Question Answering using Constraint Satisfaction: QA-by-Dossier-with-Constraints

John Prager
T.J. Watson Research Ctr.
Yorktown Heights
N.Y. 10598
jprager@us.ibm.com

Jennifer Chu-Carroll
T.J. Watson Research Ctr.
Yorktown Heights
N.Y. 10598
jencc@us.ibm.com

Krzysztof Czuba
T.J. Watson Research Ctr.
Yorktown Heights
N.Y. 10598
kczuba@us.ibm.com

Abstract

QA-by-Dossier-with-Constraints is a new approach to Question Answering whereby candidate answers' confidences are adjusted by asking auxiliary questions whose answers constrain the original answers. These constraints emerge naturally from the domain of interest, and enable application of real-world knowledge to QA. We show that our approach significantly improves system performance (75% relative improvement in F-measure on select question types) and can create a "dossier" of information about the subject matter in the original question.

1 Introduction

Traditionally, Question Answering (QA) has drawn on the fields of Information Retrieval, Natural Language Processing (NLP), Ontologies, Data Bases and Logical Inference, although it is at heart a problem of NLP. These fields have been used to supply the technology with which QA components have been built. We present here a new methodology which attempts to use QA holistically, along with constraint satisfaction, to better answer questions, without requiring any advances in the underlying fields.

Because NLP is still very much an error-prone process, QA systems make many mistakes; accordingly, a variety of methods have been developed to boost the accuracy of their answers. Such methods include redundancy (getting the same answer from multiple documents, sources, or algorithms), deep parsing of questions and texts (hence improving the accuracy of confidence measures), inferencing (proving the answer from information in texts plus background knowledge) and sanity-checking (verifying that answers are consistent with known facts). To our knowledge, however, no QA system deliberately asks additional questions in order to derive constraints on the answers to the original questions.

We have found empirically that when our own QA system's (Prager et al., 2000; Chu-Carroll et al., 2003) top answer is wrong, the correct answer is often present later in the ranked answer list. In other words, the correct answer is in the passages retrieved by the search engine, but the system was unable to sufficiently promote the correct answer and/or deprecate the incorrect ones. Our new approach of QA-by-Dossier-with-Constraints (QDC) uses the answers to additional questions to provide more information that can be used in ranking candidate answers to the original question. These *auxiliary questions* are selected such that natural constraints exist among the set of correct answers. After issuing both the original question and auxiliary questions, the system evaluates all possible combinations of the candidate answers and scores them by a simple function of both the answers' intrinsic confidences, and how well the combination satisfies the aforementioned constraints. Thus we hope to improve the accuracy of an essentially NLP task by making an end-run around some of the more difficult problems in the field.

We describe QDC and experiments to evaluate its effectiveness. Our results show that on our test set, substantial improvement is achieved by using constraints, compared with our baseline system, using standard evaluation metrics.

2 Related Work

Logic and inferencing have been a part of Question-Answering since its earliest days. The first such systems employed natural-language interfaces to expert systems, e.g. SHRDLU (Winograd, 1972), or to databases e.g. LUNAR (Woods, 1973) and

LIFER/LADDER (Hendrix et al. 1977). CHAT-80 (Warren & Pereira, 1982) was a DCG-based NL-query system about world geography, entirely in Prolog. In these systems, the NL question is transformed into a semantic form, which is then processed further; the overall architecture and system operation is very different from today's systems, however, primarily in that there is no text corpus to process.

Inferencing is used in at least two of the more visible systems of the present day. The LCC system (Moldovan & Rus, 2001) uses a *Logic Prover* to establish the connection between a candidate answer passage and the question. Text terms are converted to logical forms, and the question is treated as a goal which is "proven", with real-world knowledge being provided by Extended WordNet. The IBM system PIQUANT (Chu-Carroll et al., 2003) uses Cyc (Lenat, 1995) in answer verification. Cyc can in some cases confirm or reject candidate answers based on its own store of instance information; in other cases, primarily of a numerical nature, Cyc can confirm whether candidates are within a reasonable range established for their subtype.

