TREC2003 QA at BBN: Answering Definitional Questions

Jinxi Xu, Ana Licuanan and Ralph Weischedel

BBN Technologies

50 Moulton Street

Cambridge, MA 02138

1 INTRODUCTION

In TREC 2003, we focused on definitional questions. For factoid and list questions, we simply re-used our TREC 2002 system with some modifications.

For definitional QA, we adopted a hybrid approach that combines several complementary technology components. Information retrieval (IR) was used to retrieve from the corpus the relevant documents for each question. Various linguistic and extraction tools were used to analyze the retrieved texts and to extract various types of kernel facts from which the answer to the question is generated. These tools include name finding, parsing, co-reference resolution, proposition extraction, relation extraction and extraction of structured patterns. All text analysis functions except structured pattern extraction were carried out by Serif, a state of the art information extraction engine (Ramshaw, et al, 2001) from BBN.

Section 2 summarizes our submission for factoid and list qeustion answering (QA). The rest of the paper focuses on defintional questions. Section 4 concludes this work.

2 FACTOID AND LIST QUESTIONS

The factoid system is the same as our system for TREC 2002 (Xu, et al, 2003), except for a couple of modifications. One modification is to boost the score for answers that occurred multiple times in the corpus. This is similar to previous studies (e.g. Clarke, et al, 2001) that employed redundancy information to improve QA performance. Assuming the occurrences of an answer a are $a_1, a_2, \ldots a_n$, which are ranked by IR score, then the final score for a is

$$score(a_1) + c \sum_{1<i<=n} score(a_i)$$

We set c=0.001.

The other modification is to use additional information to validate answers. Specifically,

- **Validation based on question type**: For questions of the form "What X is ...", the system uses WordNet to verify that the question type is a hypernym of the answer. (For example, "Tagalog" is a valid answer for "What language ..." because "language" is a hyperym of "Tagalog".)
- **Validation of answers for questions looking for a date**: We required certain constraints on date answers based on the form of the question. For example, "When (was|did) ... " questions are likely to refer to a specific date in the past, and a good answer candidate should contain a year. "When is ..." questions may refer to either a specific

date in the past (in which case they should contain a year), or a relative date, such as "the day after X". "What month ..." questions should contain a month, etc.

- **Validation of answers for questions looking for a measurement**: We wrote patterns to detect four common measurement questions: dimension, duration, speed and temperature. Valid candidate answers are expected to have both a quantity and a unit of the appropriate type.

- **Validation of answers for questions looking for an author**: Specific patterns were written to extract "who-wrote" answers from the text.

- **Validation of answers for questions looking for an inventor**: Specific patterns were written to extract "who-invented" answers from the text.

- **Validation based on verb-argument**: We used WordNet to match verbs (for example, "Who killed X?" = "Y shot X"). In addition, we refined the scoring of slot matching to take into account the number of "filler" words that appear in between the question words.

List questions were processed like factoid questions, except that answers were ranked and the list of answers was truncated when the score of an answer drops below 90% of that of the best one.

We submitted three runs, BBN2003A, BBN2003B and BBN2003C. The differences are:

- BBN2003A. The Web was not used in answer finding.

- BBN2003B. For factoid questions, answers were found from both the TREC corpus and the Web. For list questions, it is the same as BBN2003A, except the constant *c* in the score computing formula was increased to 0.1.

- BBN2003C. For factoid questions, it is the same as BBN2003B, except for one difference. If an answer was found from both the TREC corpus and the Web, its score was boosted. For list questions, it is the same as BBN2003B.

Our technique to use the Web for QA is documented in our TREC 2002 work (Xu et al, 2003). Table 1 shows the NIST scores for the three runs. Two observations can be made. First, using the Web improved factoid QA. This is not surprising given previous TREC results by our group as well as by other groups. Compared with our TREC 2002 results, however, the impact of using the Web on QA performance is much smaller, improving accuracy from 0.177 to 0.208, a 3% improvement absolute. In comparison, for TREC 2002, using the Web produced a much larger improvement (10% absolute) in accuracy. More work is needed to determine if the reduced benefit of using the Web is due to changes in the characteristics of the questions or due to the modifications we made to last year's system. Second, using a large *c* significantly improved the performance of the list questions (from 0.087 to 0.097). This indicates that taking advantage of answer redundancy is more crucial for list questions than for factoid questions.

