TREC2001 Question-Answer, Web and Cross Language Experiments using PIRCS

K.L. Kwok, L. Grunfeld, N. Dinstl and M. Chan
Computer Science Department, Queens College, CUNY
Flushing, NY 11367

1 Introduction

We applied our PIRCS system for the Question-Answer, ad-hoc Web retrieval using the 10-GB collection, and the English-Arabic cross language tracks. These are described in Sections 2,3,4 respectively. We also attempted to complete the adaptive filtering experiments with our upgraded programs but found that we did not have sufficient time to do so.

2 Question-Answering (QA) Track

The QA Track requires obtaining 50-byte answer strings to 500 questions (later truncated to 492). The answers are to be retrieved from documents made up from the TREC collections: AP1-3, WSJ1-2, SJMN-3, FT-4, LA-5 and FBIS-5.

2.1 Approach

Our QA system is constructed using methods of classical IR, enhanced with simple heuristics. It does not have natural language understanding capabilities, but employs simple pattern matching and statistics. We view QA as a three-step process: 1) retrieving a set of documents that are highly related to the topic of the question; 2) weighing sentences in this document set that are most likely to answer the question according to the query type and its description; and 3) selecting words from the top-scoring sentences to form the answer string. This approach was quite successful for the 250-byte answer task at TREC-9 [1]. This year we added more heuristics, better pattern recognition and entity recognition.

2.2 Methodology

For the first step, retrieving a set of documents related to the question under focus, we employ both the NIST supplied document list as well as one generated by our PIRCS system. We also use a combination of these two lists that prove to be the best.

For the second step, weighting prospective sentences in the top ranked list of documents, we continue to employ the methods introduced in TREC-9, which are summarized below:

1) Coordinate Matching: counting words in common between the question and a document sentence.
2) Stemming: counting stems as opposed to words in 1). We use Porter's algorithm for stemming.
3) Synonyms: matching based on a manually created dictionary of common synonyms. Its size has increased to 420 terms from 300. It also contains unusual word forms, which are not handled well by stemming. Most of the entries were taken directly from Wordnet
4) RSV: use of the retrieval score of a document from PIRCS to resolve ties for sentences that have the same weight based on word or stem matching.
5) ICTF: use of Inverse Collection Term Frequency to give more credit to less frequently occurring words. For practical reasons, the collection used to obtain the frequencies is the N top retrieved documents.
6) Exact: giving extra credit for matching certain important words which must occur in the answer. At present, these are the superlatives: first, last, best, highest etc. However, one must be careful: 'best' is good but 'seventh best' is not.
7) Proximity: giving extra credit for query words in close proximity in a sentence. They are likely to refer to the same concept as the query. This is done only if all query content words are matched.
8) Heading: giving credit for query words in the headline tag even if they do not occur in a sentence.

9) Phrases: giving extra credit if consecutive words in the query occur in consecutive order in a sentence.
10) Caps: giving extra credit to matching of capitalized query words, assuming they are more important.
11) Quoted: giving extra credit to matching of quoted query words, assuming they are more important.

The query analyzer recognizes a number of specialized query types. 'Who', 'Where' and 'What name' queries are processed by the capitalized answer module, while 'When', 'How many', 'How much' and 'What number' are processed by the numerical answer module.

For 'Name' answers, heuristics were included to identify the following:

a) Persons: capitalized word not preceded by 'the'.
b) Places: capitalized words preceded by 'on', 'in', 'at'. Place names are also recognized by cue words such as 'located', 'next to', 'east of', neighboring, 'borders', etc.
c) Capitalized words: when no other clues are available.
d) Date entities, such as days, months and currency are screened out as incorrect answers.

For 'Numeric' answers, heuristics were included to identify the following:

a) Units: there are classes of queries, which require units. Our system recognizes common units of: length, area, time, speed, currency, temperature and population.
b) Date: there are some queries that have a date or year in the question. We require this date to occur in the sentence or within the Date Tag of a document.
c) Other entities are recognized such as time, address, telephone number, zip codes and percent.
d) Numbers: when no other clues are available.

Selecting a 50-byte answer from the top sentences is quite a challenge as the third step. We used the proximity to query words criterion in most cases, which misses many answers.

We also compiled several lists for countries, states, continents and oceans. We felt it may be useful for the list retrieval task.

2.3 Results and Discussions

Three runs named pir1Qqa{1,2,3} were submitted: pir1Qqa1 utilized the 50 top documents of the PRISE system; pir1Qqa2 used the top 400 subdocuments retrieved by our PIRCS system; pir1Qqa3 combines the two retrievals. PIRCS preprocesses the original documents and returns subdocuments of about 500 words long. Historically, tag information such as heading and (some) date were not captured in our system, which may result in some small degradation in the final score. Table 2.1 compares the submitted runs to the TREC overall median.