At a more abstract level, the use of constraints discussed in this paper can be viewed as simply an example of finding support (or lack of it) for candidate answers. Many current systems (see, e.g. (Clarke et al., 2001), (Prager et al., 2004)) employ redundancy as a significant feature of operation: if the same answer appears multiple times in an internal *top-n* list, whether from multiple sources or multiple algorithms/agents, it is given a confidence boost, which will affect whether and how it gets returned to the end-user.

Finally, our approach is somewhat reminiscent of the scripts introduced by Schank (Schank et al., 1975, and see also Lehnert, 1978). In order to generate meaningful auxiliary questions and constraints, we need a model ("script") of the situation the question is about. Among others, we have identified one such script modeling the human life cycle that seems common to different question types regarding people.

3 Introducing QDC

QA-by-Dossier-with-Constraints is an extension of on-going work of ours called QA-by-Dossier (QbD) (Prager et al., 2004). In the latter, definitional questions of the form "Who/What is X" are answered by asking a set of specific factoid questions about properties of X. So if X is a person, for example, these auxiliary questions may be about important dates and events in the person's life-cycle, as well as his/her achievement. Likewise, question sets can be developed for other entities such as organizations, places and things.

QbD employs the notion of *follow-on questions*. Given an answer to a first-round question, the system can ask more specific questions based on that knowledge. For example, on discovering a person's profession, it can ask occupation-specific follow-on questions: if it finds that people are musicians, it can ask what they have composed, if it finds they are explorers, then what they have discovered, and so on.

QA-by-Dossier-with-Constraints extends this approach by capitalizing on the fact that a set of answers about a subject must be mutually consistent, with respect to constraints such as time and geography. The essence of the QDC approach is to initially return instead of the best answer to appropriately selected factoid questions, the top *n* answers (we use *n*=5), and to choose out of this top set the highest confidence answer combination that satisfies consistency constraints.

We illustrate this idea by way of the example, "*When did Leonardo da Vinci paint the Mona Lisa?*". Table 1 shows our system's top answers to this question, with associated scores in the range 0-1.

	Score	**Painting Date**
1	.64	2000
2	.43	1988
3	.34	1911
4	.31	1503
5	.30	1490

Table 1. Answers for "*When did Leonardo da Vinci paint the Mona Lisa?*"

The correct answer is "1503", which is in 4th place, with a low confidence score. Using QA-by-Dossier, we ask two related questions "*When was Leonardo da Vinci born?*" and "*When did Leonardo da Vinci die?*" The answers to these auxiliary questions are shown in Table 2.

Given common knowledge about a person's life expectancy and that a painting must be produced while its author is alive, we observe that the best dates proposed in Table 2 consistent with one another are that Leonardo da Vinci was born in 1452, died in 1519, and painted the Mona Lisa in 1503. [The painting date of 1490 also satisfies the constraints, but with a lower confidence.] We will examine the exact constraints used a little later. This example illustrates how the use of auxiliary questions helps constrain answers to the original question, and promotes correct answers with initial low

confidence scores. As a side-effect, a short dossier is produced.

	Score	**Born**		Score	**Died**
1	.66	1452		.99	1519
2	.12	1519		.98	1989
3	.04	1920		.96	1452
4	.04	1987		.60	1988
5	.04	1501		.60	1990

Table 2. Answers for auxiliary questions "*When was Leonardo da Vinci born?*" and "*When did Leonardo da Vinci die?*".

3.1 Reciprocal Questions

QDC also employs the notion of *reciprocal questions*. These are a type of follow-on question used solely to provide constraints, and do not add to the dossier. The idea is simply to double-check the answer to a question by inverting it, substituting the first-round answer and hoping to get the original subject back. For example, to double-check "Sacramento" as the answer to "What is the capital of California?" we would ask "Of what state is Sacramento the capital?". The reciprocal question would be asked of all of the candidate answers, and the confidences of the answers to the reciprocal questions would contribute to the selection of the optimum answer. We will discuss later how this reciprocation may be done automatically. In a separate study of reciprocal questions (Prager et al., 2004), we demonstrated an increase in precision from .43 to .95, with only a 30% drop in recall.