	BBN2003A	BBN2003B	BBN2003C
Factoid	0.177	0.208	0.206
List	0.087	0.097	0.097

Table 1: Scores for factoid and list questions

3 DEFINITIONAL QUESTIONS

3.1 System Overview

Our system processed a definitional question in a number of steps. First, question classification identified the question type, i.e. whether a question is a *who* or a *what* question. The distinction is necessary because some subsequent processing treats the two types of questions differently. Also in this step, the question target was extracted from the question text, by stripping of "What is", "Who is" etc.

Second, information retrieval pulled documents about the question target from the TREC corpus. This was achieved by treating the question target as an IR query. BBN's HMM IR system (Miller, et al, 1999) was used for this purpose. For each question, the top 1000 documents were retrieved.

Third, heuristics were applied to the sentences in the retrieved documents to determine if they mention the question target. Sentences that do not mention the question target were dropped.

Fourth, *kernel facts* that mention the question target were extracted from sentences by a variety of linguistic processing and information extraction tools. A kernel fact is usually a phrase extracted from a sentence. The purpose of using kernel facts is twofold: to minimize irrelevant materials in the answer and to facilitate redundancy detection.

Fifth, all kernel facts were ranked by their type and their similarity to the *profile* of the question. The question profile is a word centroid that models the importance of different words in answering the question.

Finally, heuristics were applied to detect redundant kernel facts. Up to a cap on the total answer length, facts that survived redundancy detection were output as the answer to the question.

3.2 Checking if a Sentence Mentions a Question Target

First, we check if a document mentions a question target at all. If a document does not mention a question target, we drop the whole document from consideration. For *who* questions, we require a document to contain a word sequence "*F…L*" (*F* and *L* are the first and last names of the question target) and the distance between *F* and *L* is less than 3. The purpose is to match "George Bush" with "George Walker Bush". We assume the first and last names are the first and last word of the question target respectively. For *what* questions, we require the document to contain the exact string of the question target, except that plural forms were converted to singular before string comparison.

If a sentence contains a noun phrase that either matches the question target directly (via string comparison) or indirectly (through co-reference), we think it contains the question target. We

used Serif (Ramshaw et al, 2001) for co-ref resolution and parsing. For *who* questions, only the last name was used in string comparison.

3.3 Extraction of Kernel facts

3.3.1 Appositives and Copula Constructions

An example appositive is the phrase "George Bush, the US President". An example copula is the sentence "George Bush is the US President." In both cases, the phrase "the US President" is a definition for "George Bush". Appositives and copulas were extracted from the parse trees of the sentences based on simple rules using Serif.

3.3.2 Propositions

Propositions represent an approximation of predicate-argument structures and take the form: predicate ($role_1$: arg_1, … , $role_n$: arg_n). In the context of this work, the predicate is typically a verb. Arguments can be either an entity or another proposition. The most common roles include logical subject, logical object, and object of a prepositional phrase modifying the predicate. For example, "Smith went to Spain" is represented as went(logical subject: Smith, PP-to: Spain). Propositions were extracted from parse trees using Serif.

We classified propositions into special propositions and ordinary ones. We manually created a list of predicate-argument structures that we thought were particularly important in defining an entity. For example, "<PERSON> was born on <DATE>" is one of the predicate -argument structures for persons. Propositions that matched one of such pre-defined structures were classified as special while others were classified as ordinary.

3.3.3 Structured Patterns

We handcrafted over 40 rules to extract structured patterns that are typically used in defining a term. Similar techniques were also used by Columbia University (Blair-Goldensohn, et al, 2004) and Language Computer Corporation (Harabagiu, et al, 2004) in their TREC 2003 work. For example, one such rule is "<TERM> ,? (is|was)? also? <RB>? called|named|known+as <NP>". Applied to a parsed sentence, the rule will match the question target (<TERM>), optionally followed by a coma, optionally followed by "is" or "was", optionally followed by "also", optionally by an adverb (<RB>), followed by "called", "named" or "known as" and followed by a noun phrase (<NP>). In the pattern, the "?" denotes optional, "+" concatenation, and "|" alternative. If the question is "What are tsunamis?", the pattern will extract the phrase "Tsunamis, also known as tidal waves" from the sentence "Tsunamis, also known as tidal waves, are caused by earthquakes."

