Reconstructing Hard Problems in a Human-Readable and Machine-Processable Way

Rolf Schwitter

Centre for Language Technology
Macquarie University
Sydney NSW 2109, Australia
`schwitt@ics.mq.edu.au`

Abstract. This paper shows how a controlled natural language can help to reconstruct a logic puzzle in a well-defined subset of natural language and discusses how this puzzle can then be processed and solved using a state of the art model generator. Our approach relies on a collaboration between humans and machines and bridges the gap between a (seemingly informal) problem description and an executable formal specification.

1 Introduction

Recent work in Language Technology has mainly focused on broad-coverage language processing techniques using statistical methods [2]. Hard questions about the role of structural semantics in combination with conceptual knowledge and inference for high-quality natural language understanding have been largely ignored. For example, most methods that are used in the Pascal Recognizing Text Entailment (RTE) challenge are rather shallow than deep, apart from a few exceptions that are doing quite well [3]. Without doubt many NLP applications could benefit from more complete and precise understanding of texts. Solving logic puzzles is an interesting target domain for research in structural semantics and automated reasoning since the solutions must be logically inferred and this requires a formal representation of the problem descriptions. In contrast to the RTE challenge, logic puzzles consist of more than two or three sentences, are clearly situated in a specific context, and have a clear evaluation metric since there is no disagreement about their correct solution [4]. The transition of a puzzle's informal problem description into its formal specification is rarely straightforward and requires very often major reconstructions of the original text on a case by case basis and the addition of inference-supporting background knowledge. For many people, authoring knowledge in formal logic is difficult, and I believe that providing a high-level interface language is of benefit for this user group. The interface language that I promote is a machine-oriented controlled natural language[1] that offers a number of attractive features: the language looks like English and is easy to understand by humans and easy to process by a machine; furthermore, the language is precisely defined so that it can be translated unambiguously into first-order logic and thus is in fact a formal language.

---

[1] See `http://www.ics.mq.edu.au/~rolfs/controlled-natural-languages`

T.-B. Ho and Z.-H. Zhou (Eds.): PRICAI 2008, LNAI 5351, pp. 1046–1052, 2008.

2 Einstein's Riddle

Einstein's Riddle is a logic puzzle which is said to have been invented by Albert Einstein when he was a boy in the late 1800s. Some sources claim that Einstein said that only 2 percent of the world's population can solve this puzzle. There exist several versions of the puzzle which is also known as the Zebra Puzzle but there is no hard evidence for Einstein's authorship – nevertheless the puzzle is a nice one. The version below is quoted from the first known publication [5]:

1. There are five houses.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the right of the ivory house.
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.
    Now, who drinks water? Who owns the zebra?

In the interest of clarity, it must be added that each of the five houses is painted a different color, and their inhabitants are of different national extractions, own different pets, drink different beverages and smoke different brands of American cigarettes. One other thing: In Statement 6, *right* means *your right.*

There are many **formal** solutions to this puzzle available in popular programming languages (Prolog, Lisp, C/C++) and the puzzle has also been used as a benchmark for testing automated theorem provers [7]. It is usually straightforward for a computer to solve a formalised problem description of a puzzle but it is usually hard for a human to correctly formalise a puzzle, and it is even harder for a machine to take the original natural language description as input and generate the formal representation automatically. To the best of my knowledge there exists **no language processing and reasoning tool** that takes Einstein's original text as input and produces the correct solution.

3 Logical Reconstruction of the Puzzle

Let us try to reconstruct the puzzle using a controlled natural language that supports the writing of production rules, and let us further assume that the writing process is backed up by a **predictive authoring tool** [6] that enforces the constraints of the controlled natural language. For this reconstruction process, we use the following simple grammar rules as guidelines:

```
S2 ->  [if], S1, [then], S1.         N1   ->  Adj, Noun | Noun.
S2 ->  [it,is,false,that], S1.       VP   ->  V1, Conj, V1.
S1 ->  S, Conj, S.                   VP   ->  V1.
S1 ->  S.                            V1   ->  TV, NP.
S  ->  NP, VP.                       V1   ->  Cop, RAdj, NP.
S  ->  [there,is], NP.               V1   ->  Cop, [not], RAdj, NP.
NP ->  Spec, N1, {NVar}.             V1   ->  Cop, Adj.
NP ->  MNoun | PN | NVar | Adj.      Conj ->  [and] | [or].
```

This gives us just the necessary power to reconstruct the puzzle in a way that makes the logical content of the puzzle explicit and machine-processable. Of course, current controlled natural languages have many more grammar rules but this set of rules is sufficient for our purpose.

