Towards deep reasoning with respect to natural language text in scientific domains

Chitta Baral, Shanshan Liang and Vo Nguyen
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ

Abstract

In this paper we take some initial steps towards deep reasoning with respect to natural language text in scientific domains. In particular we consider answering Why and How questions with respect to a high school Biology text. In comparison to other kinds of questions, Why and How questions (the later referred to as procedural questions by some) have been less explored in the Question Answering literature. However, they have been considered to a greater degree in the Reasoning about Actions literature and in the Knowledge Representation and Reasoning literature. In this paper we borrow some ideas from those literature and use answer set programming (ASP) as the knowledge representation language. A key concern in our representation of natural language statements and questions is that one should be able to obtain the ASP representation of natural language text in an automated way. We also briefly discuss how some of the background knowledge needed for deep reasoning can be obtained automatically and how various levels of approximate reasoning can be done.

Introduction

The long term goal of our proposed research is to develop systems that can read and "understand" text. By understanding text we mean that the system can take text and questions in natural language and answer the questions with respect to the given text. This is closely related to the goals of Project Halo[1](Gunning et al. 2010), initiated in 2003, which aimed at encoding knowledge in particular chapters of a high school chemistry book so that it could answer AP level questions. Three groups involved in this project were able to do this by using humans to encode the knowledge; but they later analyzed that it cost $10,000 per page of encoding. Hence, an approach based on human coding of knowledge in text is not scalable. In fact, despite significant progress in the research area of knowledge representation and reasoning, we believe the reason currently there are comparatively few knowledge based intelligent systems is because human encoding of knowledge in appropriate knowledge representation (KR) languages requires the humans to be trained in that KR language and often the skills needed to be well trained in KR is different from the skills developed by domain experts, thus resulting in a knowledge acquisition bottleneck. We believe that the way to address this knowledge acquisition bottleneck is to develop systems that can read text and acquire knowledge in those text by translating the text to statements in an appropriate KR language.

In regards to translating English to knowledge representation languages there are several relevant directions and works. Researchers in natural language semantics (for example, (Cooper 1989)) have considered various target logics, but many of them are theoretical proposals developed to express specific natural language features. Several researchers (Hobbs 1985; Blackburn and Bos 2005; Bos et al. 2004; Moldovan et al. 2002) have proposed methods to translate natural language to first order logic. Another approach involves using the output of parsers such as the Stanford parser (de Marneffe, MacCartney, and Manning 2006) or Link Grammar parser (Sleator and Temperley 1993) and obtaining a semantic representation from them. In (Balduccini, Baral, and Lierler 2008) our group processed logic forms produced by the LCC system as well as output of Link Grammar parser to obtain semantic information that is then used with domain knowledge. Recently, our group has developed a learning based approach (Baral et al. 2011) similar to ones used by (Zettlemoyer and Collins 2005; Wong and Mooney 2007) to translate English sentences to targeted KR languages. In this approach the system is given example of sentences and their translations, and the system learns λ-calculus formulas of words and associated weights. These are then used to translate new sentences. Our approach (Baral et al. 2011) differs from (Zettlemoyer and Collins 2005; Wong and Mooney 2007) in that we use an inverse-λ algorithm that can find the λ-calculus formula f given λ-calculus formulas g and h such that (a) $f \circ g = h$ and (b) $g \circ f = h$. This allows the system to learn the meaning of words (represented as λ-calculus formulas) from the meaning of a sentence and some of the other words in that sentence whose meanings have already been determined.

The motivation behind mentioning the above methods to automatically translate English to KR formalisms is that in *our current goal of doing deep reasoning with respect to natural language text in a high school biology text book*, we would like to always keep in mind that representation of

[1]http://projecthalo.com/

each sentence, both in the questions and in the text, needs to be in a form that *can be obtained by an automated method*. This is different from traditional KR approaches where humans read the text and are told about the type of questions that may be asked and they come up with a representation of what they have read so that reasoning can be done to answer the given type of questions.