3.3.4 Relations

As discussed in Section 3.3.2, propositions simply consist of lexical predicates. Since different lexical predicates can represent the same underlying relation, normalizing these propositions into relations that are commonly found in an ontology, where possible, is obviously desirable for definitional QA.

In this work, relations were extracted by Serif. Serif can extract the 24 binary relations defined in the ACE guidelines (Linguistic Data Consortium, 2002). Using lexicalized patterns, Serif

extracts those relations from the propositions. For example, the relation role/general-staff("Gunter Blobel", "Rockefeller University") will be extracted from the sentence "Dr. Gunter Blobel of The Rockefeller University won the Nobel Prize for medicine today for protein research that shed new light on diseases including cystic fibrosis and early development of kidney stones.".

The QA guidelines require the answer to a definitional question to be a list of textual strings rather relations. We mapped a relation extracted by Serif to a phrase by finding the smallest phrase in the parse tree that contains a mention of the question target and the other argument of the relation. For the above example, the extracted phrase would be "Dr. Gunter Blobel of The Rockefeller University".

3.3.5 Sentences

In addition to the above types of kernel facts, we used full sentences as fall back facts in order to deal with sentences from which none of the above-mentioned types of kernel facts can be extracted.

3.4 Ranking the Kernel Facts

The ranking order of the kernel facts is based on two factors: their type and their similarity to the profile of the question. Appositives and copulas were ranked at the top, then structured patterns, then special propositions, then relations and finally ordinary propositions and sentences. Within each type, kernel facts were ranked based on their similarity to the question profile. The similarity is the tf.idf score where both the kernel fact and the question profile were treated as a bag of words. We used the tf.idf function described by Allan et al, 2000.

The question profile was created in three possible ways. First, we searched for existing definitions of the question target from a number of sources. The resources include: WordNet glossaries, Merriam-Webster dictionary (www.m-w.com), the Columbia Encyclopedia (online at www.bartleby.com), Wikipedia (www.wikipedia.com), the biography dictionary at www.s9.com and Google. To search for biographies on Google, we used the person name and the word "biography" as a query (e.g. "George Bush, biography"). A simple rule-based classifier was used to weed out false hits. If definitions were found from these sources, the centroid (i.e,. vector of words and frequencies) of the retrieved definitions was used as the question profile.

If no definitions were found from the above sources, we considered two options. If the question is a *who* question, we used the centroid of a collection of 17,000 short biographies from www.s9.com as the question profile. Our hope is that using a large number of human created biographies, we can predict what words are important for biography generation. If the question is a *what* question, we used the centriod of all kernel facts about the question target as the question profile. The assumption here is that the most frequently co-occurring words with the question target are the most informative words for answering the question. Similar techniques have been used in definitional QA (Blair-Goldsenshon, et al, 2003) and summarization (Radev, et al, 2000).

3.5 Redundancy Removal

The goal of redundancy removal is to determine if the information in a kernel fact *f* is covered by a set of kernel facts *S* that are already in the answer. Three methods were used to decide if *f* is redundant with respect to *S*:

- If *f* is a proposition, we check if one of the facts in *S* is equivalent to *f*. Two propositions are considered equivalent if they share the same predicate (e.g., verb) and same head noun for each of the arguments. If such a fact is found, *f* is considered redundant.

- If *f* is a structured pattern, we check how many facts in *S* were extracted using the same underlying rule. If two or more facts were found, we consider *f* redundant.

- Otherwise, we check the percentage of content words in *f* that have appeared at least once in the facts in *S*. If the percentage is very high (i.e., >0.7), we consider *f* redundant.

3.6 Results and Discussion

The algorithm to generate an answer for a definitional question is:

1. Set the answer set *S*={}

2. Rank all kernel facts based on their similarity to the question profile regardless of their type. Iterate over all facts: In each iteration, discard a fact if it is redundant with respect to *S*. Otherwise, add the fact to *S*. Go to the next step if *S* has *m* facts.

