Chapter 6

Related work

A number of formal logical languages was used in the past to formalize domains with common sense involved. For instance, in [Mueller, 2004], the author uses discrete event calculus to understand real stories involving kidnapping and terrorist attacks. Mueller also develops a commonsense knowledge base, but he uses different techniques from the ones presented in this work.

In addition, other domains have been formalized in ASP, such as the travel domain (see [Gelfond, 2006]). Although different in first glance, this domain could be related to the ideological conflict domain in cases, where traveling between countries is involved.

Moreover, there is a recent call to build micro-theories using non-monotonic reasoning. In fact, building micro-theories, learning how to expand them and how to combine them in larger modules, is one of the most interesting challenges we are facing now.

Recently, a group of scientists have been working on an *Influence Net* modeling for analyzing the causal relations of complex situations [Rosen and Smith, 1998]. They use the combination of two established methods of decision analysis: Bayesian inference net analysis, originally employed by the mathematical community; and influence diagramming techniques, originally employed by operations researchers. In that work, the term conflict includes situations of economic instability, ideological or cultural contrasts, as well as the more traditional political and diplomatic security concerns. The authors state that if these crisis situations are left untended, they tend toward armed conflict situations that affect the global stability.

In the next section, we review the work of Erik Mueller, which is very close to our work.

6.1 The work of Erik Mueller

Erik Mueller works at IBM Thomas J. Watson Research Center. He is using commonsense reasoning in understanding script-based stories. He considers that the problem of constructing a system that take stories as input and understand them, is an important one. There are many applications, which need the ability to understand stories. Examples are advisory, dialogue, filtering, information retrieval, question answering, and summarization systems. However, understanding a story is difficult to automate, because some of the details are not stated in the story and need to be extracted by the system itself [Mueller, 2004].

The existing story understanding systems can be separated into groups according to their:

1. breadth of coverage,
2. depth of understanding,
3. ability to handle new stories, and
4. ability to handle real-world input such as text and speech.

The breadth of coverage can be *narrow* or *broad*. Narrow coverage systems deal with stories about a particular domain. On the other hand, broad coverage systems deal with stories that encircle many domains.

The depth of understanding can be *shallow* and *deep*. Shallow understanding systems build a superficial understanding of the story, while deep understanding systems build a deep understanding.

Some systems can handle only the story they are built for and must be changed in order to understand different stories. On the other side, there are systems that can handle new stories.

Finally, some systems are able to deal with naturally occurring text or speech, while other systems require edited text as an input.

In [Mueller, 2004], the author presents a system for deep understanding that handles new and real stories. However, the system is narrow coverage system, because it only deals with stories involving particular scripts. A script is define as a stereotypical activity consisting of two or more events. They use the classical logic event calculus and an efficient, satisfiability-based method for reasoning in the event calculus. As we described in chapter 1, the system works as follows:

1. The input is a template about a terrorism event produced by an information extraction system.
2. The script classifier decides what script is active in the template.
3. The reasoning problem builder specific to the script converts the template into a commonsense reasoning problem.
4. The commonsense reasoner takes the problem and a commonsense knowledge base and produces a model that deeply represents the story.

6.2 Building commonsense reasoning problems

In this section we present how Erik Muellers system constructs commonsense reasoning problems from MUC [1] terrorism templates. The script classifier takes the template and decides which script is active in the template. The available scripts are: Arson, Kidnapping, ShootingAttack, or TimeDelayBombing. After that, a reasoning problem builder, specific to the active script, constructs a commonsense reasoning problem. A commonsense reasoning problem consists of an initial state, optional intermediate states, and a narrative of event occurrences. In the next step, the commonsense reasoning problem is fed to the commonsense reasoner. As a result, the reasoner uses the axioms of the commonsense knowledge base to make inferences and form detailed models [Mueller, 2004].

But how are the semantics of the template, script, and event calculus formulas of the reasoning problem related? The template determines the script, which determines which event calculus reasoning problem builder is used. The template specifies the roles of the script. Each script role is represented as a logical constant in the event calculus. In addition, the template specifies the stage of execution of the script, which determines how many events of the script are represented as having occurred in the event calculus. As an example, consider the following initial state and narrative, that have been generated for the Arson script:

1. Initially the physical targets are at a first location and the perpetrator is at a second location.
2. Initially the physical targets are intact.
3. The perpetrator walks to the first location.
4. The perpetrator sets fire to the physical targets.

[1] Message Understanding Conferences

5. The perpetrator walks back to the second location.

6.3 Application to question answering

A question answering system combines the results of a collection of question answering specialists, each of which uses different knowledge source or answer finding strategy. In this context, the Mueller's system could serve as a script-based understanding specialist and could be added to a question answering system. The following algorithm can be used to answer yes-no questions about space, using the final models produced by the system.

1. Given the question "Was/were actor a **present** when e?":
2. If for every time point t at which e occurs, the location of a at time point t is equal to the location of the actor of e at time point t, answer "Yes".
3. Otherwise, if for every time point t at which e occurs, the location of a at time point t is not equal to the location of the actor of e at time point t, answer "No."
4. Otherwise, answer "Some of the time".

For instance: *Question*: Was Miguel Jiménez **present** when the four explosive charges exploded? *Answer*: Yes.

In addition, question about time can be answered, following these rule:

1. Given the question "f **before** e?", where f is a fluent and e is an event:
2. If f is true for all time points less than or equal to t, answer "Yes".
3. Otherwise, if f is false for all time points less than or equal to t, answer "No".
4. Given the question "f **after** e?", where f is a fluent and e is an event:
5. If f is true for all time points greater than t, answer "Yes".

6. Otherwise, if f is false for all time points greater than t, answer "No".

For example: *Question*: Were the youths angry at the terrorists **after** the terrorists threatened the youths? *Answer*: Yes.

6.4 Closing remarks

Erik Mueller's system uses commonsense reasoning to understand texts involving scripts. However, the system has some limitations and the most critic is that it may produce inaccurate models [Mueller, 2004]. This happens when the MUC templates do not contain all the information needed by the reasoning problem builder and commonsense reasoner. A solution to this problem, is to explicitly add the missing information to the templates [Mueller, 2004]. On the other hand, there is no evidence that the representations of the travel domain, the zoo problem, and the space shuttle obtain inaccurate models, when using ASP.

Another limitation of the system is that it is unable to deal with uncertainty. We know that in commonsense reasoning very often we need to deal with uncertainty. Moreover, opens problems remain. How far can Mueller's approach be taken? To what degree can story understanding be performed using scripts?

In order to achieve deep understanding, Mueller proposes that the commonsense reasoning modules be assembled along with information extraction modules. Again, the concept of commonsense reasoning plays an important role. In fact, commonsense reasoning can be applied to the problem of understanding an unlimited number of real-world texts involving scripts, as shown in [Mueller, 2004]. However, much remains to be done in order to build a system that can produce highly accurate models.