Although the reciprocal questions seem to be symmetrical and thus redundant, their power stems from the differences in the search for answers inherent in our system. The search is primarily based on the expected answer type (STATE vs. CAPITAL in the above example). This results in different document sets being passed to the answer selection module. Subsequently, the answer selection module works with a different set of syntactic and semantic relationships, and the process of asking a reciprocal question ends up looking more like the process of asking an independent one. The only difference between this and the "regular" QDC case is in the type of constraint applied to resolve the resulting answer set.

3.2 Applying QDC

In order to automatically apply QDC during question answering, several problems need to be addressed. First, criteria must be developed to determine when this process should be invoked. Second, we must identify the set of question types that would potentially benefit from such an approach, and, for each question type, develop a set of auxiliary questions and appropriate constraints among the answers. Third, for each question type, we must determine how the results of applying constraints should be utilized.

3.2.1 When to apply QDC

To address these questions we must distinguish between "planned" and "ad-hoc" uses of QDC. For answering definitional questions ("Who/what is X?") of the sort used in TREC2003, in which collections of facts can be gathered by QA-by-Dossier, we can assume that QDC is *always* appropriate. By defining broad enough classes of entities for which these questions might be asked (e.g. people, places, organizations and things, or major subclasses of these), we can for each of these classes manually establish once and for all a set of auxiliary questions for QbD and constraints for QDC. This is the approach we have taken in the experiments reported here. We are currently working on automatically learning effective auxiliary questions for some of these classes.

In a more ad-hoc situation, we might imagine that a simple variety of QDC will be invoked using solely reciprocal questions whenever the difference between the scores of the first and second answer is below a certain threshold.

3.2.2 How to apply QDC

We will posit three methods of generating auxiliary question sets:

- By hand
- Through a structured repository, such as a knowledge-base of real-world information
- Through statistical techniques tied to a machine-learning algorithm, and a text corpus.

We think that all three methods are appropriate, but we initially concentrate on the first for practical reasons. Most TREC-style factoid questions are about people, places, organizations, and things, and we can generate generic auxiliary question sets for each of these classes. Moreover, the purpose of this paper is to explain the QDC methodology and to investigate its value.

3.2.3 Constraint Networks

The constraints that apply to a given situation can be naturally represented in a network, and we find it useful for visualization purposes to depict the constraints graphically. In such a graph the entities and values are represented as nodes, and the constraints and questions as edges.

It is not clear how possible, or desirable, it is to automatically develop such constraint networks (other than the simple one for reciprocal questions), since so much real-world knowledge seems to be

required. To illustrate, let us look at the constraints required for the earlier example. A more complex constraint system is used in our experiments described later. For our Leonardo da Vinci example, the set of constraints applied can be expressed as follows[1]:

```
Date(Died) <= Date(Born) + 100
Date(Painting) >=  Date(Born) + 7
Date(Painting) <=  Date(Died)
```

The corresponding graphical representation is in Figure 1. Although the numerical constants in these constraints betray a certain arbitrariness, we found it a useful practice to find a middle ground between absolute minima or maxima that the values can achieve and their likely values. Furthermore, although these constraints are manually derived for our prototype system, they are fairly general for the human life-cycle and can be easily reused for other, similar questions, or for more complex dossiers, as described below.

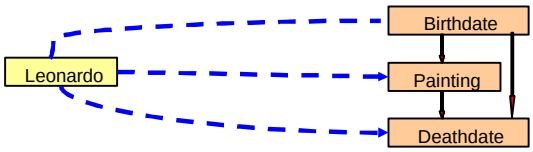

Figure 1. Constraint Network for Leonardo example. Dashed lines represent question-answer pairs, solid lines constraints between the answers.

We also note that even though a constraint network might have been inspired by and centered around a particular question, once the network is established, any question employed in it could be the end-user question that triggers it.

There exists the (general) problem of when more than one set of answers satisfies our constraints. Our approach is to combine the first-round scores of the individual answers to provide a score for the dossier as a whole. There are several ways to do this, and we found experimentally that it does not appear critical exactly how this is done. In the example in the evaluation we mention one particular combination algorithm.

