IBM's PIQUANT in TREC2003

John Prager, Jennifer Chu-Carroll, Krzysztof Czuba, Christopher Welty, Abraham Ittycheriah, Ruchi Mahindru
{jprager,jencc,kczuba,welty,abei,rkalra}@us.ibm.com
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Introduction

For the most part, the system we used for TREC2003 was a smooth evolution of the one we ran in TREC2002 [Chu-Carroll et al, 2003b]. We continued to use our multi-source and multi-agent architecture. For Factoid questions we used all of our previous answering agents with an additional pattern-based agent, an enhanced answer resolution algorithm, and increased coverage of the Cyc sanity checker. We will devote a portion of this paper to performing a post-mortem of our experiences with Cyc this year. For List questions, which we did not attempt previously, we ran our Factoid system with different parameters. For Definition questions we took an entirely new approach, which we call QA-by-Dossier, and which will be the other focus of this paper. While we think that our system performed reasonably well in this subtask, the NIST evaluation results do not reflect this, raising some questions about the Definition subtask specification and evaluation.

The PIQUANT System

We will briefly describe the PIQUANT system, which is depicted in Figure 1. A fuller description can be found in [Chu-Carroll et al., 2003a] and [Prager et al., 2003]. The processing begins with Question Analysis, which involves deep parsing, named-entity recognition and feature-structure unification. Question Analysis produces a QFrame that contains the required answer type, the question type, query keywords and a simple semantic form. The QFrame is passed to the QPlan generator. A QPlan is in principle a general program which directs the subsequent processing, but is currently little more than a list of one or more of the available agents which are to be run on the QFrame, with results passed to Answer Resolution. Answer Resolution combines the candidate answers from multiple agents and using a voting mechanism and pre-learned weights, generates a final list of answers, with confidences.

PIQUANT Agents

The agents will now be briefly described. The Linguistic Query Agent (LQA) uses our Predictive Annotation techniques ([Prager et al., 2000]) to generate a query which includes the desired answer type as a query term, and searches an index made from a pre-annotated corpus. Our GuruQA search engine returns passages of size 1-3 sentences. The top 10 such hits (an empirically and theoretically determined optimum, see [Prager, 2002]) are then passed to an Answer Selection module, which determines the best answer candidates based primarily on syntactic features. This agent is a general-purpose agent, in the sense that it is designed to find answers for any of the approximately 100 answer types that the QFrame might propose and that our named-entity recognizer can detect.

The Description Agent (DSA) is used primarily for "Who is" and "What is" questions, and looks for syntactic constructions such as appositions and relative clauses that are likely loci of descriptions of people and things. Because there is no specific answer type to prime the query with, this agent tends to have lower recall, but it can find constructions that the LQA cannot.

The Pattern-based Agent (PBA) is similarly motivated to the Description Agent, but uses a much more sophisticated matching algorithm. It is somewhat similar to the pattern-based approach of [Ravichandran and Hovy, 2002], with parse trees being the level of representation at which patterns are matched. It is a

high precision agent that can be used for many question types, but its coverage is currently very low so its recall is also low.

