Syntactic and Semantic Decomposition Strategies for Question Answering from Multiple Resources*

Boris Katz, Gary Borchardt and Sue Felshin

Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139
{boris, borchardt, sfelshin}@csail.mit.edu

Abstract

This paper describes three language-motivated techniques for question answering from multiple resources. The techniques are situated within a QA architecture that combines language processing, reasoning, and information access.

1. Introduction

Building a QA system on the basis of multiple resources—particularly structured and semi-structured resources—introduces a distinct challenge: because resources differ in content, a QA system must possess the ability to decide where to look for particular pieces of information, and, in the case of complex questions, choose subquestions that can be readily answered on the basis of information contained in individual resources.

This paper describes a suite of three techniques we have been using to address this issue. These techniques are:

- a strategy for syntactically decomposing questions into subquestions on the basis of linguistic knowledge plus language-based descriptions of resource content,
- a strategy for semantically decomposing questions into subquestions on the basis of domain-motivated explanation patterns plus language-based descriptions of resource content, and
- a strategy for semantically decomposing both questions and resource content into lower-level assertions that provide a basis for comparison and inference.

These three strategies operate primarily within the middle layer of a three-layer question answering architecture as illustrated in Figure 1.

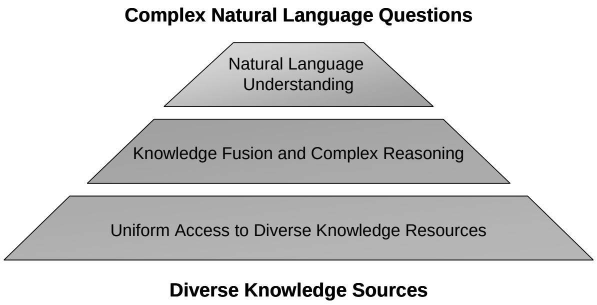

Complex Natural Language Questions
Natural Language Understanding
Knowledge Fusion and Complex Reasoning
Uniform Access to Diverse Knowledge Resources
Diverse Knowledge Sources

Figure 1: Question answering architecture

The top layer of this architecture provides NLP functionality for analyzing questions, engaging in system-user dialog, and generating responses. The middle layer provides functionality for question decomposition, reasoning, and knowledge fusion. The bottom layer provides functionality to enable uniform access to information contained in diverse sources.

These ideas are embodied in the joint operation of our START, IMPACT and Omnibase systems and have been applied to several large resources, including the CIA World Factbook, IMDB, Biography.com, the Monterey Weapons of Mass Destruction Terrorism Database, and the MIPT Terrorism Knowledge Base.

As this is work in progress, the paper concludes with a description of remaining steps in the elaboration of these ideas, plus a pair of evaluations planned for this work.

2. Parameterized Annotations

START [Katz, 1997; Katz, 1990] introduced the notion of *natural language annotations*, in which attached sentences and phrases are used to describe the contents of information segments for purposes of question answering. These annotations essentially use language itself as an indexing scheme. Given the enormous range of expression exhibited by simple phrases and sentences, natural language annotations have proven to be quite useful at describing a range of information content, including multimedia content.

In combination with sentence-level natural language processing techniques, annotations can enable questions to

*This work is supported by the Advanced Research and Development Activity as part of the AQUAINT Phase II research program.

be asked in many variant forms. As an example, the annotation

The headquarters of Procter & Gamble are located in Ohio.

can match a range of questions:

Where are the headquarters of P&G?
Where are Procter & Gamble's headquarters?
What is the location of P&G's headquarters?
Where are PG's headquarters located?
Are the headquarters of Procter & Gamble in Ohio?
What company's headquarters are in Ohio?
Procter & Gamble's headquarters are in what state?

Several techniques combine to make these matches possible. START begins by expressing both questions and annotations as sets of nested *ternary expressions* (constituent–relation–constituent triples) which highlight useful relationships for matching. Terms within these expressions are then matched using lexical techniques that associate synonyms, hyponyms (e.g., Procter & Gamble is a company, and Ohio is a state), and inflected forms of the same root, and by incorporating reference resolution strategies. Finally, structural matches are accomplished by invoking *S-rules*, which bridge differences in syntax that arise from the use of different verb alternations [Katz and Levin, 1988; Katz, 1990] and other paraphrases.