3.2.4 Kinds of constraint network

There are an unlimited number of possible constraint networks that can be constructed. We have experimented with the following:

Timelines. People and even artifacts have life-cycles. The examples in this paper exploit these.

Geographic ("Where is X"). Neighboring entities are in the same part of the world.

Kinship ("Who is married to X"). Most kinship relationships have named reciprocals e.g. husband-wife, parent-child, and cousin-cousin. Even though these are not in practice one-one relationships, we can take advantage of sufficiency even if necessity is not entailed.

Definitional ("What is X?", "What does XYZ stand for?") For good definitions, a term and its definition are interchangeable.

Part-whole. Sizes of parts are no bigger than sizes of wholes. This fact can be used for populations, areas, etc.

3.2.5 QDC potential

We performed a manual examination of the 500 TREC2002 questions[2] to see for how many of these questions the QDC framework would apply. Being a manual process, these numbers provide an upper bound on how well we might expect a future automatic process to work.

We noted that for 92 questions (18%) a non-trivial constraint network of the above kinds would apply. For a total of 454 questions (91%), a simple reciprocal constraint could be generated. However, for 61 of those, the reciprocal question was sufficiently non-specific that the sought reciprocal answer was unlikely to be found in a reasonably-sized hit-list. For example, the reciprocal question to "How did Mickey Mantle die?" would be "Who died of cancer?" However, we can imagine using other facts in the dossier to craft the question, giving us "What famous baseball player (or Yankees player) died of cancer?", giving us a much better chance of success. For the simple reciprocation, though, subtracting these doubtful instances leaves 79% of the questions appearing to be good candidates for QDC.

4 Experimental Setup

4.1 Test set generation

To evaluate QDC, we had our system develop dossiers of people in the creative arts, unseen in previous TREC questions. However, we wanted to use the personalities in past TREC questions as independent indicators of appropriate subject matter. Therefore we collected all of the "creative" people in the TREC9 question set, and divided them up into classes by profession, so we had, for example, male singers Bob Marley, Ray Charles, Billy Joel and Alice Cooper; poets William Wordsworth and Langston Hughes; painters Picasso, Jackson Pollock

[1] Painting is only an example of an activity in these constraints. Any other achievement that is usually associated with adulthood can be used.

[2] This set did not contain definition questions, which, by our inspection, lend themselves readily to reciprocation.

and Vincent Van Gogh, etc. – twelve such groupings in all. For each set, we entered the individuals in the "Google Sets" interface (http://labs.google.com/sets), which finds "similar" entities to the ones entered. For example, from our set of male singers it found: Elton John, Sting, Garth Brooks, James Taylor, Phil Collins, Melissa Etheridge, Alanis Morissette, Annie Lennox, Jackson Browne, Bryan Adams, Frank Sinatra and Whitney Houston.

Altogether, we gathered 276 names of creative individuals this way, after removing duplicates, items that were not names of individuals, and names that did not occur in our test corpus (the AQUAINT corpus). We then used our system manually to help us develop "ground truth" for a randomly selected subset of 109 names. This ground truth served both as training material and as an evaluation key. We split the 109 names randomly into a set of 52 for training and 57 for testing. The training process used a hill-climbing method to find optimal values for three internal rejection thresholds. In developing the ground truth we might have missed some instances of assertions we were looking for, so the reported recall (and hence F-measure) figures should be considered to be upper bounds, but we believe the calculated figures are not far from the truth.

4.2 QDC Operation

The system first asked three questions for each subject X:

In what year was X born?
In what year did X die?
What compositions did X have?

The third of these triggers our named-entity type COMPOSITION that is used for all kinds of titled works – books, films, poems, music, plays and so on, and also quotations. Our named-entity recognizer has rules to detect works of art by phrases that are in apposition to "the film …" or the "the book …" etc., and also captures any short phrase in quotes beginning with a capital letter. The particular question phrasing we used does not commit us to any specific creative verb. This is of particular importance since it very frequently happens in text that titled works are associated with their creators by means of a possessive or parenthetical construction, rather than subject-verb-object.