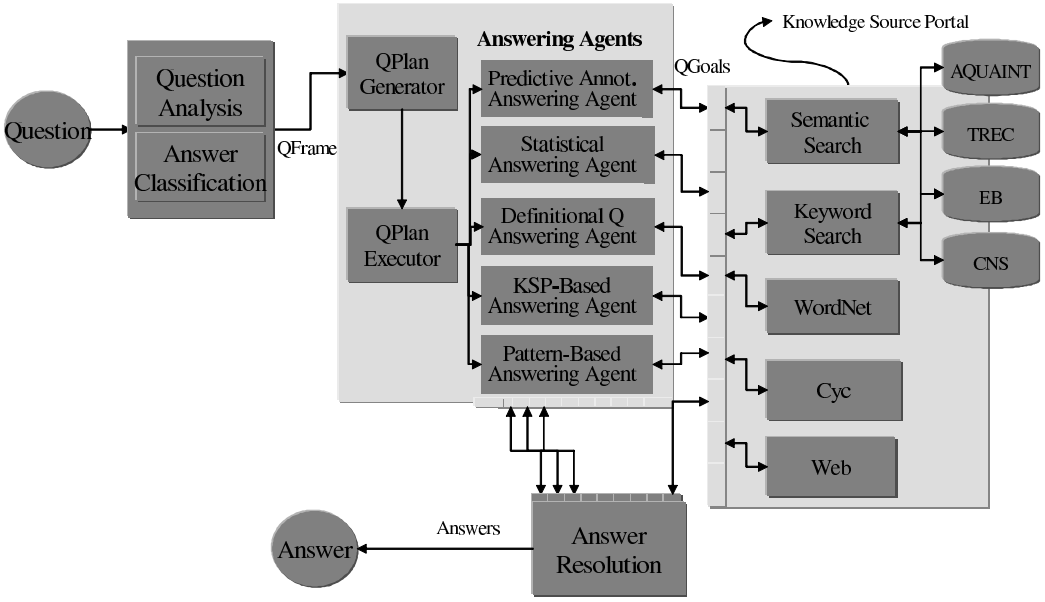

Figure 1 PIQUANT Architecture

The Definition Agent (DFA) employs the Virtual Annotation technique ([Prager et al, 2001] to answer "What is" questions. The focus of the question is looked up in WordNet to find all hypernyms, and the ones that are most likely to co-occur with the question focus in the reference collection, penalized by WordNet path length, are returned. Passages are then retrieved that contain both the question focus and the selected hypernym(s).

The Structured Knowledge Agent (SKA) works in a similar fashion to the Definition Agent. When the QFrame contains a logical form predicate fully expressing the question, the predicate is used as a query through the Knowledge Source Portal, behind which are collections of facts obtained from the Web and elsewhere. When an answer is returned, it is then located in the corpus. Of particular interest and use for Definition questions was its access to data from the Who2 site (http:// www.who2.com) which provided us with biographical information including text snippets that described what a person was *famous-as* and *known-for*.

The Statistical Query Agent (SQA) is another general-purpose agent. We have previously shown ([Chu-Carroll et al., 2003a]) that use of this agent substantially increases our performance, so we used it again this year, for all questions. For TREC2003, the SQA was largely unchanged from last year [Ittycheriah and Roukos, 2002] with the following exceptions:

1. Web pages were retrieved from a popular search engine. Exact answers were extracted from the resulting search results page and the top two web pages as indicated by the search engine. Answers that exceeded a rejection threshold were added to the query for retrieval from the AQUAINT corpus. This improved the precision of this agent: in separate testing the Document MRR of the retrieved set using this agent alone moved from 0.4 to 0.421 by adding these expansion terms while the recall rate remained at 94.75% at Top-1000 documents.

2. The answer selection used a maximum entropy model for chunk selection trained from true sentences of previous evaluations, followed by a maximum entropy chunk ranking model trained on

our system output for 5K questions. This substantially reduced the number of inexact answers compared with our 2002 implementation.

The GuruQA Agent (GQA) is a new agent built specially for our *QA-by-Dossier* methodology, described later. Its function is similar to that of the Linguistic Query Agent, except with no identified answer type, so it can be thought of as a traditional Information Retrieval engine. It is used in conjunction with other facilities, such as the Structured Knowledge Agent, or WordNet, whose glosses contain descriptive phrases or sentences about objects and people. The GuruQA Agent is used to locate instances of these descriptions in a corpus.

Sanity Checking

The integration with the Cyc knowledge base was expanded in a number of ways from our TREC 2002 system. The primary directions we pursued were to expand the coverage of our semantic mapping to Cyc predicates, and to resolve issues with the previous interface regarding the treatment of known answers.

