Available online at www.sciencedirect.com

ScienceDirect

Journal of Applied Logic 7 (2009) 100–112

JOURNAL OF APPLIED LOGIC

www.elsevier.com/locate/jal

Applying automated deduction to natural language understanding

Johan Bos

University of Rome "La Sapienza", Italy

Available online 13 July 2007

Abstract

Very few natural language understanding applications employ methods from automated deduction. This is mainly because (i) a high level of interdisciplinary knowledge is required, (ii) there is a huge gap between formal semantic theory and practical implementation, and (iii) statistical rather than symbolic approaches dominate the current trends in natural language processing. Moreover, abduction rather than deduction is generally viewed as a promising way to apply reasoning in natural language understanding. We describe three applications where we show how first-order theorem proving and finite model construction can efficiently be employed in language understanding.

The first is a text understanding system building semantic representations of texts, developed in the late 1990s. Theorem provers are here used to signal inconsistent interpretations and to check whether new contributions to the discourse are informative or not. This application shows that it is feasible to use general-purpose theorem provers for first-order logic, and that it pays off to use a battery of different inference engines as in practice they complement each other in terms of performance.

The second application is a spoken-dialogue interface to a mobile robot and an automated home. We use the first-order theorem prover SPASS for checking inconsistencies and newness of information, but the inference tasks are complemented with the finite model builder MACE used in parallel to the prover. The model builder is used to check for satisfiability of the input; in addition, the produced finite and minimal models are used to determine the actions that the robot or automated house has to execute. When the semantic representation of the dialogue as well as the number of objects in the context are kept fairly small, response times are acceptable to human users.

The third demonstration of successful use of first-order inference engines comes from the task of recognising entailment between two (short) texts. We run a robust parser producing semantic representations for both texts, and use the theorem prover VAMPIRE to check whether one text entails the other. For many examples it is hard to compute the appropriate background knowledge in order to produce a proof, and the model builders MACE and PARADOX are used to estimate the likelihood of an entailment.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Automated reasoning; Natural language understanding; First-order logic; Theorem proving; Model building

1. Introduction

Since the mid 1990s I've been using tools from automated deduction, mainly general-purpose theorem provers and model builders for first-order logic, to solve problems in natural language understanding.[1] I've done this both from the perspective of *computational linguistics* (testing and improving a linguistic theory by means of a computational

E-mail address: bos@di.uniroma1.it.

[1] The concept *natural language understanding* is not easy to define. Roughly, I take it to mean the capability to arrive at consistent semantic interpretations of texts and the ability to draw correct inferences from texts, such as recognising whether one text follows from another.

1570-8683/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jal.2007.07.008

implementation) as well as that of *natural language processing* (using inference in applications such as question answering and recognising textual entailment). In this article I will present three applications where these tools were successfully employed, with the aim of convincing researchers—both from the natural language processing and the automated deduction communities—that automated inference engines for first-order logic can be efficiently applied to natural language understanding.

For an outsider it might seem obvious that one needs some form of computerised reasoning to model natural language understanding. However, it is quite surprising to see how few tools from automated reasoning have actually made it in linguistic applications, and I believe there are some specific reasons for this.

First of all, a rather high level of cross-disciplinarity is required. Knowledge about linguistics (and in particular formal semantics) is required, but also about natural language processing, logic, lexical resources, knowledge representation, and automated reasoning. As a matter of fact, only few researchers feel comfortable in the intersection of these areas, and collaborations between researchers of these fields are rare. If anything, comparing the current situation with that of the late 1980s and early 1990s, it seems that less and less research is devoted to this topic.