The top five answers, with confidences, are returned for the *born* and *died* questions (subject to also passing a confidence threshold test). The *compositions* question is treated as a list question, meaning that all answers that pass a certain threshold are returned. For each such returned work W_i, two additional questions are asked:

What year did X have W_i?
Who had W_i?

The top 5 answers to each of these are returned, again as long as they pass a confidence threshold. We added a sixth answer "NIL" to each of the date sets, with a confidence equal to the rejection threshold. (NIL is the code used in TREC ever since TREC10 to indicate the assertion that there is no answer in the corpus.) We used a two stage constraint-satisfaction process:

Stage 1: For each work W_i for subject X, we added together its original confidence to the confidence of the answer X in the answer set of the reciprocal question (if it existed – otherwise we added zero). If the total did not exceed a learned threshold (.50) the work was rejected.

Stage 2. For each subject, with the remaining candidate works we generated all possible combinations of the date answers. We rejected any combination that did not satisfy the following constraints:

DIED >= BORN + 7
DIED <= BORN + 100
WORK >= BORN + 7
WORK <= BORN + 100
WORK <= DIED
DIED <= WORK + 100

The apparent redundancy here is because of the potential NIL answers for some of the date slots. We also rejected combinations of works whose years spanned more than 100 years (in case there were no BORN or DIED dates). In performing these constraint calculations, NIL satisfied every test by fiat. The constraint network we used is depicted in Figure 2.

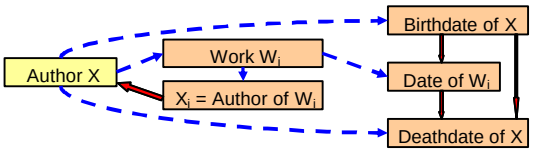

Figure 2. Constraint Network for evaluation example. Dashed lines represent question-answer pairs, solid lines constraints between the answers.

We used as a test corpus the AQUAINT corpus used in TREC-QA since 2002. Since this was not the same corpus from which the test questions were generated (the Web), we acknowledged that there might be some difference in the most common spelling of certain names, but we made no attempt to correct for this. Neither did we attempt to normalize, translate or aggregate names of the titled works that were returned, so that, for example, "Well-

Tempered Klavier" and "Well-Tempered Clavier" were treated as different. Since only individuals were used in the question set, we did not have instances of problems we saw in training, such as where an ensemble (such as The Beatles) created a certain piece, which in turn via the reciprocal question was found to have been written by a single person (Paul McCartney). The reverse situation was still possible, but we did not handle it. We foresee a future version of our system having knowledge of ensembles and their composition, thus removing this restriction. In general, a variety of ontological relationships could occur between the original individual and the discovered performer(s) of the work.

We generated answer keys by reading the passages that the system had retrieved and from which the answers were generated, to determine "truth". In cases of absent information in these passages, we did our own corpus searches. This of course made the issue of evaluation of recall only relative, since we were not able to guarantee we had found all existing instances.

We encountered some grey areas, e.g., if a painting appeared in an exhibition or if a celebrity endorsed a product, then should the exhibition's or product's name be considered an appropriate "work" of the artist? The general perspective adopted was that we were not establishing or validating the nature of the relationship between an individual and a creative work, but rather its existence. We answered "yes" if we subjectively felt the association to be both very strong and with the individual's participation – for example, Pamela Anderson and Playboy. However, books/plays about a person or dates of performances of one's work were considered incorrect. As we shall see, these decisions would not have a big impact on the outcome.

4.3 Effect of Constraints

The answers collected from these two rounds of questions can be regarded as assertions about the subject X. By applying constraints, two possible effects can occur to these assertions:

1. Some works can get thrown out.
2. An asserted date (which was the top candidate from its associated question) can get replaced by a candidate date originally in positions 2-6 (where sixth place is NIL)

Effect #1 is expected to increase precision at the risk of worsening recall; effect #2 can go either way. We note that NIL, which is only used for dates, can be the correct answer if the desired date assertion is absent from the corpus; NIL is considered a "value" in this evaluation.