The mapping to Cyc predicates is an important element of our interface to Cyc. Rather than simply running Cyc on any question, we severely restrict the connection to predicates for which we can reliably detect the predicate and focus in questions, and reliably extract answers from passages. For these questions we rely on Cyc's knowledge and reasoning to produce a result within 10 seconds (per candidate answer). These requirements are satisfied by experimentation, thus adding a semantic mapping is not a simple matter of adding terms to a mapping table, and takes some time. In 2002, we were limited to five predicates. For TREC 2003, this was expanded to 35 predicates. The most frequently used predicate in the sanity checker expansion was the *located-in* predicate, which exploits Cyc's excellent coverage of geography for sanity checking "Where" questions. As of TREC-2003, this predicate mapped only to Cyc's geographical containment relation (*inRegion*) and was only used to validate that an answer was correct (i.e. correct answers were given a confidence boost and no answers were thrown away).

The sanity checking API in 2002 had an important problem dealing with the difference between known *ranges* and known *answers* for questions. In our initial formulation of sanity checking, we envisioned using Cyc's knowledge to simply do range checking on candidate answers, and rejecting answers that were outside the range that Cyc knew for that predicate and focus type (e.g., rejecting "200 miles" as a candidate answer for the "height of Mt. Everest", since Cyc knows mountains – *the focus type* – are between 1,000 and 30,000 ft. high). In our post-TREC analysis, we found that in half of the 2002 questions for which the sanity checker was invoked, Cyc actually *knew the answer* to the question (i.e. it knew that the height of Mt. Everest – *the focus* – is 29,200 ft.). In response to this, we expanded the API to include several possible results for each predicate, focus, and candidate answer triple: answer is known to be correct, answer is known to be incorrect, answer is in range, answer is out of range, and unknown.

In our analysis of TREC-2003 performance, the Cyc post-processor had no impact on our factoid QA results. In detailed analysis, we found that it fired correctly on 4% of the questions. In slightly over two-thirds of those cases (10 questions) the correct answer was already being returned, thus the post-processing would only have impacted the confidence in those answers, which was not part of the TREC-2003 evaluation. For the other third (four questions), the sanity checker was throwing out all the answers returned by the Linguistic Query Agent, and thus that agent produced no results and the final answer came from another agent. This latter problem can be fixed by moving sanity checking to post-processing of all agents.

In detailed failure analysis, we found that the sanity checker could have been used on 13% of the questions. The 9% difference was mainly due to the inability of our question analyzer to generate an appropriate semantic form. For example, at the time of TREC-2003, our question analyzer did not recognize "In what city…?" as a located-in question (that is, it did not emit the *located-in* semantic form). Clearly this is a problem that will decrease with time or resources, as question analysis generates a semantic form using a rule-based approach. Of the questions for which the sanity checker could have fired, Cyc had an answer or range for more than half, and of those our system was returning the wrong answer in roughly half the cases. In actual numbers, it is reasonable to project that with a perfect interface, Cyc's existing coverage of

common-sense knowledge could have improved our TREC-2003 score by 2% absolute - an addition 9 questions correct - and improved our confidence (not relevant for TREC-2003) for 30 questions in total.

Our Cyc post-processor remains a proof-of-concept, however, and with significantly more resources placed on generating a semantic form in question analysis and improving Cyc's general coverage, our evaluation indicates this approach would work for as much as 40% of TREC-style factoid questions. However, a critical factor to consider is that our system without Cyc post-processing already performs relatively well on that subset – returning a correct answer for nearly 70% of those questions. This indicates that the significant effort required to get broad coverage would only net at best a 12% absolute improvement in number of correct answers over the system without Cyc.