Secondly, there is an enormous gap between formal semantic theory and practical implementation. There is a vast amount of theory on the semantics of natural language from formal linguistics and philosophy. However, most of it does not lead easily to computational implementation. Many of the linguistic phenomena that are accounted for in formal theory are quite rare in naturally occurring data, so a natural language engineer won't gain much performance in his system when taking formal theory into account. But more crucially, from an automated reasoning point of view, almost all semantic phenomena are formalised in higher-order logic (a trend set by no-one less than the celebrated philosopher Richard Montague, who pioneered the logical approach to natural language semantics [18]). As higher-order logic allows quantification over predicates, it is extremely challenging to write an efficient theorem prover for it. In contrast, various—relatively efficient—theorem provers exist for first-order logic. But transforming higher-order theories of natural language semantics into interesting first-order fragments or approximations is often far from trivial, and attempts to combine theories that focus on specific linguistic phenomena usually give rise to logical compatibility problems [30].

Thirdly, it is not trendy to use symbolic approaches such as theorem proving in natural language processing. In contrast, stochastic approaches dominate the field nowadays, and after having been successfully applied to speech recognition and syntactic processing, it probably won't take long until statistics will dominate the field of semantic processing and inference as well. However, I am often surprised by the shared view that using theorem proving in natural language understanding is classified as the "traditional approach". As I will show below in a discussion of previous work, one can hardly talk of a tradition. Even though applying automated deduction was tried in the early years of natural language processing, it never got to work well, for the following, obvious, reasons: first, theorem proving hadn't matured into a state as we know it today; and second, machines lacked the memory and speed to perform the computations required by inference in reasonable time. As of now, it is probably fair to say that we are only starting to understand the limitations and appreciate the added value of computerised reasoning in the semantic interpretation of natural language.

1.1. Previous and related work

This article is mainly concerned with applying methods of automated deduction (more precisely, first-order theorem proving and finite model generation) to natural language semantics. The idea of using deduction in natural language understanding originated already in the early 1960s, especially in the context of automated question answering [2,21]. Research on theorem proving was given a big push by the development of Robinson's resolution algorithm [32], and AI researchers were attracted to the view of using a nice, clean, general representation language (i.e., first-order logic) to model human language and commonsense knowledge. But because there was so little space to add heuristic rules in the search for proofs, soon after, in the beginning of the 1970s, the focus of inference engines was shifted to procedural deductive systems. Two influential examples of this branch of research are Winograd's blocks world application using the inference engine Planner [38], and Charniak's work on modelling children's story comprehension with Micro-Planner [11]). It is perhaps interesting to note that Planner and Micro-Planner are often regarded as the precursors of the logic programming language Prolog [12].

After these pioneering attempts, research evolved into two main streams: one following the initial ideas of Charniak and Winograd of using incomplete Prolog-style reasoning (for instance, see [16]), the other giving up the idea of *deduction* as the principal form of inference but rather view *abduction* as the way forward. Prime examples of the latter are [12,22,24] for story comprehension and discourse processing.

The work described here follows mostly that of the first tradition, namely using first-order logic as the representation language, and automated deduction methods as a way to implement reasoning. But unlike most previous work, we will make use of general-purpose, complete inference engines for first-order logic, rather than incomplete reasoners such as Planner and Prolog. We will also view finite model building as a way to complement theorem proving, by using theorem proving and model building in parallel to solve inference problems.

1.2. Outline of this article

This article is devoted to three applications demonstrating successful use of classical first-order inference tools in natural language understanding. The first of these applications is a text-understanding system that calculates presuppositions of English sentences and performs consistency and informativeness checks on texts using a battery of theorem provers. The second is a spoken dialogue system, interfaced to a mobile robot and an automated home environment, that uses theorem proving and model building for planning its linguistic and non-linguistic actions. The third is a system for recognising textual entailment.

I will discuss the reasoning tools that were used, how they were used, what added value they provided, and what their limitations were. The current article serves mainly as a survey of these issues. For technical details of these applications I refer the reader to the papers mentioned throughout the text. I will also outline, at the end of this article, a series of unresolved issues whose solutions would contribute to a better match of theorem proving techniques and natural language understanding in general, and computational semantics in particular.