For our current goal of doing deep reasoning with respect to natural language text in a high school biology text book (Campbell and Reece 2011; Spaulding et al. 2011) we focus on "Why" and "How" questions. The "Why" questions are often used to find the *reason, cause or explanation* behind a particular statement or a conclusion. For example, in the question, "Why is surface area important to normal cell function?" the expected answer is the collection of various reasons that make surface area important to cell function. Sometimes, the statement part of the "Why" question is something that happened. To answer that one may need to do backward reasoning about what are the preconditions of an event to happen and what can trigger an event to happen. In general, answering "Why" question often involves connecting the dots in an appropriate order and the answer often starts with the word "Because".

In the literature (Aouladomar 2005) "How" questions have been referred to as procedural questions. However, some "How" questions could be factoid questions. Examples of such questions include: (a) Evaluative How, as in, "How fast is the swing?" (b) Nominal How, as in, "How do you say ..." and (c) questions such as "How many ...". We are more interested in "How" questions that are: (d) Causal, as in, "How did X happen" and (e) Procedural, as in, "How is X done" and "How does one do ...". To answer the causal and procedural "How" questions recent research in reasoning about actions and narratives is useful.

In this paper we illustrate our approach of using answer set programming to answer "Why" and "How" questions through several examples. Our subsequent goal would be to develop a general methodology for doing that. But we first give a brief background on answer set programming, our knowledge representation language of choice.

Background: Answer Set Programming

We use a broader meaning of the term *answer set programming* (ASP) than originally used in (Marek and Truszczyński 1999; Niemelä 1999). By ASP we mean logic programming under the answer set semantics. We consider ASP as one of the most developed knowledge representation language for the following reasons. ASP is non-monotonic and is expressive enough to represent several classes of problems in the complexity hierarchy. Furthermore, it has solid theoretical foundations with a large body of building block results (e.g., equivalence between programs, systematic program development, relationships to other non-monotonic formalisms) (Baral 2003); it also has a large number of efficient computational tools. Default statements and various forms of exceptions can be naturally represented in ASP (Gelfond and Leone 2002). We now review the basic notions in ASP.

We assume a first order language $\mathcal{L}$. Literals are constructed from atoms in $\mathcal{L}$. A *positive* (or *negative*) literal is of the form A (resp. $\neg A$) where A is an atom in $\mathcal{L}$.

An *ASP program* (or *program*) is a set of rules (ASP rules) of the following form:

$$a \leftarrow a_1, \ldots, a_m,\ not\ b_1, \ldots,\ not\ b_n \qquad (1)$$

where $m, n \geq 0$ and each a, a_j, and b_t is a literal in $\mathcal{L}$; and not represents negation as failure (or default negation). For a rule r of the form (1), $head(r)$ denotes a; $pos(r)$ (*positive* body) denotes the set $\{a_1, \ldots, a_m\}$; and $neg(r)$ (*negative* body) denotes $\{b_1, \ldots, b_n\}$. Intuitively, a rule r of the form (1) states that if all the literals in $pos(r)$ are believed to be true and none of the literals in $neg(r)$ is believed to be true then the literal $head(r)$ must be true.

The notion of answer set semantics for ASP programs is defined in (Gelfond and Lifschitz 1988). Let P be a ground program[2].

Let S be a set of ground literals in the language of P. S satisfies the body of a rule r if $pos(r) \subseteq S$ and $neg(r) \cap S = \emptyset$. S satisfies a rule r if either $head(r) \in S$ or S does not satisfy the body of r. S satisfies a program P if it satisfies every rule in P. S is an *answer set* of P if it is a minimal set of literals satisfying all the rules in P^S where P^S is obtained from P by

(i) Deleting all rules from P that contain some $not\ l$ in their body and $l \in S$.

(ii) All occurrences of $not\ l$ from the remaining rules.

A program P is *consistent* if it has at least one answer set. Given a program P and a literal l, we say that P *entails* l, denoted by $P \models l$, if l belongs to every answer set of P.