3. Rank all remaining facts by type (the primary field) and then by similarity (the secondary field). Iterate over the ranked facts: In each iteration, add a fact to *S* if it is not redundant. Go to step 5 if the size (i.e., the number of non-space characters) of *S* is greater than *max_answer_size* or the number of sentences and ordinary propositions in *S* is greater than *n*.

4. If *S* is empty, rank all sentences in the top 1,000 retrieved documents based on the number of shared words between a sentence and the question target. Add the top 20 sentences to *S*.

5. Return *S* as the answer to the question.

We submitted three official runs, BBN2003A, BBN2003B and BBN2003C. For all three runs, *max_answer_size*=4,000 bytes. For BBN2003A, *m*=0 and *n*=5. For BBN2003B, *m*=0 and *n*=20. For BBN2003C, *m*=5 and *n*=10. The parameters were empirically set based on a set of about 79 development questions. Table 2 shows the results of the three runs. Overall, our results are satisfactory, given the median and best scores of all runs provided by NIST. In fact, BBN2003C achieved the highest score for definitional QA at TREC 2003.

Shortly after submitting the official runs and after discussion with NIST, we submitted a baseline run. The goal of the baseline was to give every group (including us) a chance to calibrate the results of their official runs. For each question, the baseline sequentially selected from the top 1,000 documents the sentences that mention the question target. The same heuristic in Section 3.2 was used to check if a sentence mentions the question target. As a fallback, if no sentences were found to mention the question target, all sentences in the top 1,000 documents were selected and ranked by the number of shared words between the question target and the sentences. For answer generation, we iterated over the selected sentences. For each sentence, we checked the percentage of words in the sentence that had occurred in previous sentences in the output answer. If the percentage was greater than 70%, the sentence was considered redundant and was skipped. Otherwise, the sentence was appended to the answer. The iteration continued

until all sentences were considered or the answer length (i.e. the number of non-space characters in the answer) is greater than 4,000. Note that we applied a large length threshold because the F-score favors recall over precision. The baseline run was assessed by NIST in the same way as the official runs.

As shown in Table 2, the baseline performed surprisingly well, with an F-score 0.49. In fact, it outperformed all runs NIST received except BBN2003A, B&C. Our official runs (BBN2003A, B&C) are higher than the baseline, but the improvements are modest. One possible explanation for the unexpectedly good baseline is that the current state of the art of definitional QA is immature. The other is that with $\beta=5$ the F-score is overly recall-oriented and as such was "fooled" by the long answers produced by the baseline.

BBN2003A	BBN2003B	BBN2003C	Baseline
0.521	0.520	0.555	0.49

Table 2: Results for Definitional QA

Table 3 shows the scores for *who* and *what* questions for BBN2003C. The average score for *who* questions is somewhat better that that of *what* questions, but due to the relatively small number of questions, it is hard to determine if the difference is statistically meaningful.

TYPE	Number of questions	NIST F score
Who questions	30	0.577
What questions	20	0.522
Total	50	0.555

Table 3: BBN2003C score breakdown based on types of definitional questions

Figure 1 shows the score distribution over the 50 definitional questions sorted by F score. Quite a few questions (10) get a score of zero or close to zero. An initial analysis shows that a major source of failures is faulty assumptions we made in interpreting the question target. One example is "What is Ph in biology?". Our system assumed the literal string "Ph in biology" is the question target and tried to find it in text. Understandably, it failed. Another example is "Who is Akbar the Great?". Our system assumed the last name is "Great". These problems can be fixed..

Another major source of errors is erroneous redundancy removal. For example, for the question "Who is Ari Fleischer?", the inclusion of the kernel fact "Ari Fleischer, Dole's former spokesman who now works for Bush" in the answer masks the fact "Ari Fleischer, a Bush spokesman". The latter was considered to be redundant because all the words in it appeared in the former one. We hope better redundancy detection strategies will overcome such problems.