2. Presupposition projection

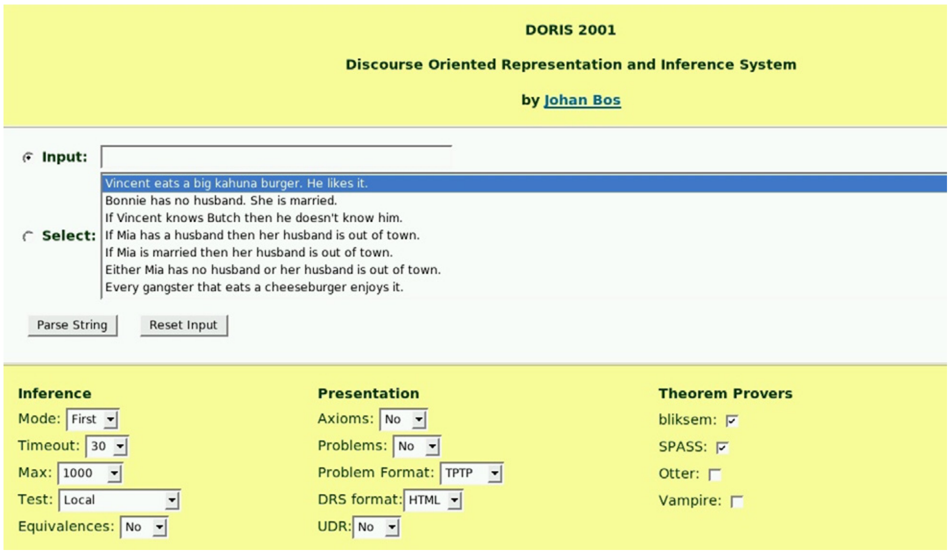
In the mid 1990s I started to implement tools for the semantic analysis of English texts, as part of my thesis work at the University of the Saarland in Saarbrücken, Germany. One of the aims was to follow formal linguistic theory as closely as possible and see how much of it could be implemented straight away. As a starting point I took Discourse Representation Theory (DRT), initially developed by the philosopher Hans Kamp in the 1980s [26]. Initially, DRT was developed to allow existential quantifiers to bind variables dynamically, solving some long-standing linguistic puzzles to do with pronouns and quantifiers. Over the years, DRT developed into a well-documented theory covering a wide range of linguistic phenomena in a unified framework, which was one of the main motivations for choosing it as the basis for semantic formalism. DRT is viewed as a representational theory of semantics, using a box-like representation called Discourse Representation Structure to represent the meaning of natural language expressions [25]. In particular, I was interested in modelling the behaviour of presuppositions in texts, and aimed at implementing an influential proposal by Rob van der Sandt, whose theory of presupposition was cast in DRT [35].[2]

My implementation efforts resulted in the DORIS system (Fig. 1). This system was able to parse English sentences and compute a discourse representation structure (as formulated by DRT) of the input text. The linguistic phenomena covered by the DORIS system included pronouns, quantifier and negation scope, and presuppositions [7]. It had a reasonable grammar coverage: substantially more than a toy grammar, but certainly not reaching the level of today's wide coverage grammars.

Presupposition is a linguistic phenomenon, and can be explained as a proposition that is taken for granted by the speaker. A commonly used test to check whether p presupposes q if both p and $\neg p$ entail q. Presuppositions are known to be "triggered" by certain words or phrases. For instance, the noun phrase "Mia's husband" presupposes that Mia is married, and that Mia is a woman, because the sentence "Jody likes Mia's husband" and its negated form "Jody doesn't like Mia's husband" both entail that Mia is married and is a woman.