With large resources, of course, it is impractical to annotate all of the content. However, resources of all types—structured, semi-structured and unstructured—can contain significant amounts of parallel material. *Parameterized annotations* address this situation by combining fixed language elements with "parameters" that specify variable portions of the annotation. As such, they can be used to describe whole classes of content while preserving the indexing power of non-parameterized annotations. As an example, the parameterized annotation (with parameters in italics)

number people live in the metropolitan area of *city.*

can describe, on the data side, a large table of population figures for various cities. On the question side, this annotation can support questions submitted in many forms:

How many people reside in Chicago?
Do many people live in the metropolitan area of Pittsburgh?
What number of people live in Seattle's metropolitan area?
Are there many people in the Boston area?

In combination with other parameterized annotations that describe the population figures in other ways (for example, using the terms "population" or "populous", or referring to people "being in" the city, or the city being "large" or "small"), a significant range of questions can be processed from the given data as a result of matches to parameterized annotations.

As an extensive application of parameterized annotations, our Omnibase system [Katz *et al.*, 2002] supports the START system by answering object–property queries on the basis of information in a variety of resources. Parameterized annotations serve as the interface between START and Omnibase, allowing the combined systems to answer questions about a country's population, area, GDP or flag, for example, or a city's population, location or subway map, or a famous individual's place of birth, date of birth, or spouse.

Omnibase additionally supports START by acting as an external gazetteer for resource-specific terminology, with variants of terms calculated automatically from objects' names, extracted from semi-structured material in resources, or manually defined. This maintains the integrity of the abstraction layer: resource terminology is kept together with resource processing.

As an example, Omnibase will accept the variants

Thomas Alva Edison, Thomas Edison, Edison

and resolve any of them to the same identifier, composed of the resource name, the category of object, and the object's name, which is unique within the resource and category:

{Biography.com, *biography-person*, "Edison, Thomas Alva"}

Additionally, in many cases, a term will find matches in several classes. The term "France", for example, maps to identifiers in three classes:

{World Factbook, *factbook-country*, "France"}
{Infoplease.com, *infoplease-country*, "France"}
{Biography.com, *biography-person*, "France, Anatole"}

When a parameter in an annotation matches a term in the query, the system "binds" the parameter to the identifier(s) found by Omnibase. This helps the matcher identify cases in which particular questions match the annotations associated with particular resources. Also, where terms are ambiguous—e.g., "President Roosevelt"—the system can interactively query the user to resolve the ambiguity, and then use the identifier for the chosen entity.

When multiple resources are characterized using parameterized annotations, then, the matching process can serve to route input questions to the appropriate places. For example, given two parameterized annotations containing the parameters *factbook-country and infoplease-country*, the question "What are the biggest cities in Sudan?" will be routed to Infoplease.com, whereas the question "Is Arabic spoken in Sudan?" will be routed to the CIA World Factbook.

3. Syntactic Decomposition

With multiple resources, of course, it is often the case that two or more resources each contain a portion of the answer. In some cases, the question can be decomposed syntactically into parts that may be answered by different resources. The system must select which parts go to which resources, and in what order, in a way that is both accurate and efficient.

Syntax on its own informs us of accurate decomposition. For example, a question such as "Is the French president old?" refers to both the age and nationality of a president, but while we may find that the French president is Jacques Chirac, and then find how old Jacques Chirac is, it would be incorrect to find an old president (perhaps Bill Clinton, depending on context and interpretation) and then answer whether he is French. (A related and more realistic question would be "How old is the French president".) We can see that accurate decomposition relies on location of related nodes in the parse tree and not on particular types of relations (such as nominal adjectives vs. predicate adjectives) because a question such as "Is the president who is French old?" must be decomposed and answered in the same order as the first example.

While syntax guides us to accurate decompositions, in some cases there are several acceptable ways to decompose a question. In these situations, the system could consult its base of parameterized annotations to determine which of the acceptable subquestions do indeed have answers. Our approach implements a back-off strategy in which the absence of an answer for a subquestion leads to successive inclusion of additional ternary expressions, along the lines of the parse structure.

