Testing an Entity Ranking Function for English Factoid QA

K.L. Kwok and N. Dinstl

Computer Science Department
Queens College, City University of New York

1 Introduction

Much progress has been made in the English question-answering task since it was initiated in TREC-8 and our last participation in TREC-2001. QA is a complex semantics-oriented task, and it is necessary that much linguistic processing, auxiliary resources, and learning steps are needed to come up with adequate performance to the task [1]. Chinese language factoid QA was first introduced during NTCIR-5 and -6 [2,3] and in which we participated. Because Chinese QA was new with little training data and Chinese linguistic tools and resources were not as readily available as in English, we employed a simple answer ranking technique that depends only on surface term usage statistics in questions and document sentences [4,5]. The outcome has been surprisingly satisfactory, achieving results ~80% of the best [2,3]. We were curious how this approach may perform for English factoid questions, comparing with median result for example. The program was modified to support English in this TREC-2007 QA task. As in Chinese, we made little use of linguistic tools or training data, and no auxiliary resources. It can form a basis result from which more sophisticated tools may enhance.

The Sections 2-5 describes our question classification, document processing and retrieval, entity extraction and answer ranker respectively. Section 6 shows our results.

2 Question Classification

Each question was analyzed via a rule-based pattern-matching procedure as to what it wants as an answer, and assigned a question type. This procedure has been used in our English-Chinese CLQA system [4], and further updated based on questions from TREC 2006. This classification procedure provides 8 basic types (person, location, organization, date, time, money, percent and number). Questions needing acronym definition (acrn), or entities with titles (art) (such as artistic entities: titles of books, films, operas, shows, etc.) as answers were also identified. We do not have sub-categories within each class.

Type:	per	loc	org	dat	tim	mony	prct	num	acrn	Art	unk	List	Other
#	71	39	32	61	1	8	4	65	2	8	69	85	70

Table 1: Question Classification

Questions that are not successful for the above 10 categories are assigned an ‘unknown’ type, and these include all the 70 ‘Other’ questions. The 85 ‘List’ questions

got assigned types but are not considered here. Table 1 shows that out of 515-155 = 360 factoid questions, 69 (~19%) failed to be assigned a type. We have not analyzed how accurate the classification is for the other ~81% that a type was assigned.

3 Document Processing & Retrieval

There were two different document streams from which to discover answers for TREC 2007 QA: one from the newswire-oriented AQUAINT-2 collections, and the other from blog data. Only the AQUAINT collection was processed in-house via our PIRCS retrieval system to return the top 100 documents for each question. This set of AQUAINT documents were merged with the NIST-supplied top 50 blog pages for each question. These documents were split into sentence units. A second retrieval was done to rank the top d sentences for each question. We have submitted results with retrieval depth d = 30, 50 and 70 sentences.

QA questions came as sets, each with a 'target' (topic) phrase. We added the topic phrase to each question within a set to form queries for retrieval. Only initial retrieval lists were used.

4 Entity Extraction

The d top document sentences were processed by BBN's IdentiFinder [6] for entity extraction. This system can only provide entities of the first seven types (Section 2). We wrote some pattern-based routines to extract the other three types: numbers, acronym, and entities with titles. We consider the last category as those strings that are enclosed within special symbol pairs such as: {``,"}, {`,'}, {","}, etc. (Care is taken to remove those that occur close to utterance wordings like: 'said', 'saying', etc.). Extracted entities are tagged with their types and sentence source, and form an entity pool. Frequencies of repeated entries are also captured. From this pool, we attempt to extract the one correct answer for each question using a ranking function described below.

5 Ranker for Answer Extraction

Typically, out of the top n (e.g. 50) sentences for each question, there can be a hundred or more different candidate entities. If a question has a type, the number of candidate entities for that type is substantially reduced, perhaps to half a dozen or less. It is still a challenge to pick one candidate as the correct answer. Most QA approaches face this problem, and they may use various evidential clues from syntax, semantics or logic, and sources such as the Wikipedia or WordNet, etc. Our simple ranker employs only surface term usage statistics of the question and top-ranked document sentences.