An interesting aspect of presuppositions is that they are sometimes neutralised by the linguistic context, and that it is quite hard to pin down exactly when they are and when they are not, especially in sentences that contain some form of implication, negation, or disjunction. Consider, for instance, the following three sentences (with the relevant presupposition trigger typeset in bold face):

[2] There are many different approaches to explaining presupposition, ranging from default logics and partial logics to pragmatic theories. In the scope of this article I restrict myself to Van der Sandt's work, and have no intention to compare it with other theories of presupposition.



DORIS 2001

Discourse Oriented Representation and Inference System

by Johan Bos

Input:

Select:
Vincent eats a big kahuna burger. He likes it.
Bonnie has no husband. She is married.
If Vincent knows Butch then he doesn't know him.
If Mia has a husband then her husband is out of town.
If Mia is married then her husband is out of town.
Either Mia has no husband or her husband is out of town.
Every gangster that eats a cheeseburger enjoys it.

Parse String | Reset Input

Inference	**Presentation**	**Theorem Provers**
Mode: First	Axioms: No	bliksem: ✓
Timeout: 30	Problems: No	SPASS: ✓
Max: 1000	Problem Format: TPTP	Otter: ☐
Test: Local	DRS format: HTML	Vampire: ☐
Equivalences: No	UDR: No	

Fig. 1. Screenshot of the DORIS system. In the upper window users can type or select an English sentence. The lower window provides several parameters, for instance the selection of various theorem provers.

(1) If Mia is dating Vincent, then **her husband** is out of town.
(2) If Mia has a husband, then **her husband** is out of town.
(3) If Mia is married, then **her husband** is out of town.

Here (1) presupposes that Mia is married, but (2) and (3) do not. This is strange, as all sentence exhibit the same syntactic structure. Van der Sandt's theory explains this by constructing various possibilities of positioning the presupposition, and then checking whether they are acceptable by posing a set of acceptability constraints upon them. These constraints ensure that the text has a consistent and informative interpretation. Furthermore, they are applied both to the global and local levels of discourse. Let me illustrate Van der Sandt's theory by looking at an example.

For (3), there are two positions where the presupposition can "land". The first possibility corresponds to the interpretation where the proposition that Mia has a husband is presupposed, since the presupposition is projected on the global level of discourse. This is a wrong interpretation, for sentence (3) does not presuppose that Mia has a husband. The second possibility describes the situation where the presupposition is neutralised, since it is accommodated on a local level of discourse. This corresponds to the correct interpretation. Hence, the acceptability constraints should rule out the first possibility, and support the latter.

Below, the discourse representation structures generated for these two possibilities are shown, including paraphrases in English of the meaning they express, with the presupposition trigger phrase set in bold face, and the triggered presupposition underlined.

(3-a)
$$\begin{array}{|c|}\hline x\ y \\ \hline \text{mia}(x) \\ \text{has}(x,y) \\ \text{husband}(y) \\ \begin{array}{|c|}\hline \\ \hline \text{married}(x) \\ \hline\end{array} \Rightarrow \begin{array}{|c|}\hline \\ \hline \text{outoftown}(y) \\ \hline\end{array} \\ \\ \hline\end{array}$$

"<u>Mia has a husband</u>. If Mia is married, then **her husband** is out of town".

(3-b)
$$\begin{array}{|c|}\hline \mathrm{x} \\ \hline \mathrm{mia(x)} \\ \begin{array}{|c|}\hline \mathrm{y} \\ \hline \mathrm{married(x)} \\ \mathrm{has(x,y)} \\ \mathrm{husband(y)} \\ \hline \end{array} \Rightarrow \begin{array}{|c|}\hline \\ \hline \mathrm{outoftown(y)} \\ \hline \end{array} \\ \\ \hline \end{array}$$

"If Mia is married and has a husband, then **her husband** is out of town".