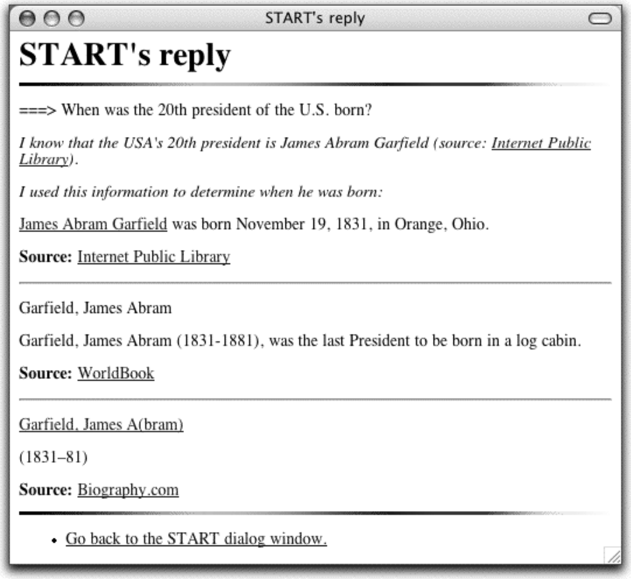
In more complex cases, there may be many acceptable decompositions of a question. This occurs when there are multiple relations expressed within the same syntactic projection or multiple, interdependent unknowns (for example, "What American companies have introduced a generic drug in a European country?"). In such cases, it can be computationally expensive to generate and evaluate all syntactically legal decompositions of the question. Instead, the matcher can be consulted in a separate step to identify candidate matches between parameterized annotations and subsets of the question's set of ternary expressions.

Whichever technique is employed, by evaluating the matching annotations for a chosen subset of the question's ternary expressions, we can enumerate a set of sub-responses, then simplify the remaining ternary expressions and repeat until the entire question has been resolved.

Handling complex questions on the basis of a single resource has been a focus of research since the earliest question answering systems, including the LUNAR system [Woods *et al.*, 1972], LADDER [Hendrix *et al.*, 1978] and PLANES [Waltz, 1978]. Systems that target the problem of question answering over multiple resources have typically taken the approach of first translating an input question into an intermediate logical representation, and in the realm of this intermediate representation matching parts of the question to content supplied by various resources. These systems include the Janus system [Bobrow *et al.*, 1990] and the more recent QUARK system [Waldinger *et al.*, 2004]. Our approach stays much closer to language in matching portions of questions directly to parameterized annotations describing resource content. We have found that this approach affords us a considerable amount of flexibility, as we can move to new domains and add new resources simply by composing new sets of parameterized annotations that work in combination with a general-purpose, domain-independent grammar.

Figure 2 illustrates the use of the syntactic decomposition strategy. Using the syntax of the question as a guide, the decomposition focuses first on identifying the 20th president of the United States (and not the president of the United States, for example, or the date of birth for a person). This portion of the input question is matched to a parameterized annotation of the form

USA's *ordinal* president



START's reply

===> When was the 20th president of the U.S. born?

I know that the USA's 20th president is James Abram Garfield (source: Internet Public Library).

I used this information to determine when he was born:

James Abram Garfield was born November 19, 1831, in Orange, Ohio.

Source: Internet Public Library

Garfield, James Abram

Garfield, James Abram (1831-1881), was the last President to be born in a log cabin.

Source: WorldBook

Garfield, James A(bram)

(1831–81)

Source: Biography.com

- Go back to the START dialog window.

Figure 2: A response produced by the syntactic decomposition strategy

Evaluating this annotation with its parameter bindings derived from the question yields a sub-response "James Abram Garfield" via the Internet Public Library. This value is substituted into the ternary expressions representing the remaining portion of the question. Then, the resulting set of ternary expressions is found to match the following annotation which contains a parameter related to Biography.com, as well as two similar annotations related to the Internet Public Library (IPL) and the WorldBook Encyclopedia:

biography-person is born in *date*

Matching of English terms to parameters is carried out using our Omnibase system's gazetteer capability, and the English term "James Abram Garfield" is found to be a member of each of the classes *"biography-person"*, *"ipl-president"* and *"worldbook-person"*, in some cases through synonymy with another form of the name.