That limitation is based on our current approach to question analysis, which is intentionally simple – we do not generate a deep semantic analysis of the question. At the moment, for example, the semantic form of a question is only generated for questions that ask for a property of an object, e.g., *"How big is Mars?"* Therefore questions like *"How far away is the moon?"* would not generate a semantic form since the correct form requires a binary predicate: distance(Earth, Moon). More significantly for analyst usage, any kind of temporal qualification would require such an expanded semantic form. It has always been our plan to start with simple semantic analysis and deepen it as necessary. We are yet in the early stages of evaluating how these kinds of improvements in semantic analysis would impact overall performance.

A critical early goal for our use of Cyc was exploitation of its reasoning capabilities. Within the knowledge representation world, Cyc takes a particular approach to representation and reasoning that ignores tractability and computational problems and uses a language (CycL) and a representation style with maximum expressiveness. This proved to be a significant barrier in practice to utilizing Cyc in our system. Early experiments would run for several days on a single question.

Cyc's ready answer to problems like this are micro-theories – basically modules of knowledge within which a reasoning task can be bounded to a significantly smaller amount of information. We performed some simple experiments to test whether this was the case and could be used by our system. We began with the question, *"How large is the everglades?"* Cyc does not know the answer to this question, however it knows that the Everglades is in Florida, it knows the size of Florida, and it has a common-sense rule expressing the constraint that a spatial region cannot exceed the size of any spatial region it is contained in (e.g. the Everglades cannot be larger than Florida). This should give us knowledge to throw away any proposed answer that is larger than the size of Florida.

Without using micro-theories, a single query for size(Everglades, *n*) took on the order of two days to produce a true/false result. Given that our system performs a query for each candidate answer, and receives on the order of 10 candidate answers from each of five answering agents, that performance is not even close to being acceptable. Limiting the query to the US Geography micro-theory, however, allows answers to come back in 2-5 seconds (two if the answer is provably true or false, and five if not).

This seems like a good result, however we were not able to find any general way to map from questions to the appropriate micro-theory; there is no "meta knowledge" in a sense that tells us, for a question we have not yet seen, what the best micro-theory is. What information from the question, and what information from a micro-theory, could impact micro-theory selection? After several days trying to make sense of the micro-theory structure of Cyc, we found it to be either inscrutable or completely unprincipled and arbitrary. We found micro-theories that appeared to contain things that were true during a particular year, things that were true in some fictional world, information relevant to a particular project, information culled from a particular source, information in some domain of interest, information that didn't seem to belong anywhere else yet, etc.

In the actually TREC-2003 system, therefore, the only reasoning used was "inheritance" of range constraints down the generalization hierarchy.

NIL Processing

To address No-Answer questions, we used a two-pronged approach. We had developed in training a confidence threshold of 0.26; questions whose confidence fell below this were classified as NIL. In addition to the threshold, for those questions for which the SKA returned an internal candidate answer, if the answer found in the corpus itself was not the same, then our system classified the question as NIL. Our performance was 7/85 precision and 7/30 recall.

We parlayed our NIL threshold logic over to List questions. There, our system generated as many candidate answers as it could of the sought type in the top 40 documents, and all those below a threshold of 0.3 were rejected.

Performance

We submitted three runs which differ primarily in the use of the SQA agent. The *IBM2003a* run used the SQA agent for Factoid questions only; *IBM2003b* used SQA for all question types, but only as support for List (it could not propose answers); and *IBM2003c* included SQA as a primary agent for all question types. In addition, other internal parameters were adjusted for *IBM2003c* to favor recall. Our scores were as in the following table. The difference in scores between runs *b* and *c* for the Definition task is entirely due to inconsistent judging since the submissions were identical but were assessed differently for 6 questions. Since we did best in IBM2003c, we will use that run as the basis for subsequent discussions.