After generating these possible interpretations, the acceptability constraints are applied to them. The first solution, shown in (3-a) above, violates a constraint: the antecedent of the conditional is not locally informative with respect to its global context: if Mia has a husband, then the fact that she is married is not perceived as new information, given the background knowledge that if someone has a husband, she is married. The second solution, pictured in (3-b), satisfies all acceptability constraints: it is consistent and informative, both on the global and local levels. As a consequence, interpretation (3-a) is rejected, and (3-b) is accepted as possible interpretation. Note that, for example (1), global accommodation of the presupposition would not lead to a violation of the informativeness constraint.

Despite the adequacy of the predictions of the theory, there was still a practical problem when trying to implement it in DORIS: the acceptability constraints require logical inference. Even though DRT was an established semantic theory backed up with a model-theory, there were no (efficient) theorem provers available that could reason with the discourse representation structures employed by DRT.[3] Discussions with Patrick Blackburn and Michael Kohlhase (both in Saarbrücken, at the time) developed the idea of using first-order theorem provers to implement Van der Sandt's acceptability constraints, with the help of a translation function from the DRT language to ordinary first-order formula syntax [4]. This translation function, $(.)^{d2f}$, is recursively defined as follows:

$$\left(\begin{array}{|c|}\hline \mathrm{x}_1,\ldots,\mathrm{x}_n \\ \hline \gamma_1 \\ \vdots \\ \gamma_m \\ \hline \end{array}\right)^{d2f} = \exists \mathrm{x}_1 \cdots \exists \mathrm{x}_n ((\gamma_1)^{d2f} \wedge \cdots \wedge (\gamma_m)^{d2f})$$

$$(R(\mathrm{x}_1,\ldots,\mathrm{x}_n))^{d2f} = R(\mathrm{x}_1,\ldots,\mathrm{x}_n)$$
$$(\tau_1 = \tau_2)^{d2f} = \tau_1 = \tau_2$$
$$(\neg B)^{d2f} = \neg(B)^{d2f}$$
$$(B_1 \vee B_2)^{d2f} = (B_1)^{d2f} \vee (B_2)^{d2f}$$

$$\left(\begin{array}{|c|}\hline x_1,\ldots,x_n \\ \hline \gamma_1 \\ \vdots \\ \gamma_m \\ \hline \end{array} \Rightarrow B\right)^{d2f} = \forall \mathrm{x}_1 \cdots \forall \mathrm{x}_n \big(((\gamma_1)^{d2f} \wedge \cdots \wedge (\gamma_m)^{d2f}) \rightarrow (B)^{d2f}\big)$$

With this translation at our disposal, all we needed was a theorem prover for first-order logic. The theorem prover BLIKSEM, developed by Hans de Nivelle, was among the first that was put to the test [17]. And the first results looked promising: BLIKSEM could handle most of the presupposition problems given to it, in reasonable time.

However, as some natural language examples could cause hundreds of consistency checking tasks (due to a combinatorial explosion of linguistic ambiguities), it took a long time before BLIKSEM had dealt with them all. This problem was solved with the help of the MATHWEB middleware, a collection of web-based inference services [20].

[3] It is unlikely that specialized inference engines would help at all for the core fragment of DRT, which is equivalent to first-order logic. However, one could think of specialized provers that go beyond the first-order DRT fragment.

What MATHWEB did was farming out a set of inference problems to different computers using a common software bus. Using the Internet and many machines around the world (I recall that there were machines running in Edinburgh, Budapest, Saarbrücken, and Sydney, among other sites), MathWeb could basically be viewed as a parallel supercomputer.[4]

To cut a long story short, DORIS was interfaced directly to MATHWEB, and many different theorem provers for first-order logic were added: SPASS [37], FDPLL [1], OTTER [29], and VAMPIRE [31]. This experiment showed that there was a diversity in the way inference engines dealt with the inference problems generated by DORIS: some provers were better at satisfiable problems, others were faster at small problems, some problems were problematic for all provers, and so on, backing up the idea that it is a good idea to use a collection of various theorem provers with different strategies in a parallel environment.