As shown in Table 2.1, our best entry pir1Qqa3 scored 0.326, 39% above the TREC median. It also demonstrates that combining retrievals is useful and improves over the results from individual retrievals pir1Qqa1 or pir1Qqa2. A new feature of TREC2001 is that a system might mark as NIL for a query that has no definite answer [2]. Since most correct answers occur at the top positions, a promising strategy is to mark all position 5 answers as NIL. We contemplated doing this but did not do so. The bottom 3 lines of the table show the improvement gained by this NIL strategy.

	All Queries	Compare to TREC	not NIL Queries	NIL Queries
TREC2001	0.234	+0%	0.239	0.193
Official:				
pir1Qqa1	0.300	+28%	0.333	0.000
pir1Qqa2	0.314	+34%	0.348	0.000
pir1Qqa3	0.326	+39%	0.362	0.000
NIL Strategy:				
pir1Qqa1	0.317	+36%	0.330	0.200
pir1Qqa2	0.328	+40%	0.342	0.200
pir1Qqa3	0.340	+45%	0.355	0.200

Table 2.1 QA Results: MRR Values and Comparison with Median

Pir1Qqa3 has 126 questions with rank 1 answers correct, 39 with rank 2, 22 rank 3, 14 rank 4, and 5 rank 5 correct. Since there are 49 questions for which the correct answer is NIL, the aggressive strategy of making every rank 2 answer NIL would do even better!

Question type	Number	Trec Med	pir1Qqa1	pir1Qqa2	pir1Qqa3	pirQqa3 compared to Trec
what	117	0.26	0.36	0.35	0.38	50%
what long	201	0.21	0.28	0.30	0.31	47%
stands for	4	0.42	0.88	0.63	0.75	77%
who	44	0.23	0.27	0.31	0.32	40%
who short	2	0.33	0.00	0.25	0.00	-100%
date	42	0.25	0.32	0.35	0.32	26%
where	26	0.24	0.27	0.24	0.25	7%
population	5	0.15	0.25	0.24	0.25	62%
why	4	0.25	0.25	0.33	0.21	-16%
what unit	29	0.24	0.21	0.24	0.26	7%
unknown	18	0.26	0.28	0.27	0.32	24%

Table 2.2 MRR Performance by question type.

Table 2.2 shows we did well for 'what' questions, both the definition and the longer types, and 'who' questions. The results are not as good for date ('when'), 'what unit' and 'where' type of questions.

The queries may be ranked by the overall performance by all the participants. It is instructive to look at some easy queries that we missed. It happens, that in many cases we retrieved the correct sentences but did not select the correct string. In many cases the correct answer is within the selected answer string, but the other words added (such us names and numbers) make the answer ambiguous.

2.4 Context and List Tasks

A week before the deadline we decided to try the context and list tracks by making minor changes. For the context track, we submitted two runs, pir1Qctx2 and pir1Qctx3. They are essentially the same as our main QA system. pir1Qctx1 (un-submitted) used the PRISE retrieval, pir1Qctx2 used PIRCS retrieval and pir1Qctx3 is a combination as before. The PIRCS retrieval is different in that it combines the series of questions into one query, aiming to retrieve documents that have all or many of the words in the series.

Considering all questions to be independent and evaluate as in main QA, we get the results shown in Fig.2.3. It seems retrieving on all query words for pirQctx2 did not substantially improve the results. Combination of retrievals again proved its usefulness as pir1Qctx3 outperformed its individual retrievals. The context task is an interesting and important task and more intelligence must be crafted into a system to take advantage of the knowledge gained from a succession of previous questions (which we did not do).

We made two changes in the QA system with an eye towards improving performance in the list task. We added a list of countries, states and oceans, and we improved our duplicate answer detection, so that similar forms will be considered equivalent and suppressed. We submitted two runs, pir1Qli1 based on PRISE retrieval and pir1Qli2 based on PIRCS retrieval. There was a bug in the second run output routine that truncated all results to the first word.

	MRR Score	Compare to TREC Med
TREC2001 average	0.298	0%
pir1Qctx1 (unofficial)	0.310	+4%
pir1Qctx2	0.314	+5%
pir1Qctx3	0.329	+10%

Table 2.3: Context Task Results

	> med	= med	< med
pir1Qli1	14[2]	10(1)	1
pir1Qli2	7[1]	11(6)	7(7)

Table 2.4: List Task:. Comparison with Median

Table 2.4 shows the performance of the submitted runs compared with the median of all runs. The un-bracketed values are the actual number better, worse or same as the median; the numbers in square brackets denote best, and the numbers in parenthesis denote worst scores.