By inspection, performances and other indirect works (discussed in the previous section) were usually associated with the correct artist, so our decision to remove them from consideration resulted in a decrease in both the numerator and denominator of the precision and recall calculations, resulting in a minimal effect.

The results of applying QDC to the 57 test individuals are summarized in Table 3. The baseline assertions for individual X were:

- Top-ranking birthdate/NIL
- Top-ranking deathdate/NIL
- Set of works W_i that passed threshold
- Top-ranking date for W_i /NIL

The sets of baseline assertions (by individual) are in effect the results of QA-by-Dossier WITHOUT Constraints (QbD).

	Assertions			Micro-Average			Macro-Average		
	Total	Correct	Truth	Prec	Rec	F	Prec	Rec	F
Baseline	1671	517	933	.309	.554	.396	.331	.520	.386
QDC	1417	813	933	.573	.871	.691	.603	.865	.690

Table 3. Results of Performance Evaluation. Two calculations of P/R/F are made, depending on whether the averaging is done over the whole set, or first by individual; the results are very similar.

The QDC assertions were the same as those for QbD, but reflecting the following effects:

- Some {W_i, date} pairs were thrown out (3 out of 14 on average)
- Some dates in positions 2-6 moved up (applicable to birth, death and work dates)

The results show improvement in both precision and recall, in turn determining a 75-80% relative increase in F-measure.

5 Discussion

This exposition of QA-by-Dossier-with-Constraints is very short and undoubtedly leaves may questions unanswered. We have not presented a precise method for computing the QDC scores. One way to formalize this process would be to treat it as evidence gathering and interpret the results in a Bayesian-like fashion. The original system confidences would represent prior probabilities reflecting the system's belief that the answers are correct. As more evidence is found, the confidences would be updated to reflect the changed likelihood that an answer is correct.

We do not know a priori how much "slop" should be allowed in enforcing the constraints, since auxiliary questions are as likely to be answered incor-

rectly as the original ones. A further problem is to determine the best metric for evaluating such approaches, which is a question for QA in general.

The task of generating auxiliary questions and constraint sets is a matter of active research. Even for simple questions like the ones considered here, the auxiliary questions and constraints we looked at were different and manually chosen. Hand-crafting a large number of such sets might not be feasible, but it is certainly possible to build a few for common situations, such as a person's life-cycle. More generally, QDC could be applied to situations in which a certain structure is induced by natural temporal (our Leonardo example) and/or spatial constraints, or by properties of the relation mentioned in the question (evaluation example). Temporal and spatial constraints appear general to all relevant question types, and include relations of precedence, inclusion, etc. For certain relationships, there are naturally-occurring reciprocals (if X is married to Y, then Y is married to X; if X is a child of Y then Y is a parent of X; compound-term to acronym and vice versa). Transitive relationships (e.g. greater-than, located-in, etc.) offer the immediate possibility of constraints, but this avenue has not yet been explored.

5.1 Automatic Generation of Reciprocal Questions

While not done in the work reported here, we are looking at generating reciprocal questions automatically. Consider the following transformations:

> "What is the capital of California?" -> "Of what state is <candidate> the capital?"
>
> "What is Frank Sinatra's nickname?" -> "Whose (or what person's) nickname is <candidate>?"
>
> "How deep is Crater Lake?" -> "What (or what lake) is <candidate> deep?"
>
> "Who won the Oscar for best actor in 1970?" -> "In what year did <candidate> win the Oscar for best actor?" (and/or "What award did <candidate> win in 1970?")

These are precisely the transformations necessary to generate the auxiliary reciprocal questions from the given original questions and candidate answers to them. Such a process requires identifying an entity in the question that belongs to a known class, and substituting the class name for the entity. This entity is made the subject of the question, the previous subject (or trace) being replaced by the candidate answer. We are looking at parse-tree rather than string transformations to achieve this. This work will be reported in a future paper.

