all runs. Standard deviations are shown in parenthesis.

RUNID	SOFT AVG. PREC.	HARD AVG. PREC.	SOFT PREC. @ 20	HARD PREC. @ 20
ciirtbas	0.3518 (0.2777)	0.3091 (0.2737)	0.5365 (0.3681)	0.4250 (0.3341)
ciirtdbas	0.3314 (0.2508)	0.2969 (0.2452)	0.5198 (0.3462)	0.4156 (0.3239)
ciirtpsgbas	**0.3052** (0.2371)	**0.2529** (0.2279)	**0.4719** (0.3334)	**0.3438** (0.2822)
ciirtdpsgbas	0.3029 (0.2106)	0.2640 (0.2073)	**0.4708** (0.2988)	**0.3708** (0.2837)
ciirtp	**0.3049** (0.2367)	**0.2526** (0.2275)	**0.4719** (0.3334)	**0.3438** (0.2822)
ciirtmda	0.3500 (0.2811)	0.3136 (0.2815)	0.5344 (0.3689)	0.4260 (0.3361)
ciirtmdap	**0.3056** (0.2540)	0.2682 (0.2497)	**0.5031** (0.3593)	0.3844 (0.3094)
ciirtmdgp	**0.3036** (0.2519)	0.2662 (0.2472)	**0.5031** (0.3593)	0.3844 (0.3094)
ciirtrt	0.3430 (0.2644)	0.3016 (0.2644)	0.5469 (0.3809)	0.4250 (0.3639)
ciirtcftt	0.3146 (0.2669)	0.2942 (0.2668)	0.5448 (0.3900)	0.4469 (0.3810)

1.8.2 Passage-Level Evaluation

Table 6 shows the results of passage-level evaluation for each of the submitted runs. As in Table 5, boldface numbers indicate runs that were significantly different (using the Student's t-test at the .05 significance level). Each significance test was performed against the ciirtdbas entry in the same column.

Results are similar to those in the previous section. None of the runs performed significantly better than the baseline. The baseline run with highest R-Precision is ciirtdbas. Three of the submitted runs, ciirtmdap, ciirtmdgp and ciirtrt, gave higher R-Precision and Passage Precision @ 20 docs than the best baseline run, but the differences were not statistically significant.

Using granularity information to retrieve passages of different size improved R-precision for 12 queries. Of the remaining queries, 19 had the granularity value "Document". The R-precision for these queries was unaffected by incorporating granularity. For 6 other queries, no relevant documents were retrieved in the baseline run, ciirtbas. Since this list was the starting point for passage retrieval, R-Precision remained at 0. For 6 queries, R-precision was adversely affected by the use of granularity values. Therefore, using metadata values helps some queries, but overall, it does not significantly affect performance.

Table 6. Passage-level evaluation of all runs. Standard deviations are shown in parenthesis.

RUNID	R-PRECISION	PASSAGE PREC. @ 20	F-MEASURE
ciirtdbas	0.2301 (0.221)	0.282 (0.3441)	0.209 (0.2061)
ciirtbas	0.2261 (0.2326)	0.2801 (0.3585)	0.2063 (0.2001)
ciirtpsgbas	**0.1853** (0.2076)	0.226 (0.2984)	0.1576 (0.1622)
ciirtdpsgbas	**0.1942** (0.1871)	0.2474 (0.3234)	0.1627 (0.1522)
ciirtp	0.2093 (0.1913)	0.2563 (0.3279)	**0.0929** (0.0878)
ciirtmda	0.2261 (0.2306)	0.2805 (0.3434)	0.211 (0.2102)
ciirtmdap	0.2542 (0.2367)	0.292 (0.3686)	0.1943 (0.2035)
ciirtmdgp	0.2541 (0.2367)	0.2917 (0.3687)	0.1943 (0.2035)
ciirtrt	0.2327 (0.2371)	0.2842 (0.3312)	0.2105 (0.2127)
ciirtcftt	0.2229 (0.2235)	0.3144 (0.4012)	0.1974 (0.2038)

2 Question Answering track

In the QA track, we developed a dynamic passaging retrieval system to identify passages likely to contain answers.

CIIR last participated in the QA task in TREC 9 (2000). At that time we fielded the Marsha system [11]. This system was based on an INQUERY document retrieval engine followed by the application of a series of heuristics rules to identify 250-byte long passages in the retrieved documents that were likely to contain the desired answers.

This year's passage sub-task in the QA track has allowed us to participate once again utilizing our current approach to passage retrieval with language models. We developed a dynamic passaging system that retrieved document passages based on the simplest implementation of language models: cross-entropy between bag-of-word models for a question and a candidate passage.