Illustration of Answering Why Questions

We illustrate our approach via two "Why" questions: Why is surface area important to normal cell function? Why are prokaryotes so much smaller than eukaryotes? We start with representing the questions.

```
% Q1: Why is surface area important to
% normal cell function?

question(q1).
qtype(q1,why).
qrelation(q1,important).

qrelationattr(q1,subject,surface_area).
qrelationattr(q1,object,normal_cell_fn).

aspects(surface_area,surface).
aspects(surface_area,area).
aspects(normal_cell_fn,normal).
aspects(normal_cell_fn,cell).
aspects(normal_cell_fn,function).

%  Q8: Why are prokaryotes so much
% smaller than eukaryotes?
```

[2]Rules with variables are replaced by the set of its ground instantiations.

```
question(q8).
qtype(q8,why).
qrelation(q8,so_much_smaller).

qrelationattr(q8,subject,prokaryotes).
qrelationattr(q8,object,eukaryotes).

aspects(so_much_smaller, smaller).
```

The above illustrates the representation of Why questions of the form, "Why is X R Y", where R is a relation and X is the subject of R and Y is the object of R. X and Y could be phrases and there meaning could be further elaborated. Here we use a simple elaboration listing the various aspects of X and Y explicitly.

We now show the encoding of the statement that can be used to answer the question q1.

```
% A1: A high surface-to-volume ratio
% facilitates the exchange of materials
% between a cell and its environment.

statement(e1).
statementtype(e1,event).
reltype(e1,facilitate).
relattr(e1,subject,high_surface_vol_ratio).
relattr(e1,object,e2).

aspects(high_surface_vol_ratio,high).
aspects(high_surface_vol_ratio,surface).
aspects(high_surface_vol_ratio,volume).
aspects(high_surface_vol_ratio,ratio).

statement(e2).
statementtype(e2,event).
reltype(e2,exchange).
relattr(e2,subject,cell).
relattr(e2,subject,environment).
relattr(e2,object,material).
```

In the above encoding e1 refers to the complete statement. The first five lines encode that e1 is an event of the type 'facilitate' and the subject and object are as given above. The object is itself an event e2 of the type 'exchange' and the subjects are 'cell' and 'environment' and the object is 'material'. The various parts of the phrase "high surface to volume ration" are stated as aspects. For now we aim to do a superficial reasoning with such phrases. For more accurate reasoning with respect to this phrase a more detailed semantic representation would be needed.

Now that we have given the representation of the question and the representation of the statement from where the answer is supposed to come lets analyze what rules and background knowledge we may need to answer the question. We need to address three connections.

1. We need to connect that one way to show X to be important to Y is to show X facilitates Y. (There are, of course, other ways of showing such importance.)
2. We need to connect the phrase 'surface area' in Q1 with the phrase 'high surface to volume ratio'.
3. We need to connect the phrase 'normal cell function' in Q1 with e2 of A1.

The first one can be addressed by adding domain knowledge about evidences that support X is important to Y. One way to do this is to search in the web for the phrase "is important to ... because" and find sentences that have them and from them extract the evidences that imply X is important to Y. Such a method with respect to the search engine Google yields the fact that one can show X is important to Y by showing X facilitates Y. We encode this fact as follows:

```
explainrel(important,facilitate).
```

The second one can be addressed in a superficial way by noting that the word 'surface' appears in both phrases. While this is a superficial way to address it, many people not mathematically knowledgeable enough to precisely connect 'surface area' with 'surface to volume ration' will have the only option of making a superficial connection. While such superficial connections may lead to a wrong reasoning, that is natural in the absence of deeper background knowledge. We achieve this superficial connection by the relax predicate often used in co-operative question answering. The first rule below says that X and X' can relaxed if they have a common aspect. The second rule below is used to connect the phrase 'normal cell function' with e2.

```
relax(X,X'):-aspects(X,Y),aspects(X',Y).
relax(X,X'):-aspects(X,Y),aspects(X',Z),
                          isa(Z,Y).
relax(X,X'):-aspects(X,X').
relax(X,X'):-aspects(X',X).
```