5.2 Final Thoughts

Despite these open questions, initial trials with QA-by-Dossier-with-Constraints have been very encouraging, whether it is by correctly answering previously missed questions, or by improving confidences of correct answers. An interesting question is when it is appropriate to apply QDC. Clearly, if the base QA system is too poor, then the answers to the auxiliary questions will be useless; if the base system is highly accurate, the increase in accuracy will be negligible. Thus our approach seems most beneficial to middle-performance levels, which, by inspection of TREC results for the last 5 years, is where the leading systems currently lie.

We had initially thought that use of constraints would obviate the need for much of the complexity inherent in NLP. As mentioned earlier, with the case of "The Beatles" being the reciprocal answer to the auxiliary *composition* question to "Who is Paul McCartney?", we see that structured, ontological information would benefit QDC. Identifying alternate spellings and representations of the same name (e.g. Clavier/Klavier, but also taking care of variations in punctuation and completeness) is also necessary. When we asked "Who is Ian Anderson?", having in mind the singer-flautist for the Jethro Tull rock band, we found that he is not only that, but also the community investment manager of the English conglomerate Whitbread, the executive director of the U.S. Figure Skating Association, a writer for New Scientist, an Australian medical advisor to the WHO, and the general sales manager of Houseman, a supplier of water treatment systems. Thus the problem of word sense disambiguation has returned in a particularly nasty form. To be fully effective, QDC must be configured not just to find a consistent set of properties, but a number of independent sets that together cover the highest-confidence returned answers[3]. Altogether, we see that some of the very problems we aimed to skirt are still present and need to be addressed. However, we have shown that even disregarding these issues, QDC was able to provide substantial improvement in accuracy.

6 Summary

We have presented a method to improve the accuracy of a QA system by asking auxiliary questions for which natural constraints exist. Using these constraints, sets of mutually consistent answers can be generated. We have explored questions in the biographical areas, and identified other areas of applicability. We have found that our methodology exhibits a double advantage: not only can it im-

[3] Possibly the smallest number of sets that provide such coverage.

prove QA accuracy, but it can return a set of mutually-supporting assertions about the topic of the original question. We have identified many open questions and areas of future work, but despite these gaps, we have shown an example scenario where QA-by-Dossier-with-Constraints can improve the F-measure by over 75%.

7 Acknowledgements

We wish to thank Dave Ferrucci, Elena Filatova and Sasha Blair-Goldensohn for helpful discussions. 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

Chu-Carroll, J., J. Prager, C. Welty, K. Czuba and D. Ferrucci. "A Multi-Strategy and Multi-Source Approach to Question Answering", Proceedings of the 11th TREC, 2003.

Clarke, C., Cormack, G., Kisman, D.. and Lynam, T. "Question answering by passage selection (Multitext experiments for TREC-9)" in Proceedings of the 9th TREC, pp. 673-683, 2001.

Hendrix, G., E. Sacerdoti, D. Sagalowicz, J. Slocum: Developing a Natural Language Interface to Complex Data. VLDB 1977: 292

Lehnert, W. *The Process of Question Answering. A Computer Simulation of Cognition.* Lawrence Erlbaum Associates, Publishers, 1978.

Lenat, D. 1995. "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11.

Moldovan, D. and V. Rus, "Logic Form Transformation of WordNet and its Applicability to Question Answering", Proceedings of the ACL, 2001.

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

Prager, J., 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.), to appear in 2004.

Schank, R. and R. Abelson. "Scripts, Plans and Knowledge", *Proceedings of IJCAI'75.*

Voorhees, E. "Overview of the TREC 2002 Question Answering Track", *Proceedings of the 11th TREC*, 2003.

Warren, D., and F. Pereira "An efficient easily adaptable system for interpreting natural language queries," *Computational Linguistics*, 8:3-4, 110-122, 1982.

Winograd, T. Procedures as a representation for data in a computer program for under-standing natural language. *Cognitive Psychology*, 3(1), 1972.

Woods, W. Progress in natural language understanding --- an application in lunar geology. Proceedings of the 1973 National Computer Conference, AFIPS Conference Proceedings, Vol. 42, 441--450, 1973.