3.1 Specifying the Background Information

People who solve this puzzle by hand apply automatically and unconsciously a surprisingly large amount of background knowledge. We explicitly specify this background knowledge by rules that convey the intended meaning. We structure this information around the two classes *person* and *house* and describe the relevant activities and properties with the help of five conditional sentences:

1b. If there is a person then the person lives in the first house or lives in the second house or lives in the third house or lives in the fourth house or lives in the fifth house.
2b. If there is a person then the person owns the dog or owns the snail or owns the horse or owns the fox or owns the zebra.
3b. If there is a person then the person drinks orange juice or drinks coffee or drinks tea or drinks milk or drinks water.
4b. If there is a person then the person smokes Kools or smokes Chesterfields or smokes Old Gold or smokes Lucky Strike or smokes Parliaments.
5b. If there is a house then the house is red or is yellow or is blue or is green or is ivory.

In the next step, we make sure that all elements are distinct since it is, for example, not possible that the Englishman and the Spaniard live in the same house or that the first house and the third house have the same colour. We can express these uniqueness conditions as follows in controlled natural language:

6b. It is false that a person X1 lives in a house and a person X2 lives in that house and X1 is not equal to X2.

This can be done in a similar way for [7b.-10b.] with *person/owns/animal*, *person/drinks/beverage*, *person/smokes/brand*, and *house/has/colour*.

So far, we did not specify any class axioms or declare any class membership. That means the theorem prover would, for example, not recognise those classes that are subsumed by the superclass *person* or those individuals that belong to the class *brand*. Subclass relationships such as *11b*, *12b* and *13b* provide the necessary class axioms and statements such as *14b* and *15b* express class membership of specific individuals:

11b. Every Englishman is a person.
12b. Every dog is an animal.
13b. Tea is a beverage.
14b. Kools is a brand.
15b. Red is a colour.

This kind of ontological information is sometimes available in a linguistic resource but there is no guarantee that this resource is complete for any application, therefore the user needs to be able to add this background information – if necessary. The original problem description uses the relations *next to* (see *11*, *12* and *15*) and *right of* (see 6) but does not specify that the houses are in a row. This information is missing and needs to be added: the relation *next to* is symmetric and can be defined with the help of the *right of* relation:

16b. If X is right of Y then X is next to Y.
17b. If Y is right of X then X is next to Y.
18b. If X is next to Y then X is right of Y or Y is right of X.

In principle, it does not matter whether the houses are counted from left to right or from right to left, what is important is the order but not the direction. Therefore, we assume that the fifth house is the rightmost one and specify:

19b. The fifth house is right of the fourth house.
20b. The fourth house is right of the third house.
21b. The third house is right of the second house.
22b. The second house is right of the first house.

As we can see, an automatic solution of the puzzle needs a substantial amount of nontrivial background information (*1b-22b*) that is not available in textual form from the original problem description.

3.2 Specifying the Premises

Once we have specified the required background information, we can consider the premises of the original puzzle in more detail and start with those sentences which provide factual information (see *3*, *5*, *10* and *14*). These sentences do not require any reconstruction and are simply repeated here:

3. The Spaniard owns the dog.
5. The Ukrainian drinks tea.
10. The Norwegian lives in the first house.
14. The Japanese smokes Parliaments.

Now, we take all pair wise clues of the original puzzle into consideration (see *2*, *4*, *7*, *8*, *9* and *13*) and reconstruct them as conditional sentences since they are in essence rules:

2r. If the Englishman lives in a house then the house is red.
4r. If a person drinks coffee and lives in a house then the house is green.

7r. If a person smokes Old Gold then the person owns a snail.
8r. If a person smokes Kools and the person lives in a house then the house is yellow.
9r. If a person drinks milk then the person lives in the third house.
13r. If a person smokes Lucky Strike then the person drinks orange juice.

In the next step, we consider the neighborhood clues (see *6*, *11*, *12*, and *15*) and express them with the help of the following conditional sentences:

6r. If a house X1 is green and a house X2 is ivory then X1 is right of X2.
11r. If a person X1 owns the fox and lives in a house Y1 and a person X2 smokes Chesterfields and lives in a house Y2 then Y1 is next to Y2.
12r. If a person X1 smokes Kools and lives in a house Y1 and a person X2 owns the horse and lives in a house Y2 then Y1 is next to Y2.
15r. If the Norwegian lives in a house Y1 and Y1 is next to a house Y2 then Y2 is blue.