Finally, the three matched annotations are evaluated to produce sub-responses representing answers to the question. START prepares a set of introductory sentences (italicized in Figure 2) to explain the returned answers. This is accomplished by invoking an English generation capability within START. To produce an appropriate English sentence, START's English generator takes as input the ternary expressions from the question fragment, ancillary information from the original parse of the question, and linguistic context specifications provided by START's discourse manager.

Natural language generation plays an important role in communicating the results of the syntactic decomposition strategy. START's generator includes facilities for combining fragments of information and realizing proper nouns and common nouns in different forms. In performing generation, the system obeys relevant linguistic constraints and discourse principles, for example, to use short names or pronouns as subsequent references to an entity to eliminate redundancy in context, or to include an object type in order to avoid confusion with like-named objects of other types, e.g., "the country of Georgia" vs the state of Georgia".

4. Semantic Decomposition

In other cases, the question can be decomposed semantically. We create simple rules that knit particular "key" domain questions—as specified by system designers and users—to subquestions that are answerable by particular resources.

This strategy makes use of "knowledge templates" as a quick way to bring structured data into the realm of language. The Monterey Weapons of Mass Destruction Terrorism Database (http://cns.miis.edu/) illustrates the use of this technique. Individual records in the Monterey WMD Terrorism Database describe terrorist incidents relating to weapons of mass destruction. The database records contain 39 fields, of which we extract seven for insertion within a fixed, language-based template

In [*year*], [*group type*] [*group name*] carried out a [*event type*] in [*country*], involving [*agent type*] [*agent name*].

The following example instantiates this template for a particular record in the database:

In [1995], [religious cult] [Aum Supreme Truth] carried out a [use of agent] in [Japan], involving [chemical agent] [sarin].

With a resource's data depicted in such a manner, as a first step toward question answering, we can associate these knowledge templates directly with parameterized annotations. This enables the QA system to support a range of questions concerning the immediate content of the resource.

For the Monterey WMD Terrorism Database, parameterized annotations associated with the knowledge template employed in the above example enable our system to answer questions such as the following:

What agent type did Aum Shinrikyo use to execute an attack in Japan?

In what country did the PKK try to acquire a chemical agent?

Have any groups been involved in an attack in the US?

What groups have carried out an attack in the US?

Did the Japanese Red Army issue a threat with a chemical agent in Japan?

Has the KKK been able to carry out an attack in the US?

Did Aum Supreme Truth plot to use a chemical agent in the United States?

Building on this foundation, we may then insert semantic decomposition rules between parameterized annotations and the knowledge templates that directly access information from resources. These semantic decomposition rules can combine information from a single resource or from multiple resources, or they may build upon the results of other semantic decomposition rules. The semantic decomposition rules operate by collecting lists of responses for each of their sub-queries, then performing constraint propagation to enumerate a list of responses for the rule head. In this manner, they process all solutions in parallel and generate a relatively small number of suitably generalized sub-queries to other rules or to external resources. In addition, for resources that require specified values—constants or finite sets of alternative values—in particular query argument positions, a delay mechanism will postpone evaluation of a sub-query until constraint propagation resulting from other, sister sub-queries has had a chance to constrain otherwise unbounded variables to finite value sets.

In all cases, the semantic decomposition rules are motivated by patterns of explanation; that is, their purpose is to present plausible conclusions to a human user, along with human-understandable evidence that supports the conclusions. It is up to the user to pass judgment on the conclusions, and as such, these rules are intentionally weaker than strict inference rules. By focusing on key domain questions and their associated patterns of explanation, this approach can operate independently of—or in concert with—an underlying domain theory as used in systems such as QUARK [Waldinger *et al.*, 2004] in the context of multiple resource question answering, or WEBCOOP [Benamara, 2004] in the context of cooperative question answering.