Run	Factoid	List	Definition	Overall
IBM2003a	.298	.070	.124	.197
IBM2003b	.298	.065	.177	.210
IBM2003c	.298	.077	.175	.212

Definition Questions

The Definition task required "Who is X?" and "What is X?" questions to be answered by a list of facts or *nuggets* describing X. The guidelines did not specify what kind of facts should be included in these lists, nor how "atomic" each fact needed to be. This effectively provided a framework, but not a precise specification. To guide our effort to come up with an implementation we interpreted the framework as follows. We determined that obituaries and short encyclopedia articles were in effect answers to "Who is X?" questions, and it seemed to us that the Definition task therefore could be viewed as the gathering of the raw information that would go into such articles. Organizations and objects could be similarly described.

QA-by-Dossier

QA-by-Dossier (QbD) is a new technique which we used for the first time for Definition questions in TREC2003. It was developed last year under the ARDA AQUAINT program. The essence of the approach is that the original question is not necessarily asked directly but instead a number of *auxiliary questions* are asked instead. In doing so, the entire PIQUANT system is called recursively. The answers to the auxiliary questions are assembled into a *dossier* and returned to the user. QbD was co-developed with a more sophisticated counterpart, QA-by-Dossier-with-Constraints (QDC), in which answers to the auxiliary questions (plus possibly some others asked just for this purpose) are checked for consistency with each other. QDC is described in [Prager et al., forthcoming], but was not ready to be used for TREC2003.

The fact-gathering by QbD seemed to us to align very nicely with the requirements of the Definition task. Specific factoid questions could be formulated to find the information that is "always" present in definitional articles, and more open-ended techniques, such as our Description Agent uses, could attempt to find unanticipated facts.

Our Approach

The QA-by-Dossier approach has been adapted for TREC2003 to answer three types of definition questions where the question focus is a PERSON, an ORGANIZATION, or a THING. This classification is

performed by our question analysis module, and a different set of auxiliary questions is issued for each question type. Our auxiliary question set may in principle contain multiple rounds of follow-up questions in which different subsequent questions may be issued based on answers to earlier questions. The most obvious application of this idea is to ask profession-dependent questions after the occupation of a person is discovered. For example, if the person is a composer, then one might ask what music he has written; if he is a scientist or engineer, then what he has discovered or invented; if he is a politician, then with whom has he had an affair. To support this functionality of automatically determining focus-dependent follow-up questions, we needed a mapping from occupation name to characteristic activity (i.e. verb), but we did not have this ready either for TREC2003.

The auxiliary question sets we developed therefore consisted of two kinds of questions. The first kind is the general life-cycle type of question that should be applicable to all subjects. For the second, because we did not have the ability to automatically determine follow-up questions, we decided to also ask what would be reasonably general follow-up questions, on the assumption that if the question was wholly inapplicable, no high-confidence answers would be returned and our rejection threshold would take care of eliminating any weak answers that were found.

The task instructions gave no guidance as to what kind of information would be required and/or acceptable for Definition questions. It was stated that a list of facts or properties were desired, unlike earlier year's QA-tracks which sought short noun phrases, but did not attempt to classify these facts. The Definition Question Pilot run under the ARDA/AQUAINT program was not informative for this task. Therefore, to answer PERSON-type Definition questions, we made an informal survey of obituaries and encyclopedia articles in an attempt to determine what information was considered important to give in these pieces, which were in effect implicit answers to "Who was X?" questions. We found that a common element was the set of major events in a person's life-cycle (birth, college, marriage, death), which we could clearly seek, plus some specialized facts that we hoped our general methods would locate. These articles did not typically mention single incidents in the people's lives, unless they had historical significance. Based on this analysis, we manually derived auxiliary question sets for each question type.