For each question, all extracted entities are collected into a pool. They are then ranked by the following formula where each factor (except W_w) defaults to 1 when it has no influence [5]:

$W = W_c * W_w * W_f * W_p * W_s$

W_c is a category score. When the question type and answer type agrees (except {art} which is less certain), the highest score Wc1 = 200 is assigned. Lesser values Wc2 = 50 are given when they do not agree but both belong to one of the entity groups: {person, location, organization}, {time, date}, {money, percent, number}; Wc3 = 10 when they belong to {art, unk}; and Wc4 = 2 if answer type is {person, location, organization} but question type is {unk} (i.e. these three types are more prevalent even in the failed types).

W_w concerns whether an answer string appears in the question text or not and has Boolean value 0 or 1. We assume that normally answers do not appear in a question.

W_f is the frequency (f) evidence of a candidate. If it repeatedly occurs and confirms itself in the top-ranked sentences, we give it a higher score of the form: $W_f = 1 + a_f * \log f$, where a_f is a constant (1/3).

W_p is a proximity score for an answer candidate based on the matching question content terms and their positions in a document sentence. For each question content term (word or sequence of adjacent words) present in the sentence, it contributes a score: n/distance_to_candidate where n denotes the number of words in a term: $W_p=1+a_p*\Sigma_{j\varepsilon\{\text{matching terms}\}}\ n_j/\text{distance}_j$, a_p is a constant (1/4), and distances are measured both before or after a candidate.

W_s = is a similarity score based on matching terms between question and sentence as well as the sentence retrieval rank r: $W_s=1+a_s* [\Sigma_{i=1..5}\ m_i*\log(n_i)]] / \log(\text{sentence-length}) / r^2$. m_i counts the number of term matching of length n_i.

When evaluating W_w, W_p, W_s scores, one counts matching between words. In English, although words are well delimited by blanks, there are issues of whether one should keep stop-words or not, use original word spelling or stems, keep their case or use one case. For the purpose of raising matching instances at the expense of precision, we discard stop-words, (Porter) stem all words, and generally use only one case.

We did not attempt to identify questions with NIL answers. Since our ranker provides candidates in order, we answer 'List' questions by providing candidates as answers from maximum score (Wmax) to low until a cut-off W < 0.75*Wmax. For 'Other' questions, we provide additional candidates beside the top ranked answer. These additional ones come from all questions of a set and satisfy the following conditions: they are ranked 2nd or 3rd, has W >= 0.75*Wmax in their respective question, and are unique among answers of a question set. Uniqueness is measured by surface term matching only; we did not use WordNet for example.

6 Results and Discussion

We submitted 3 runs with the following characteristics:

pircT07qa1: retrieval depth d = 50, one case matching, 0.75 cut-off
pircT07qa2: retrieval depth d = 30, one case matching, 0.75 cut-off
pircT07qa3: retrieval depth d = 70, mixed case matching, 0.75 cut-off.

Their factoid only results are tabulated in Table 2 below. They all have quite similar effectiveness. In general, there is a substantial number of inexact answers (X); the number of locally relevant (L) and unsupported (U) answers are fewer. On closer analysis of pircT07qa2 results, it showed that slightly more than ½ of the inexact answers are due to the return of incomplete names rather than full names that have been extracted successfully (such as only the surname of a person, or a popular shortened name of an entity). This we believe can be corrected. It turns out that for the parameters of this run, d=30 provides the best result in the range of 10 to 70.