The following is an example of a semantic decomposition rule used in our system (variables are specified as "some group", "some country", etc.):

[some group] could carry out an attack in [some country] using a [some agent type]. ↔

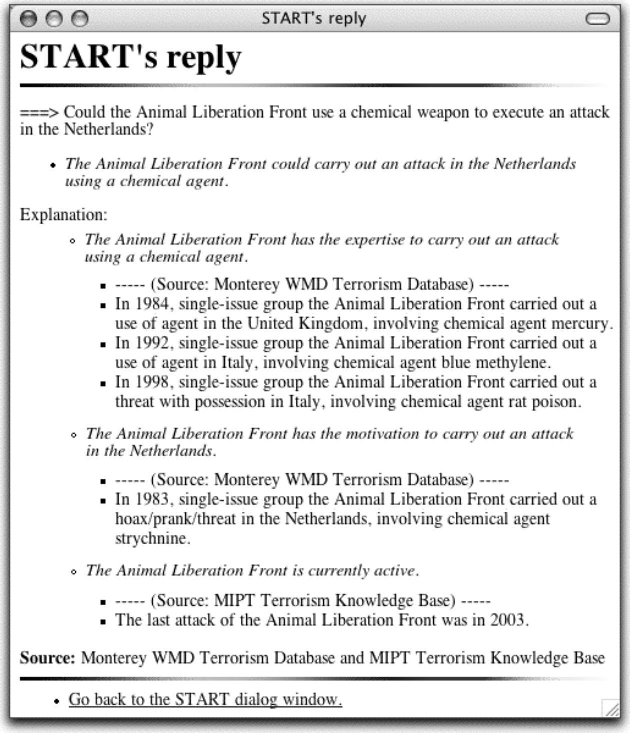
[some group] has the expertise to carry out an attack using a [some agent type]. AND

[some group] has the motivation to carry out an attack in [some country]. AND

[some group] is currently active.

This rule specifies a pattern by which the capability of a group to carry out an attack is explained in terms of the group's expertise with the specified weapon type, motivation to carry out an attack in the specified country, and status as an active group. Other, supporting rules provide patterns for explaining these relationships in terms of information directly accessible from external resources, including the Monterey WMD Terrorism Database and the MIPT Terrorism Knowledge Base (http://www.tkb.org/). This cluster of rules and sub-rules provides support for a key question type in the domain of terrorist group characteristics and capabilities. Other key questions are supported by rules that identify entities described in various resources and rules that extract relationships such as terrorist groups being associated with particular countries or particular types of agents.

Figure 3 illustrates the use of this technique on a question that invokes the decomposition rule illustrated above.



START's reply

===> Could the Animal Liberation Front use a chemical weapon to execute an attack in the Netherlands?

- *The Animal Liberation Front could carry out an attack in the Netherlands using a chemical agent.*

Explanation:

- *The Animal Liberation Front has the expertise to carry out an attack using a chemical agent.*
  - ----- (Source: Monterey WMD Terrorism Database) -----
  - In 1984, single-issue group the Animal Liberation Front carried out a use of agent in the United Kingdom, involving chemical agent mercury.
  - In 1992, single-issue group the Animal Liberation Front carried out a use of agent in Italy, involving chemical agent blue methylene.
  - In 1998, single-issue group the Animal Liberation Front carried out a threat with possession in Italy, involving chemical agent rat poison.
- *The Animal Liberation Front has the motivation to carry out an attack in the Netherlands.*
  - ----- (Source: Monterey WMD Terrorism Database) -----
  - In 1983, single-issue group the Animal Liberation Front carried out a hoax/prank/threat in the Netherlands, involving chemical agent strychnine.
- *The Animal Liberation Front is currently active.*
  - ----- (Source: MIPT Terrorism Knowledge Base) -----
  - The last attack of the Animal Liberation Front was in 2003.

Source: Monterey WMD Terrorism Database and MIPT Terrorism Knowledge Base

- Go back to the START dialog window.

Figure 3: A response produced by the semantic decomposition strategy

By associating parameterized annotations with the heads of semantic decomposition rules, a range of question variants can be handled. The rule listed above and illustrated in Figure 3, for example, can support the following questions:

Could the KKK be involved in an attack using biological weapons?

Could an attack be carried out in Italy involving chemical weapons?

Are any groups trying to conduct an attack in the United States?

What groups will be able to carry out an attack in the US?

In what countries could Hizballah execute an attack?

Aum Shinrikyo could carry out an attack with what agent types?

With the Omnibase system acting as an external gazetteer, differences in terminology on the calling end are bridged at the time sub-questions are answered. Resource-specific terminology that is returned in a sub-response can be mapped to standardized terminology using a post-evaluation mapping.