But we need another fact to complete the connection. The needed fact given below can be obtained from a typical semantic hierarchy of words which tells us 'exchange' is a type of 'function'.

```
isa(exchange,function).
```

Now we are ready to write the rule that will be used to answer several type of 'Why' questions.

```
answer(Q,Z):-question(Q), qtype(Q,why),
qrelation(Q,R),qrelationattr(Q,subject,X),
qrelationattr(Q,object,Y),explainrel(R,R'),
relax(X,X'),relax(Y,Y'),statement(Z),
statementtype(Z,event),reltype(Z,R'),
relattr(Z,subject,X'),relattr(Z,object,Y').
```

The above rule says that Z is an answer of Q if Q is a 'Why' question of relation R with subject X and object Y, and R can be explained by R', X and Y can be relaxed to X' and Y', and Z is a statement about relation R' with subject X' and object Y'.

Now let us consider the statement A8 that can be used to answer Q8 and its translation.

```
% A8: Lacking a true nucleus and the
% other membrane-enclosed organelles
% of the eukaryotic cell, the prokaryotic
```

```
% cell is much simpler in structure.

statement(s2).
statementtype(s2,rel_cause).
reltype(s2,simpler).
relattr(s2,subject,prokaryotic_cell).
relattr(s2,object,eukaryotic_cell).
causetype(s2,lacking).
causeattr(s2,object,true_nucleus).
causeattr(s2,object,other_m_e_o).
```

To use A8 to answer Q8 we need to connect 'so much smaller' with 'simpler', 'prokaryotic cell' with 'prokaryotes' and 'eukaryotic cell' with 'eukaryotes'. In addition we need to either slightly generalize the rule with answer(Q,Z) on its head or add another similar rule to address statements that are of rel_cause type.

Illustration of Answering How Questions

We illustrate our approach via two "How" questions: How do vesicles move within a cell? How is the diploid number restored after halving in meiosis? As before, we start with representing the two questions.

```
% Q4: How do vesicles move within
% a cell?

question(q4).
qtype(q4,how).
qrelation(q4,move).

qrelationattr(q4,subject,vesicle).
concept(vesicle).

qrelationattr(q4,location,cell).

% Q5: How is the diploid number
% restored after halving in meiosis?

question(q5).
qtype(q5,how).
qrelation(q5,restore).
qrelationattr(q5,object,diploid_number).
aspects(diploid_number, number).
aspects(diploid_number, diploid).

% Translating "after halving in meiosis"
precondition(q5, p1).
statement(p1).
statementtype(p1, precondition).
reltype(p1, halve).
relattr(p1, process, meiosis).

concept(diploid_number).
```

The above illustrates the representation of how questions of two forms:

- How does X happen under C, where X is an event and C is a condition.
- How is Y achieved after S, where Y is a fluent and S is a situational condition.

One may notice the similarity between the two questions and planning, in particular HTN planning (Sacerdoti 1974). In the first case X is a non-primitive perform task which has to be broken down to primitive tasks making sure that the condition C is true throughout, while in the second case Y is an achieve task that needs to be achieved from an initial state defined by S.

While the above insight will be handy in general and one can use KR and reasoning rules from existing literature (Son, Baral, and McIlraith 2001), for the particular text that we have we can answer the above questions in a simpler way. Lets first show the translation of the text that we have.

For answering question Q4 two sentences are relevant. Their representation is given below.

```
% A4: After leaving the ER, many
% transport vesicles travel to the
% Golgi apparatus.

statement(m2).
statementtype(m2,event).
reltype(m2,travel).
relattr(m2,subject,transport_vesicle).
relattr(m2,destination,golgi_apparatus).

precondition(m2, m1).
statement(m1).
statementtype(m1,precondition).
reltype(m1,leave).
relattr(m1,source,er).

aspects(transport_vesicle, vesicle).
aspects(transport_vesicle, transport).
aspects(golgi_apparatus, golgi).
aspects(golgi_apparatus, apparatus).


%A5: A Golgi stack receives and
% dispatches transport vesicles
% and the products they contain.

statement(m3).
statementtype(m3,event).
reltype(m3,receive).
relattr(m3,subject,golgi_stack).
relattr(m3,object,transport_vesicle).

statement(m4).
statementtype(m4,event).
reltype(m4,dispatch).
relattr(m4,subject,golgi_stack).
relattr(m4,object,transport_vesicle).

aspects(golgi_stack, golgi).
aspects(golgi_stack, stack).
```