TREC-2007 (360 questions)	**pircT07qa1 (d=50)**	**pircT07qa2 (d=30)**	**pircT07qa3 (d=70, mixed case)**
Factoid Accuracy	.128	.131	.128
R/L/X/U/W	46/5/19/7/283	47/4/17/7/285	46/5/21/8/280

Table 2: TREC-2007 Factoid Result

Figs.1 shows the pircT07qa2 effectiveness of our ranking function with single factors, and cumulatively. It shows that, with our current set-up, proximity is most effective followed by question classification when used singly. Our question classification accuracy is probably low (81% at best). The cumulative plot shows the synergistic nature of employing multiple factors simultaneously. These effects have been observed previously [4,5] for Chinese QA.

Fig.1: TREC-2007 Factoid QA
Effect of Different/Cummulative Ranking Factors

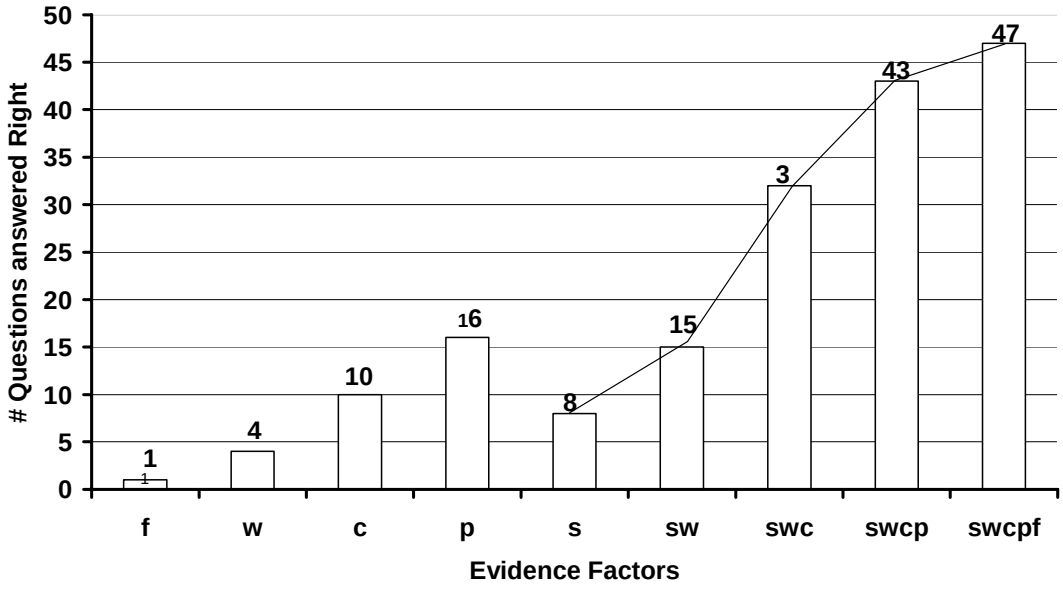
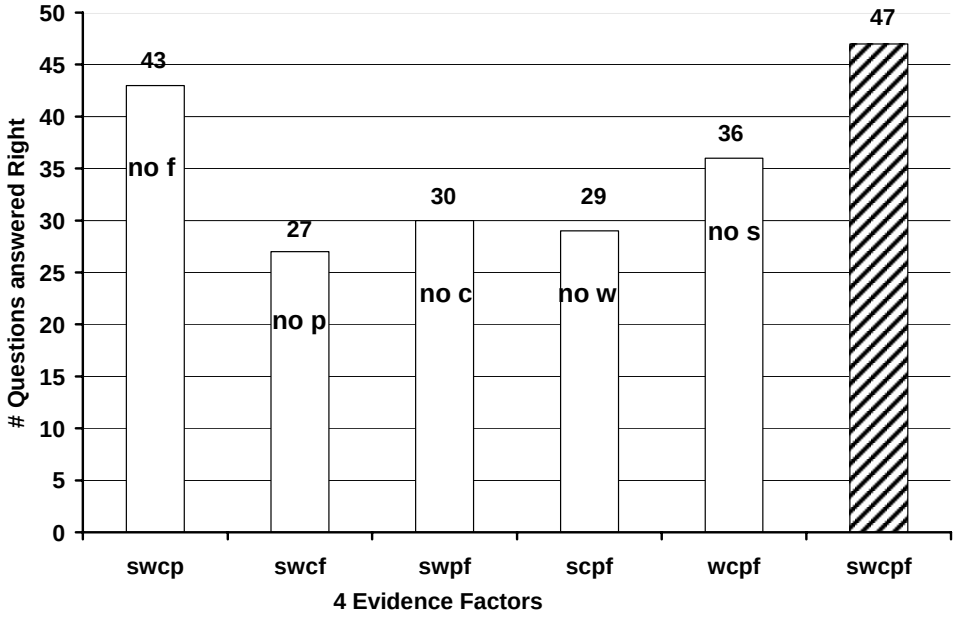