5. Decomposition to Lower-Level Assertions

The syntactic and semantic decomposition strategies described above work to answer key questions in a domain on the basis of information in specific sets of resources. However, many other user questions remain unanswered. To address this issue, START employs a number of techniques to help its user take advantage of the answerable questions. These techniques redirect unanswered questions to nearby, answerable key questions or provide meta-level responses that help characterize the range of questions the system can answer. In addition, we are incorporating techniques to extend the range of questions answered by the system.

One particular problem that arises when more and more resources are added is that it becomes increasingly difficult to identify all of the potential semantic interactions between content in different resources. This limits the number of questions the system can answer.

We are focusing on event instances in particular, with the intent of decomposing these instances into collections of lower-level assertions that describe what happens during the events. In this way, we hope to automatically identify a number of inter-event relations, such as when the occurrence of several events implies or contradicts the occurrence of another event. This work is grounded in our work on the *transition space* representation [Borchardt, 1992; Borchardt, 1994].

The transition space representation is similar in spirit to the representation used by Narayanan (as described, for example, in [Narayanan and Harabagiu, 2004]) in that it attempts to model the temporal unfolding of events in a cognitively-motivated way in order to support reasoning and detection of inter-event relationships. However, the transition space representation is more of a descriptive account of activity, whereas the core of Narayanan's approach involves active structures that serve to simulate the unfolding of an event.

In our approach, the temporal unfolding of various events is described by sets of language-based statements that specify, in particular, *changes* in the values of key attributes of event participants. These statements concerning changes are then further decomposed into statements regarding momentary presence and absence of attributes, and, ultimately, a lowest level of statements that specify whether one quantity, such as a timestamped attribute value, is equal to, not equal to, greater than, or not greater than another quantity. The following are examples of language-based statements that serve as a grounding for this representation:

The affinity between PIJ *and* Hezbollah **increases**.
The supreme leader of al-Saiqa **does not change**.
The PLF **becomes** *a part of* the PFLP-GC.
The contact between PFLP-GC *and* Hamas **appears**.
Jordan **ceases to be** *a base of operations for* al-Fatah.
The number of vehicles *at* Station #43 **decreases**.
Khalid al-Hasan **is** *a leader of* al-Fatah.

The rivalry between al-Fatah *and* PIJ **exists**.

Fathi Shaqaqi **is not** *alive*.

The affinity between PIJ *and* Hezbollah in 1986 **exceeds** *the affinity between* PIJ *and* Hezbollah in 1985.

The supreme leader of PLO in 1970 **equals** Yasser Arafat.

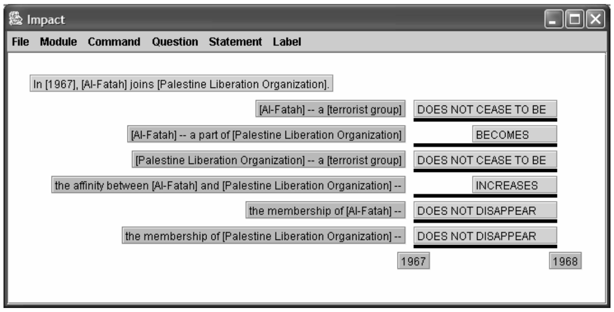
Symbolically, such statements can be depicted as compositions of four representational entities: predicates, attributes, objects and times, as in the following example:

Between [2003] and [2004], the trade deficit between [United States] and [China] -INCREASES-.

predicate:	INCREASES
attribute:	the trade deficit between [] and [] --
objects:	United States, China
times:	2003, 2004

Drawing motivation from the cognitive psychology literature (e.g., [Miller and Johnson-Laird, 1976]), attributes are taken to be either qualitative or quantitative in nature, taking one or two arguments, and changes are specified as comparisons across pairs of time points. Assuming the existence of a "null" value in the range of each attribute, a set of ten predicates suffices to describe changes: APPEARS (alternate form BECOMES), DOES NOT APPEAR (DOES NOT BECOME), DISAPPEARS (CEASES TO BE), DOES NOT DISAPPEAR (DOES NOT CEASE TO BE), CHANGES, DOES NOT CHANGE, INCREASES, DOES NOT INCREASE, DECREASES and DOES NOT DECREASE.