3 Web Track

The target collection for the Web track is the W10g disks used last year. We submitted three runs: two for title only queries pir1Wt1 and pir1Wt2, and one for all-section query pir1Wa, which is a long query. Last year [1], we noticed that several queries returned no documents because the query words are common words and screened out by our Zipf threshold. Returning a random set of documents usually is fruitless. This

year, for these 'zero' queries, we did special processing to bring

	← Query Type →		
	Title: pir1Wt1	Title: pir1Wt2	All sections: pir1Wa
Relv.Ret (at most)	2263 0 (3363)	2275 2 (3363)	2284 6 (3363)
Avg.Prec	.1660 0	.1742 5	.1715 12
P@10	.2220 0	.2160 3	.2780 11
P@20	.2070 0	.2110 4	.2370 15
P@30	.2013 0	.2040 8	.2220 18
R.Prec	.1700 0	.1894 5	.1968 9

Table 3.1: Automatic Web Results for 50 Queries

	Query Type		
	Title: pir1Wt1	Title: pir1Wt2	All sections: pir1Wa
	> = <	> = <	> = <
Avg.Prec	24,4 1 25,5	25,4 1 24,5	13,3 2 26,7
prec at 10	13,3 17 20,13	14,4 18 18,14	10,2 17 23,13
prec at 20	22,5 10 17,10	23,6 12 15,11	14,4 10 26,14
prec at 30	21,5 10 19,9	23,7 9 18,10	16,5 10 21,11

Table 3.2: Web Results - Comparison with Median

back words that were screened out due to high frequency, hoping that we might restore some precision value. Documents having these terms within a distance of 5 words in a sentence are considered. For ranking, the minimum distance and the number of such repeats are used, and no second stage retrieval was performed on these queries. This year, there were only 3 such queries (509, 518, 521), but the process was unsuccessful. This is pir1Wt2. For pir1Wt1, we additionally do this process for queries left with one term below threshold. This turns out to depress effectiveness rather than help. Also, we had no spell-check nor punctuation processing, so that queries like #509 ("steroids;what does it do to your body") was not corrected. Query #531 ("Who and whom") contains all stop words and also returns zero precision. Results of our runs are tabulated in Table 3.1 and 2.

The result for pir1Wt2 is about median. Using all sections of a query pir1Wa does not perform better – we suspect there may be some parameters set wrong in our processing. With respect to high precision, Table 3.2, it appears our system perform better at precision 20 & 30 compared to median.

4 Cross Language Track

For Arabic utf-8 coding, the most prevalent two-byte coding is similar to Chinese GB. We think that our Chinese processing can support Arabic with few changes. A student who knows Arabic expressed interest to help us in forming a stopword list and try to find stemming algorithms from the web. A number of such programs were examined, and we eventually discovered that none can process large volumes in reasonable time without drastic re-programming. We also tried to locate an Arabic-English dictionary without success. However, the website for English to Arabic translation (http://tarjin.ajeeb.com) seems useful and good. We had the given English queries translated by using this site. To meet the deadline, we finally decided to use a mixture of n-grams for indexing so that we do not have to rely on linguistic processing. Our representation is to mix 4-gram, 5-gram and single words without stemming or stopword removal.

We submitted four runs two for monolingual Arabic: pirXAtdn and pirXAtd using all sections, and title with description section respectively. The corresponding runs for English-Arabic cross language runs are: pirXEtdn and pirXEtd. Results are tabulated in Table 4.1.

	Query Type			
	Mono tdn	Cross tdn	Mono td	Cross Td
Relv.Ret (at most)	1254 (4122)	899 (4122)	974 (4122)	802 (4122)
Avg.Prec	.1036	.0440	.0852	.0360
P@10	.2440	.1280	.1720	.1040
P@20	.2120	.1220	.1540	.0920
P@30	.2000	.1200	.1520	.0867
R.Prec	.1602	.0768	.1405	.0647

Table 4.1: Automatic Mono andCross Language Results for 25 Queries

The results are way below median. Apparently, there was an error in the retrieval in that no year 2000 documents were returned in our retrieval list. We corrected the error but result still does not materially change. It also seems that we may have some system problem related to LINUX v7 where we ran this experiment. We did not pursue this cross language track further.

5 Conclusion

We continued experimenting with our QA system based on classical IR methods enhanced with simple heuristics for locating good sentences. It achieved above average results. This year we used better pattern and entity recognition. In the future, more heuristics, increased use of knowledge bases, exploring part-of-speech information and more careful query analysis will be needed for further progress. The context and list tasks were also prepared using the same methodology. They also give respectable average. It may be because the average is low, or it may perhaps show that an IR-based system is quite robust although it may be less intelligent.

Our web and cross language results are not up to expectation. For the web track, we did not employ more advanced processing such as collection enrichment, term variety, etc. because of time constraints. This year we transferred these two tasks to work on a Linux-PC platform instead of Solaris-SUN. It is possible that some system error may creep in during processing of the Arabic coding.

Acknowledgments

This work was partly supported by the Space and Naval Warfare Systems Center San Diego, under grant No. N66001-1-8912. We like to thank Khalid Yehia and Peter Deng for helping us in the Arabic processing.

References

[1] Kwok, K.L., Grunfeld, L., Dinstl, N. & Chan, M. TREC-9 cross language, web and question-answering track experiments using PIRCS. In: The Ninth Text Retrieval Conference (TREC-9). NIST SP 500-249, pp.417-426, 2001.

[2] Voorhees, E. Overview of the TREC-9 Question-Answering Track. In: The Ninth Text Retrieval Conference (TREC-9). NIST SP 500-249, pp.71-79, 2001.