Fig.2 plots the effectiveness when one evidence factor is missing for ranking. It shows how one evidence source contributes in the presence of other factors. Proximity is still the most important and leads to a decrease from 47 right answers to 27 when it is not

used. Surprisingly, the in-question-text factor W_w is as important as the question classification W_c. Apparently, many entities extracted appear also in the question, and this W_w factor helps to eliminate many non-candidates that are ranked high because of other factors.

Fig.2: TREC-2007 Factoid QA
Effect of Missing 1 Feature for Ranking



For TREC-2006 QA task as comparison, our procedure produces the factoid ^results of Table 3. TREC-2006 median factoid accuracy was .181 from all participants. Our simple approach returns a factoid accuracy of only .1315 which is ~27% worse than median. For TREC-2007, the median is at .131 much worse than that of TREC-2006. Our TREC-2006 experiments perform similarly to TREC-2007 and blog data does not seem to have much influence on our simple approach.

TREC-2006 (403 questions)	**pircT06qa1 (d=50)**	**pircT06qa2 (d=30)**	**pircT06qa3 (d=70, mixed case)**
Factoid Accuracy	.132	.127	.124
R/L/X/U/W	53/0/12/5/333	51/0/11/5/336	50/0/11/5/337

Table 3: TREC-2006 Factoid Result

References

1. Dang, H.T, Lin, J & Kelly, D. Overview of the TREC 2006 Question Answering Track. . In: The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings.

NIST Special Publication 500-272, 2007. (http://trec.nist.gov/pubs/trec15/t15_proceedings.html)
2. Sasaki, Y, Chen, H-H, Chen, K-H & Lin, C-J. Overview of the NTCIR-5 Cross-Lingual Question Answering Task (CLQA1). In: Proc of the Fifth NTCIR Workshop Meeting, Tokyo, 2005. pp.175-185. (http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings5/toc.html#CLQA
3. Sasaki, Y, Chen, H-H, Chen, K-H, Lin, C-J.: Overview of the NTCIR-6 Cross-lingual Question Answering (CLQA) Task. In: Proc. of the 6th NTCIR Workshop Meeting. NII, Tokyo (2007) 153-163. (http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings6/NTCIR/ index.html#CLQA)
4. Kwok, K.L. and Deng, P. Chinese Question-Answering: Comparing Monolingual with English-Chinese Cross-Lingual Results. Proc. of Third Asia Information Retrieval Symposium, Singapore 2006. pp.244-257.
5. Kwok, K.L, Deng, P. & Dinstl, N. NTCIR-6 Monolingual Chinese and English-Chinese Cross-Lingual Question Answering Experiments using PIRCS. In: Proc. of the Sixth NTCIR Workshop Meeting. NII, Tokyo (2007) 191-197. (http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings6/NTCIR/ index.html#CLQA)
6. Bikel, D.M, Miller, S, Schwartz, R & Weischedel, R.: A High-Performance Learning Name-Finder. In: Proc. Conference of Applied Natural Language Processing, 1997.