In combination, these representational constructs are used to model the unfolding of particular events. Figure 4 illustrates a simple transition space representation for the event of one terrorist group joining another terrorist group, showing only the top level of the representation, concerning changes.



In [1967], [Al-Fatah] joins [Palestine Liberation Organization].

[Al-Fatah] -- a [terrorist group]	DOES NOT CEASE TO BE
[Al-Fatah] -- a part of [Palestine Liberation Organization]	BECOMES
[Palestine Liberation Organization] -- a [terrorist group]	DOES NOT CEASE TO BE
the affinity between [Al-Fatah] and [Palestine Liberation Organization] --	INCREASES
the membership of [Al-Fatah] --	DOES NOT DISAPPEAR
the membership of [Palestine Liberation Organization] --	DOES NOT DISAPPEAR

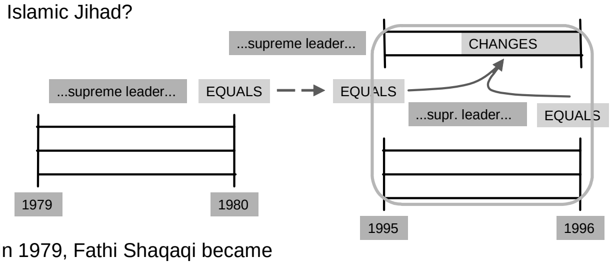
Figure 4: Representing an event in terms of lower-level assertions

This representation is standardized to a temporal granularity of years and includes information on changes and non-changes in key domain attributes: one group becomes a part of the other group, the affinity between the two groups increases, and so forth.

Given event representations of this sort, we are inserting functionality within our multi-component QA system to test for particular event occurrences given sets of other event occurrences. The underlying implementation decomposes all assertions to the lowest level of assertions, where quantities are specified as being (not) equal to or (not) greater than other quantities; then it must perform inference at this level, calculate a suitable persistence of attribute values, and finally recompose the resulting assertions into new change-level assertions and events.

Figure 5 illustrates this process in the answering of a question about one individual replacing another as leader of a terrorist group. Given declarations of each of the two individuals having ascended to the leadership of the group in various years, plus persistence of the leader of the group being declared as being equal to a particular individual, the various aspects of one leader replacing another are satisfied, such that the system can present an explanation of this possibility to the user.

In 1995, who replaced whom as the supreme leader of Palestinian Islamic Jihad?



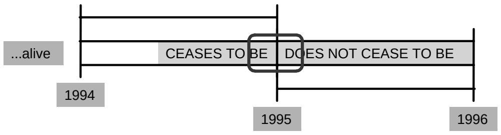
In 1979, Fathi Shaqaqi became the supreme leader of Palestinian Islamic Jihad. *(Source: MIPT Terrorism KB)*

In 1995, Ramadan Shallah became the supreme leader of Palestinian Islamic Jihad. *(Source: MIPT Terrorism KB)*

Figure 5: Restatement of an event in terms of two other events

In a similar manner, such representations can be used to detect conflicts between events. In Figure 6, a question about an individual possibly dying in one year can be answered in the negative, due to the presence of a following event in which that individual becomes the supreme leader of a group.

Did Ramadan Shallah die in 1994?



In 1995, Ramadan Shallah became the supreme leader of Palestinian Islamic Jihad. *(Source: MIPT Terrorism KB)*

Figure 6: A conflict between two events

Other question types supported by this technique include questions concerning values of particular attributes (e.g., "What can you tell me about the affinity between the PFLP-GC and the PLO?") and questions concerning the persistence of attribute values (e.g., "What countries were bases of operations for Al-Fatah in 1980?"). All of the question types supported by this technique will be available to users through the use of parameterized annotations. However, we also expect to see a substantial benefit from using this strategy in a supporting role. By factoring such inter-event relationships into the syntactic and semantic decomposition strategies described above, we hope to increase their power considerably, allowing them to bridge differences in the vocabulary of event types and situations they reference.

6. Evaluation

The strategies described in this paper operate within an end-to-end QA architecture and system and as such can be evaluated in realistic QA contexts. We are at the moment designing two evaluations which will be used to assess the effectiveness of the implemented techniques. The first evaluation will involve unpracticed users composing variant questions that exercise the syntactic and semantic range of inquiries supported by the system. The second evaluation will have unpracticed users engaging in task-based QA sessions on the basis of targeted scenarios, such as updating a report on a country, or assessing the capabilities of a group or organization.