As these examples show, variables are used instead of pronouns and provide a precise and ambiguity-free replacement for them. These variables are always introduced together with a noun and can be used afterwards like personal pronouns. Finally, we specify the two questions: *Who drinks water?* and *Who owns the zebra?*. These are simple questions but we can query all aspects of the puzzle.

4 Solving the Puzzle

The puzzle is translated with the help of a logic grammar into TPTP notation [7] and then into the input format of E-KRHyper [1]. E-KRHyper is a model generator for first-order logic with equality and uses clauses of the form:

```
HEAD :- BODY.
```

The head is either false or consists of a conjunction or a disjunction of positive literals. The body is either true (in the case of facts) or consists of a conjunction of body literals. Body literals are composed of negative literals and a number of special logical operators, among them stratified negation as failure (`\+`). Here are a few clauses which illustrate the input format to E-KRHyper. For example, the translation of sentence *1b* results in a disjunctive clause of the form:

```
live_in(P,1) ; live_in(P,2) ; live_in(P,3) ; live_in(P,4) ;
live_in(P,5) :- person(P).
```

and five facts for the definite noun phrases:

```
house(1). house(2). house(3). house(4). house(5).
```

The translation of sentence *6b* leads to an integrity constraint where the head is false and the body consists of a number of negative literals with a negation as failure operator for testing the inequality of the two variables `P1` and `P2`:

```
false :- person(P1), house(H), live_in(P1,H), person(P2),
live_in(P2,H), \+(P1=P2).
```

The translation of sentence *2r* which represents a pair wise clue results in the following definite clause with one positive literal in the head:

```
prop(H,red) :- house(H), live_in(E,H), englishman(E).
```

Finally, the translation of the two questions results in two conjunctive queries that can be added directly to the theory:

```
answer(Who) :- drink(Who,W), water(W).
answer(Who) :- own(Who,Z), zebra(Z).
```

E-KRHyper takes the entire theory in clausal form as input and generates a finite model consisting of a set of ground atoms that satisfy the clauses, for example:

```
own(j,z). own(n,f). zebra(z). water(w). coffee(c).
drink(j,c). drink(n,w). japanese(j). norwegian(n).
```

During the model generation process the variables of the answer literals are instantiated and form answers to the questions. Thus we find out that the Norwegian drinks water and that the Japanese owns the zebra.

5 Conclusions

There exists **no** natural language processing system that can take the original version of Einstein's Riddle as input and find the correct solution automatically. Without doubt most natural language processing applications which aim at high-quality understanding of texts would benefit from more precise structural semantic knowledge, conceptual knowledge and inference. But as long as it is not possible to solve these kinds of puzzles fully automatically, we will not see significant progress in axiom-based natural language understanding. It might be wiser to shift the focus and bring the user back into the loop and to use a controlled natural language together with an authoring tool that allows the user to work **in cooperation** with a machine in order to solve hard problems.

Acknowledgments

I am grateful for the constructive comments made by Norbert E. Fuchs on earlier versions of this paper and for the feedback from four anonymous reviewers.

References

1. Baumgartner, P., Furbach, U., Pelzer, B.: Hyper Tableaux with Equality. In: Fachberichte Informatik, 12-2007, Universität Koblenz-Landau (2007)
2. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. Computational Linguistics 29(4), 589–637 (2003)

3. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognizing Textual Entailment Challenge. In: Proceedings of the First PASCAL Challenges on Recognising Textual Entailment, pp. 1–8 (2005)
4. Lev, I., MacCartney, B., Manning, C.D., Levy, R.: Solving Logic Puzzles: From Robust Processing to Precise Semantics. In: Proceedings of the 2nd Workshop on Text Meaning and Interpretation at ACL 2004, pp. 9–16 (2004)
5. Life International: Who Owns the Zebra? Life International magazine 17, 95 (December 1962)
6. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE – A Look-ahead Editor for a Controlled Language. In: Proceedings of EAMT-CLAW 2003, pp. 141–150 (2003)
7. Sutcliffe, G., Suttner, C.B.: The TPTP Problem Library: CNF Release v1.2.1. Journal of Automated Reasoning 21(2), 177–203 (1998)