For PERSON-type Definition questions of type "Who is/was X?", we asked the following:

No.	Question	Agents	Answers	Threshold
P1	When was X born?	LQA	2	.3
P2	Where was X born?	LQA	1	.3
P3	When did X die?	LQA	2	.3
P4	How did X die?	PBA	variable	
P5	Who was X married to?	LQA	1	.3
P6	What occupation did X have?	LQA	1	.3
P7	What did X do?	PBA	variable	
P8	What did X invent?	LQA	1	.24
P9	What did X discover?	LQA	1	.24
P10	What did X win?	LQA	1	.3
P11	Who is X?	LQA & DSA	5	.3
P12	What compositions did X have?	LQA	1	.3
P13	X ,, <famous-activity>	SKA then GQA	variable	.3
P14	X ,, <known-for>	SKA then GQA	variable	.3

With multiple agents firing on multiple questions, we needed some criteria for deciding what to return as a final answer to the question. We generally used thresholds established in training and returned the best answer to each sub-question, as long as it beat the threshold. We noted in training that for questions #P1 and #P3 we often got the wrong answer in first place but correct in second place, so for these questions we returned the top two answers. This seemed to be a useful strategy knowing that the precision calculation gave an average allowance of 100 bytes per answer, and it took only about a dozen bytes to return each answer to #P1 and #P3.

The answers column indicates how many answers were returned from the other agents, assuming their confidence passed the indicated threshold. The PBA had its own internal threshold and returned a variable number of answers (including zero), all of which were accepted. When the SKA was used, it would find a variable number of text snippets; all of these that the GQA could locate in the TREC corpus with a confidence above the given threshold were accepted. The SQA was used for all questions for all types, and it had its own threshold (0.1).

We omitted questions about a person's education because our training found our system to be unreliable at such tasks. The last three P-questions need some explanation. The *compositions* question #P12 triggers our named-entity type COMPOS that is used for all kinds of titled works – books, films, poems, music, even physical artifacts such as Betsy Ross's "Stars and Stripes" or Lindbergh's "Spirit of St. Louis". Our named-entity recognizer has rules to detect compositions by phrases that are in apposition to "the film …" or the "the book …" etc., but by default captures any short phrase in quotes beginning with a capital letter. The particular phrasing we used in question #P12 does not commit us to a particular creative verb.

The final questions #P13-P14 are ultimately plain IR queries. The system uses the Structured Knowledge Agent's Who2 data, which has short descriptions of famous people, to find brief descriptive phrases that are then combined with the question subject into a bag-of-words which is used as a query against the TREC corpus, using a window-size of one sentence.

For THING questions, we asked the following:

No.	Question	Agents	Answers	Threshold
T1	What is X?	DSA, DFA	1	.15
T2	What is another name for X?	DFA	1	.15
T3	X „ <WordNet gloss entry>	SKA then GQA	variable	.15

And for ORGANIZATION questions

No.	Question	Agents	Answers	Threshold
O1	What does X manufacture?	LQA	1	.15
O2	Where are the headquarters of X?	LQA	1	.15
O3	Who is the CEO of X?	LQA	1	.15
O4	What did X invent?	LQA	1	.15
O5	What did X discover?	LQA	1	.15
O6	What does X do?	PBA	variable	.15
O7	What is X?	DSA	variable	.3

Answer Format

The required answer format was left undefined in the task guidelines. It was not clear whether, if the question was "Who was Leonardo da Vinci?", for example, "the Mona Lisa" alone would be an acceptable answer or if the fact that Leonardo painted it would have to be indicated. Arguing for being more inclusive was the sense that incompleteness might hurt, especially as we had no idea what the assessors would consider an atomic fact. Arguing against was the extra chance of including some incorrect material, along with a potential length penalty. Since generation was allowed, we decided that whenever we knew the relationship between the subject and the answer we would provide it, but as briefly as possible. It would be in the form "relationship: answer", for example, "death: 1066". For compositions we used the non-committal "work". This approach is in line with the "exact answer" requirement for Factoid questions, but was not required here. In fact we think this approach hurt us, since an informal analysis of different system's answer formats and scores, along with our experience with the ARDA AQUAINT *Definition Pilot* exercise, leads us now to think that longer, clause- or sentence-length answers are psychologically more appealing to assessors, even if the information content is the same.