In sum, the DORIS system was a first practical demonstration that first-order inference can play an interesting role in natural language processing, albeit with limitations [4]. As a side-effect, it generated a new application area for automated deduction. As a matter of fact, many of the problems generated by DORIS were donated to the TPTP collection [34]. The use of off-the-shelf first-order reasoning tools also shed a whole new light on research in computational semantics—one could not just build a semantic representation for a given natural language expression (which was what the state-of-the-art reflected), but also reason with the result [3].

Incidentally, DORIS also helped to precisely formulate the acceptability constraints of Van der Sandt's theory of presupposition projection [5]. The global constraints on consistency and informativeness were, using the translation function, relatively straightforward to formulate (see also [3]). The local constraints, however, were originally rather informally presented in Van der Sandt's paper [35], which did not lead straightforwardly to computational implementation. Several theoretical attempts to precisely formulate them shipwrecked on this rock. Using DORIS and MATHWEB as a testbed, a new definition of the local constraints was successfully developed and evaluated [5].

Despite these promising results, there were several obvious limitations in using first-order theorem proving for understanding natural language texts. Scalability was one of them. A state-of-the-art theorem prover would do well on a couple of sentences, but—not surprisingly given the computational properties of first-order logic—would just "choke" on larger texts. Linguistic coverage was another limitation. Some linguistic phenomena require richer semantic representations and therefore harder problems. For instance, tense and aspect require a richer sortal hierarchy, cardinal expression require counting, and plural noun phrases require elements of set theory. Formulating first-order approximations of the meaning of these linguistic phenomena results in significantly harder problems for theorem provers.

The last version of DORIS was released in 2001 [7]. Although I still view it as an important step in the development of computational semantics in general, and the implementation of Discourse Representation Theory in particular, the limited grammatical coverage of DORIS and its unfocused application domain left it without a future. At the time I thought that it would take at least twenty years to develop a parser that achieved both wide-coverage *and* syntactic representations of enough detail to construct meaningful semantic representations (I was, fortunately, very wrong! See Section 4). In order to reach a new level of sophistication in computational semantics, I instead focused on small domains, in which the grammatical coverage and necessary background knowledge could be specified a-priory. Human-computer dialogue systems turned out to be the killer application for pursuing this strategy.

3. Spoken dialogue systems

At the University of Edinburgh I was involved in developing a spoken dialogue system which was interfaced to a (real) robot. The robot in question was a RWI Magellan Pro robot, with sonars, infrared sensors, bumpers, and a video camera (Fig. 2). It was equipped with an on-board computer connected to the local network via a wireless LAN interface. The robot moved about at the basement of Buccleuch Place, and people could direct it, ask it questions, or provide it with new information, all via speech. A typical conversation could be:

Human: Robot?
Robot: Yes?

[4] This sounds perhaps quite ordinary today, but at the time it was a sensation.

Fig. 2. The talking robot, using a first-order theorem prover and finite model builder to plan its utterances, engaged in conversation with roboticist Tetsushi Oka (Buccleuch Place, Edinburgh, 2001).

Human: Where are you?
Robot: I am in the hallway.
Human: OK. Go to the rest room!

Such kinds of dialogues were relatively straightforward to model with the then state-of-the-art in human-machine dialogue. Yet, the robot was still "semantically challenged": it had no means to draw inferences. What I aimed to do was use components of the DORIS system to give the robot means to perform consistency checking, answer questions, and calculate its next actions. In particular, I was interested in letting the robot react to inconsistent information or obvious information, resulting in dialogues such as:

Human: Where are you?
Robot: I am in the hallway.
Human: You are in my office.
Robot: No, that's not true!
Human: You are in the hallway.
Robot: Yes, I know.