Similarly, for answering question Q5 the sentence that is relevant and its representation is given below.

```
% A6:  Fertilization restores the
% diploid condition by combining
% two haploid sets of chromosomes.

statement(r1).
statementtype(r1, event).
reltype(r1, restore).
relattr(r1, subject, fertilization).
relattr(r1, object, diploid_condition).
aspects(diploid_condition,diploid).
aspects(diploid_condition,condition).
method(r1, r2).

statement(r2).
statementtype(r2, event).
reltype(r2, combine).
relattr(r2,object,two_haploid_s_of_chroms).
```

In the above representation, the fact "method(r1,r2)" specifies how the non-primitive task r1 can be broken down to simpler tasks.

Next we need to have rules that allows us to connect the use of 'vesicles' in Q4 with 'transport vesicles' in A4. This is done by the relax rules that we presented before with respect to modeling 'Why' questions with one additional rule.

```
aspects(X,Y) :- reltype(X,Y).
```

We also need to connect the event 'move' in Q4 with the events 'leave' and 'travel' in A4 and 'dispatch' in A5. Here an HTN representation would be most general, but for the simple case that we have we can add the following:

```
explainrel(move,leave).
explainrel(move,travel).
explainrelRev(move,dispatch).
explainrel(X,X) :- qrelation(Y,X).
```

In the above, the relations between move and leave, move and travel, and move and dispatch can be obtained from WordNet. Note that while 'travel' and 'leave' particular kind of 'move', 'dispatch' is an event that causes 'move'; hence the use of two different predicates.

Now we are ready to write the rules that will connect the above to answer several types of 'How' questions.

First let us consider the rules needed to answer questions of the form "How does X do R?". An example of such a question is: "How does vesicles move?" In such questions there are no objects.

```
% "How does X do R?"
entails(Z, X, R, any):-explainrel(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'),relattr(Z,subject,X'),
relax(X,X').

entails(Z, X, R, any):-explainrelRev(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'), relattr(Z,object,X'),
relax(X,X').

answer(Q,Z):-question(Q), qtype(Q,how),
qrelation(Q,R),qrelationattr(Q,subject,X),
not qrelationattr(Q,object,Y), concept(Y),
statement(Z), entails(Z, X, R, Y').
```

Similarly, to answer 'How' questions that do not have subjects we have the following rules.

```
% "How is X wrt R achieved?"
entails(Z, any, R, X):-explainrel(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'),relattr(Z,object,X'),
relax(X,X').

entails(Z, any, R, X):-explainrelRev(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'), relattr(Z,subject,X'),
relax(X,X').

answer(Q,Z):-question(Q), qtype(Q,how),
qrelation(Q,R),qrelationattr(Q,object,X),
not qrelationattr(Q,subject,Y),
concept(Y), entails(Z, Y', R, X).
```

In presence of both subject and object we have the following rules.

```
%How does X do Y wrt R?
entails(Z, X, R, Y):-explainrel(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'),relattr(Z,subject,X'),
relattr(Z,object,Y'), relax(X,X'),
relax(Y,Y').

entails(Z, Y, R, X):-explainrelRev(R,R'),
statement(Z),statementtype(Z,event),
reltype(Z,R'), relattr(Z,subject,X'),
relattr(Z,object,Y'), relax(X,X'),
relax(Y,Y').

answer(Q,Z):-question(Q), qtype(Q,how),
qrelation(Q,R),qrelationattr(Q,subject,X),
qrelationattr(Q,object,Y),entails(Z, X, R, Y).
```