Definition Task Performance

We now present our answers to some of the Definition questions to illustrate the foregoing discussion. We show the answers to #1907 "Who is Alberto Tomba, #1933 "Who was Vlad the Impaler?", #1957 "What are fractals?" and #2201 "What is Bollywood?". We tag each answer with the subquestion/agent that was responsible for it. These charts are only an approximation since in some cases multiple agents proposed an answer.

Returned nugget	Agents	Sub-Question
known as `` La Bomba ," (the bomb) for his explosive skiing style	DSA	P11
work: La Bomba	LQA	P12
most successful and popular Italian skier ever	DSA	P11
personal coach	SQA	
star	SQA	
the most famous ski racer of all time	DSA	P11
vaulting him from seventh to fourth with 1:41.48	DSA	P11
born: Alberto Tomba , Italy	LQA	P2
lawyer	SQA	
job: champion	LQA	P6
work: Slalom for Peace	LQA	P12
born: Italy	LQA	P2
work: the Bomba	LQA	P12
some World Cup skiers	LQA	P11

Results for #1907 "Who is Alberto Tomba?"

Returned nugget	Agents	Sub-Question
job: prince	LQA	P6
Dracula	SQA	
Bram Stoker's main character	LQA	P11
his victims on spikes	LQA	P11
main character was inspired by	DSA	P11
Bram Stoker	SQA	
Some historians	LQA	P11
the Romanian prince	LQA	P11
Ivan the Terrible	SQA/DSA	

Results for #1933 "Who was Vlad the Impaler?"

Returned nugget	Agents	Sub-Question
Fractal geometry is a field of mathematics founded in 1975 by Dr. Benoit Mandelbrot .	SKA then GQA	T3
Endlessly repeated fractal patterns	DFA	T1
is: patterns	DFA	T1

Results for #1957 "What are Fractals?"

Returned nugget	Agents	Sub-Question
based film industry , known as	DSA	O7
churns out nearly 200 feature films in Hindi and other Indian languages every	DSA	O7
HQ: Bombay	LQA	O2
derived	SQA	
the nickname given to Bombay	DSA	O7
movie industry	DSA	O7
more	DSA	O7
makes: capital	LQA	O1
discover: backseat	LQA	O5
invent: backseat	LQA	O4
ceo: Benjamin Gaon	LQA	O3

Results for #2201 "What is Bollywood?"

Discussion of Definition Task Results

Despite the number of obviously incomplete and completely wrong answers, there was certainly useful information returned in the examples shown above. As a rough approximation, we estimated about 2-3 nuggets per example shown – in some cases possibly more depending on the fact granularity assumed by the assessors. As it happens, we scored a total of ***zero*** for the shown examples. Over all the 50 Definition questions, we scored an average recall of .174, an average precision of .325, and an average F of .175.

While our answers were aligned with our interpretation of the NIST evaluation framework, our scores indicate they were not aligned with NIST's interpretation. Upon a subjective consideration of our results, however, we believe, it is obvious that the nuggets PIQUANT produced, often provided suitable information in response to the definition questions (e.g., Alberto Tomba certainly was a famous ski racer). Based on the evaluation framework provided by NIST, we argue that it is possible to come up with actual evaluation guidelines that conform to this framework but produce drastically different outcomes.[1] Although our analysis shows that our own divergence from NIST in interpretation played as big a role in the final score for this subtask as our system's errors, we believe less potential for such variability in the interpretation of the evaluation framework for definition questions would better serve the TREC QA track.

Summary

Our analysis of the contribution of Cyc this year showed that the major limiting factor is still in the area of coverage. Major manual effort is required both to generate appropriate semantic forms and to map to Cyc's predicates, and also to add instance information into Cyc. With the current state of the system, Cyc helps more to improve our answer confidences (not a part of the evaluation this year) than to get answers right.