2.1 Theoretical QA model

Our system used a simple approach to passage retrieval for a question. We constructed a MLE model for the question by treating the words in the question independently, the so-called bag-of-words (BOW) model. A collection-smoothed, with a Dirichlet-prior was used to create a BOW model for a candidate passage. The candidate passage was then ranked by the cross-entropy between its model and the unsmoothed model of the question:

$$H(q|p) = -\sum_{i=1}^{n} P_i(q) \ln P_i(p)$$

where q denotes the question model, p the passage model and n is the number of unique words in the question. This measure is rank-equivalent to the more familiar query-likelihood formula. But we utilize this alternative formulation because it will allow us to build in the future a Bayesian classifier in combination with the use of Relevance Models where the cross-entropy is calculated between a relevance model and the passage model.

2.2 QA System implementation

Passage retrieval was done in two stages. The initial stage consisted of document retrieval using the #sum INQUERY operator. The retrieval used the an index that had INQUERY-stopped and Krovetz-stemmed the AQUAINT collection. Similar stopping and stemming was applied to the questions. We should point out that this step was done solely for the purposes of time savings. As we will comment later on this section, skipping this document retrieval step and going directly to the passage retrieval step produces essentially the same performance.

The top 60 documents (as determined by tuning on TREC 2002 questions) were then selected for the passage retrieval phase. Documents were passaged on-the-fly and sliding windows that moved forward one word at a time while guaranteeing the 250-byte evaluation limit were ranked according to the cross-entropy formula detailed above. Only one-passage per document was allowed to appear on the ranked list.

We tested whether skipping the document retrieval phase would have gained us any performance benefit by looking at the rank-1 measure on the TREC 2002 question set and retrieving 1KB passages. Skipping the document retrieval and doing passage retrieval on the whole AQUAINT collection gave us a rank-1 performance of 38.2%. The two-step retrieval gave us the same rank-1 performance.

The system did no processing to recognize NIL-answer questions so the NIL token was never returned.

2.3 QA Results

The evaluation metric (rank-1 correct) for this year was 20.1%. This compares favourably with our expectations of 19.2% obtained by tuning on the TREC 2002 question set.

3 Conclusion

We participated in the HARD and QA tracks. In both cases, we believe that our results are good, though we had hoped for better. In the case of HARD, we look forward to trying again with a more mature track, based on the lessons learned and with the training data collected this year.

Acknoweldgments

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 Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

References

[1] Corrada-Emmanuel, A., Croft, W.B., and Murdock, V. Answer passage retrieval for question answering. CIIR Technical Report, 2003.

[2] Dale, E. and J. S. Chall. (1948). A formula for predicting readability, *Education Research Bulletin*, 27, 11-20, 37-54.

[3] Gilliland, J., (1972). *Readability*. University of London Press, 1972.

[4] Griths, A., Robinson, L.A., & Willett, P. (1984). Hierarchic agglomerative clustering methods for automatic document classification.*Journal of Documentation*, 40, 175-205.

[5] Gunning, R. (1952). *The Technique of Clear Writing*. McGraw-Hill.

[6] Holquist, J.B. (1968). *A determination of whether the Dale-Chall readability formula may be revised to evaluate more validly the readability of high school science materials*. Ph.D. thesis, Colorado State University.

[7] Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines*. Ph.D dissertation, Dept. of Computer Science, Cornell University. Kluwer.

[8] Kincaid, J.P., Fishburn, R.P., Jr. Rogers, R.L., and Chissom, B.S. (1975). Derivation of new readability formulas for Navy enlisted personnel. *Research Branch Report 8-75*, Naval Air Station Memphis, Millington, Tennessee, 40 pages.

[9] Kincaid, J.P., Aagard, J.A., O'Hara, J.W., and Cottrell, L.K. (1981). Computer readability editing system. *IEEE Transactions on Professional Communications*, Vol. PC-24, No. 1, pp. 12-22.

[10] Lavrenko, V. and Croft, W.B. (2001). Relevance-based language models. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), *Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana (pp.120-127), New York: ACM.

[11] Li, X. and Croft, W.B., Evaluating Question Answering Techniques in Chinese. *Proceedings of the Human Language Technology Conference*, pp. 201-206, 2001.

[12] McLaughlin, H. (1969). SMOG grading - a new readability formula, *Journal of Reading*, 1969, 22, 639-646.

[13] Miller, D., Leek, T., & Schwartz, R. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference*, pp. 214-221.