Conclusion and Discussion

In this paper we gave an illustration of answering 'How' and 'Why' questions in Biology. This is a preliminary work and needs to be generalized further. One of our major concerns in this paper was that the representation of text should be such that it can be automatically obtained. We briefly mentioned research advances in Knowledge representation and reasoning and reasoning about actions and change and how that can be used in answering 'How' and 'Why' questions. We used the language of answer set programming to write the reasoning rules. One of our other concern was with respect to domain knowledge that is often needed (in addition to the knowledge that is in the given text) to answer questions. We assumed minimal availability of such knowledge and discussed automated ways to get them. We also discussed how to do superficial reasoning (thus reducing the

need for domain knowledge) via simple definitions of the relax predicate. However, stronger conclusions can be drawn by using more involved rules and domain knowledge to infer relax facts.

References

Aouladomar, F. 2005. Towards answering procedural questions. In *Proc. of KRAQ'05, an IJCAI05 workshop*.

Balduccini, M.; Baral, C.; and Lierler, Y. 2008. Knowledge representation and question answering. In Lifschitz, V.; van Harmelen, F.; and Porter, B., eds., *Handbook of Knowledge Representation*. Elsevier.

Baral, C.; Gonzalez, M.; Dzifcak, J.; and Zhou, J. 2011. Using inverse λ and generalization to translate english to formal languages. In *Proceedings of the International Conference on Computational Semantics, Oxford, England, January 2011*.

Baral, C. 2003. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press.

Blackburn, P., and Bos, J. 2005. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. Center for the Study of Language and Inf.

Bos, J.; Clark, S.; Steedman, M.; Curran, J. R.; and Hockenmaier, J. 2004. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04), Geneva, Switzerland.*

Campbell, N. A., and Reece, J. B. 2011. *Biology (9th Edition)*. Pearson Education, Inc.

Cooper, R. 1989. *Quantification and Syntactic Theory (Studies in Linguistics and Philosophy)*. Springer.

de Marneffe, M.-C.; MacCartney, B.; and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*.

Gelfond, M., and Leone, N. 2002. Logic programming and knowledge representation – the A-Prolog perspective. *Artificial Intelligence* 138(1-2):3–38.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In *ICLP*, 1070–1080.

Gunning, D.; Chaudhri, V. K.; Clark, P.; Barker, K.; Chaw, S.-Y.; Greaves, M.; Grosof, B.; Leung, A.; McDonald, D.; Mishra, S.; Pacheco, J.; Porter, B.; Spaulding, A.; Tecuci, D.; ; and Tien, J. 2010. Project halo update progress toward digital aristotle. *AI Magazine Fall 2010*.

Hobbs, J. 1985. Ontological Promiscuity. In *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, 61–69.

Marek, V., and Truszczyński, M. 1999. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-year Perspective*, 375–398.

Moldovan, D. I.; Harabagiu, S. M.; Girju, R.; Morarescu, P.; Lacatusu, V. F.; Novischi, A.; Badulescu, A.; and Bolohan, O. 2002. Lcc tools for question answering. In *TREC*.

Niemelä, I. 1999. Logic programming with stable model semantics as a constraint programming paradigm. *Annals of Mathematics and AI* 25(3,4):241–273.

Sacerdoti, E. 1974. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence* 5:115–135.

Sleator, D., and Temperley, D. 1993. Parsing english with a link grammar. In *Proc of Third International Workshop on Parsing Technologies*.

Son, T.; Baral, C.; and McIlraith, S. 2001. Planning with different forms of domain dependent control knowledge – an answer set programming approach. In *Proc. of LPNMR'01*, 226–239.

Spaulding, A.; Overholtzer, A.; Pacheco, J.; Tien, J.; Chaudhri, V. K.; Gunning, D.; and Clark, P. 2011. Inquire for ipad: A biology textbook that answers questions. In *Proc. of International Conference on AI in Education*.

Wong, Y. W., and Mooney, R. J. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of ACL.*, 960–967.

Zettlemoyer, L., and Collins, M. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *AAAI*, 658–666.