In the meantime, syntactic decomposition has been incorporated into our public Web server (available at http://start.csail.mit.edu/) and as such has been tested by thousands of people around the world. Semantic decomposition has been incorporated into a restricted server which covers a specific domain area and has been made available to a set of users associated with that domain.

7. Conclusion

Several types of knowledge are utilized by the techniques presented here. In addition to knowledge provided by multiple, external resources, these techniques make use of: (1) resource content specifications in the form of parameterized annotations, (2) components of knowledge used to match questions to annotations, including lexical knowledge, structure-transforming S-rules, and knowledge of synonyms and hyponyms, (3) linguistic constraints that govern decomposition of complex questions into subquestions, (4) explanation patterns in the form of semantic decomposition rules, and (5) event models, used to elaborate lower-level assertions for event instances. A unifying thread to these components of knowledge is that they all use language-based representations or serve to describe aspects of language.

We believe that future question answering systems will be faced with an ever-increasing need to handle multiple resources. Critical to handling multiple resources is the ability to describe their content, and we have found that language is a powerful tool for accomplishing this task. Given suitable descriptions of resource content, it then falls upon the question answering system to allocate QA subtasks to the available information resources in an effective manner. Whether this is accomplished by following linguistic cues provided in the questions themselves, by applying domain-inspired patterns of decomposition and explanation, or by elaborating detailed, underlying representations of meaning, a question answering system that is truly capable of responding on the basis of multiple resources will bestow an important benefit on its users by providing them with “one-stop shopping” for information.

References

Benamara, F., 2004. “Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment,” in *Proceedings of the ACL 2004 Workshop on Question Answering in Restricted Domains*, 31-38.

Bobrow, R. J., Resnik, P. and Weischedel, R. M., 1990. “Multiple Underlying Systems: Translating User Requests into Programs to Produce Answers,” in *Proceedings of the 28th Annual Meeting of the ACL*, Pittsburgh, PA, 227-234.

Borchardt, G. C., 1992. "Understanding Causal Descriptions of Physical Systems," *Proc. AAAI Tenth National Conference on Artificial Intelligence*, 2-8.

Borchardt, G. C., 1994. *Thinking between the Lines: Computers and the Comprehension of Causal Descriptions*, MIT Press.

Hendrix, G., Sacerdoti, E., Sagalowicz, D. and Slocum, J., 1978. “Developing a Natural Language Interface to Complex Data,” *ACM Transactions on Database Systems*, 3(2): 105-147.

Katz, B. and Levin, B., 1988. “Exploiting Lexical Regularities in Designing Natural Language Systems.” *Proceedings of the 12th International Conference on Computational Linguistics (COLING '88)*

Katz, B., 1990. “Using English for Indexing and Retrieving,” in P. H. Winston and S. A. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, volume 1, MIT Press, Cambridge, MA.

Katz, B., 1997. “Annotating the World Wide Web using Natural Language,” *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97).*

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A. J. and Temelkuran, B., 2002. “Omnibase: Uniform Access to Heterogeneous Data as a Component of a Natural Language Question Answering System,” *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 02).*

Miller, G. A. and Johnson-Laird, P. N., 1976. *Language and Perception*, Harvard University Press.

Narayanan, S. and Harabagiu, S., 2004. “Question Answering Based on Semantic Structures,” *Proceedings of the International Conference on Computational Linguistics (COLING 2004).*

Waldinger, R., Appelt, D. E., Fry, J., Israel, D. J., Jarvis, P., Martin, D., Riehemann, S., Stickel, M. E. , Tyson, M., Hobbs, J., and Dungan, J. L., 2004. “Deductive Question Answering from Multiple Resources,” in *New Directions in Question Answering*, M. Maybury (ed.), AAAI/MIT Press.

Waltz, D. L., 1978. “An English Language Question Answering System for a Large Relational Database,” *Communications of the ACM*, 21(7):526-539.

Woods, W. A., Kaplan, R. M. and Webber, B. N., 1972. *The Lunar Sciences Natural Language Information System: Final Report*, BBN Report 2378, Bolt Beranek and Newman, Inc., Cambridge, MA.