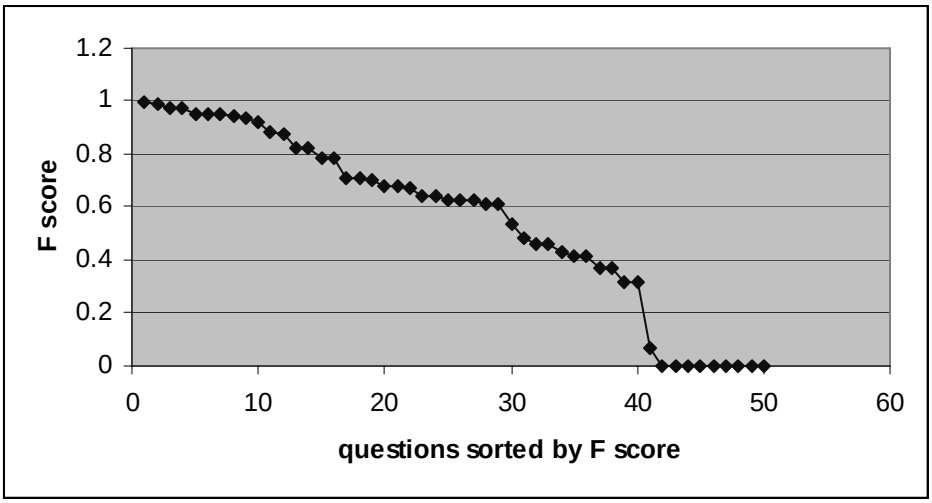

Figure 1: Score distribution over 50 definitional questions for BBN2003C

A question by question analysis shows that when a question obtains a bad F score, it is usually due to recall rather than precision. In fact, for BBN2003C, if we assume perfect precision for all questions, it would merely increase the average F score from 0.555 to 0.614. However, if we assume perfect recall for all questions, it would increase the score to 0.797. This imbalance is understandable because the F-metric used for TREC 2003 QA emphasizes recall by a factor of five over precision.

4 CONCLUSIONS

In TREC 2003 QA, we focused on definitional questions. Our approach combines a number of complementary technologies, including information retrieval and various linguistic and extraction tools (e.g., parsing, proposition recognition, pattern matching and relation extraction) for analyzing text. Our results for definitional questions are excellent compared with the results of other groups. However, much work remains as our results are only modestly better than a baseline that did little more than sentence selection using IR.

References

Allan, J., Callan, J., Feng, F., and Malin, D. 2000. "INQUERY at TREC8." In *TREC8 Proceedings*, Special publication by NIST, 2000.

Blair-Goldensohn, S., McKeown, K., and Schlaikjer, A., 2003. "Answering Definitional Questions: A Hybrid Approach." To appear in Maybury, M., editor, *New Directions in Question Answering*, AAAI Press. Chapter 13, during 2004.

Blair-Goldensohn, S., McKeown, K., and Schlaikjer, A., 2004. "A Hybrid Approach for QA Track Definitional Questions." To appear in *TREC 2003 Proceedings*, Special publication by NIST, 2004.

Clarke, C., Cormack, G., and Lynam, T., 2001. "Exploiting redundancy in question answering." In *Proceedings of SIGIR*, 2001.

Harabagiu, S., Moldovan, D., Clark, C., Bowden, M., Williams, J., and Bensley, J., 2004. "Answer Mining by Combining Extraction Techniques with Abductive Reasoning". To appear in *TREC 2003 Proceedings*, Special publication by NIST, 2004.

Linguistic Data Consortium, 2002. "ACE Phase 2: Information for LDC Annotators", http://www.ldc.upenn.edu/Projects/ACE2/.

Miller, D., Leek, T., and Schwartz, R., 1999. "A hidden markov model information retrieval system." In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

Radev, D., Jing, H., and Budzikowska, M., 2000. "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies." In *ANLP/NAACL Workshop on Summarization*, Seattle, WA, 2000.

Ramshaw, L., Boschee, E., Bratus, S., Miller, S., Stone, R., Weischedel, R., and Zamanian, A., 2001. "Experiments in Multi-Modal Automatic Content Extraction." In *Proceedings of Human Language Technology Conference*, San Diego, CA, March 18-21, 2001.

Xu, J., Licuanan, A., May, J., Miller, S. and Weischedel, R., 2003. "TREC2002 QA at BBN: Answer Selection and Confidence Estimation." In *TREC 2002 Proceedings*, Special publication by NIST, 2003.