The major novelty in our system this year was the implementation of QA-by-Dossier to answer Definition questions. Here, a collection of predetermined factoid questions are asked about the subject in order to gather facts that seem to be typically mentioned in definitional articles in newspapers and reference works. An advantage of this method over others which locate definitional syntactic constructs is that our system "knows" the nature of the relationship of the retrieved item to the subject. In the evaluation, we felt that our system had performed relatively well according to our expectations of what was required, but we were very disappointed to find that the NIST assessors had different opinions regarding acceptable answers.

[1] To illustrate this point, we developed a set of evaluation guidelines based on our interpretation of the framework put forth by NIST. These guidelines (admittedly) coincide with the principles we used in developing the QA-by-Dossier agent used in answering definition questions. Precision was calculated using the 100-byte-per-nugget allowance, following the TREC 2003 formula. Recall was approximated by pooling the NIST assessors' nuggets with additional facts found by our system. The self-assessed averages were .385 for recall, .583 for precision and .387 for F, significantly different from the NIST-assessed scores.

The task guidelines provided a framework for answering Definition questions and their subsequent evaluation, but both were left open to some interpretation. We made the assumptions that any correct fact found about the subject of the question would be at least "okay", and that all facts that are typically reported in obituaries and short encyclopedia articles would, by our definition, be "vital". The NIST assessors came up with a different instantiation of the framework for their evaluation task – and to this day we do not understand exactly what that was – and our results have shown that these different instantiations give rise to significantly different scores for the same answer set.

In this paper, we have provided our own detailed evaluation criteria for three types of Definition questions, which we hope will generate useful discussions in outlining more specific evaluation guidelines for the next TREC QA track. Similarly to the Factoid subtask, we believe the perceived output quality of a question answering system on the Definition subtask strongly depends on the user model and expectations. This has proved difficult to capture in evaluation, and this year the difficulty was compounded by the introduction of the concept of "vital" nuggets. Given our own interpretation presented above that was consistent with the initial guidelines, we believe that the concept is not well-defined and simply dropping it in favor of less restrictive judging to allow for more "okay" nuggets (based on pooling) would have made the evaluation more realistic and less controversial.

Acknowledgments

This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number MDA904-01-C-0988.

References

J. Chu-Carroll, K. Czuba, J. Prager, A. Ittycheriah. 2003a. In Question Answering, Two Heads are Better than One. *Proceedings of the Human Language Technologies Conference*. Pages 24-31.

J. Chu-Carroll, J. Prager, C. Welty, K. Czuba, D. Ferrucci. 2003b. A Multi-Strategy and Multi-Source Approach to Question Answering. *Proceedings of TREC2002*. Pages 281-288.

A. Ittycheriah and S. Roukos, "IBM's Statistical Question Answering System - TREC-11", The Eleventh Text REtrieval Conference Proceedings, Trec 2002, pp. 273-280, 2002.

J.M. Prager, E.W. Brown, A. Coden, and D. Radev. "Question-Answering by Predictive Annotation". Proceedings of SIGIR 2000, pp. 184-191, Athens, Greece.

J. Prager, D. Radev, K. Czuba. 2001. "Answering what-is questions by virtual annotation." *Proceedings of Human Language Technologies Conference*. Pages 26-30.

J.M. Prager. "In Question-Answering, Hit-List Size Matters", IBM T.J. Watson Research Center Research Report #RC22297, Jan 2002.

J. Prager, J. Chu-Carroll, E. Brown, K. Czuba. 2003. Question answering using predictive annotation. In *Advances in Question Answering*, forthcoming.

J.M. Prager, J. Chu-Carroll and K. Czuba, "A Multi-Agent Approach to using Redundancy and Reinforcement in Question Answering" in *New Directions in Question-Answering*, Maybury, M. (Ed.), AAAI Press, forthcoming.

D. Ravichandran, E. Hovy. 2002. Learning surface text patterns for a question answering system. *Proceedings of the 40th Annual Meeting of the ACL*. Pages 41-47.