I knew this was feasible because of the DORIS experience: Theorem provers can rather easily handle the amount of information necessary for this type of dialogue. Moreover, the required background knowledge, given the limited domain and environment, was relatively easy to compute and maintain. Indeed, we succeeded in building a robot that checked whether each assertion of the user was consistent and informative with its current state and knowledge of the dialogue. We only used one theorem prover, and took the first interpretation of the user's utterance that was consistent and informative as the correct one. This turned out to work well in practice. Our choice for theorem prover was SPASS [37], one of the strongest theorem provers at the time that was freely available for research purposes.

After having accomplished this, we were interested in using first-order theorem proving for planning the actions of a command given by a human to the robot. This was done primarily from a semantic point of view. We considered commands that involved negation, disjunction, or quantification, because they pose interesting challenges on a correct interpretation. Examples of utterances that illustrate this are:

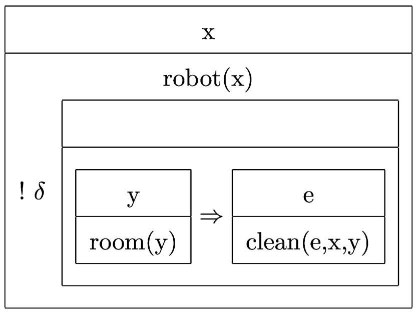
(4) Turn on a light.
(5) Clean every room.
(6) Switch on every light in the hallway.

(7) Turn off every light except the light in the office.
(8) Go to the kitchen or the rest room.

On hearing (4), the robot had to turn on a light that was currently switched off, and it had to complain when all the lights were already on. On (5), it had to clean all rooms that were currently dirty (some of the rooms could be clean already). On (6), it should only consider lights in the hallway, in other words, it needs to deal correctly with restrictive quantification. On (7), it should consider all lights minus those in the office, taking the implicit negation expressed by the preposition "except" into account. On (8), it should either go to the kitchen or the rest room.

To illustrate the problem, consider the discourse representation structure constructed for example (5):

(9)



Because discourse representation structures are recursively defined, they are hard to use directly to determine the primitive actions for the robot. In (9), the number of actions requested to the robot depends on the number of rooms in the domain.[5] With a theorem prover this is hard to find out. The only thing a theorem prover could tell us was whether a statement was inconsistent or uninformative. Even although some of the theorem provers could deal with satisfiable problems reasonably well, their output was hard to translate into something that could form the basis of a low-level action describing the robot's movements.

Instead, this seemed a natural task for a finite model builder though. Via the DORIS system I already came into contact with Karsten Konrad's model builder KIMBA [27], but I had never used model builders other than for checking satisfiability. We started using Bill McCune's MACE because it searched models by iteration over domain size, and generally generating models that were both domain-minimal and minimal in the extensions of the predicates [28]. An example output of Mace when given a modal first-order translation of (9) is shown below.[6]

```
D = {d1,d2,d3,d4,d5,d6,d7,d8}
F(possible_world) = {d1,d2,d3}
F(robot) = {(d1,d4),(d2,d4),(d3,d4)}
F(room) = {(d1,d5),(d2,d5),(d3,d5),
           (d1,d6),(d2,d6),(d3,d6),
           (d1,d7),(d2,d7),(d3,d7)}
F(action) = {(d1,d2,d3)}
F(clean_object) = {(d2,d5),(d2,d6),(d2,d7)}
F(is_dirty) = {(d1,d5),(d1,d6),(d1,d7)}
F(is_clean) = {(d3,d5),(d3,d6),(d3,d7)}
```

Here the domain contains three rooms. Information about the domain is expressed in the background knowledge given in addition to the theorem prover and model builder. Because MACE produces minimal models, no redundant

[5] Here the DRS-language is extended by two operators. The δ operator takes a DRS as argument denoting an action term. The ! operator takes an action term as argument forming a DRS-condition denoting an action that has to be executed.

[6] This translation function translates formulas with respect to variables ranging over possible worlds. For instance, given the current world w, a modal formula $\Box(p(x) \wedge q(x))$ is translated as $\forall v(R(w,v) \rightarrow (p(v,x) \wedge q(v,x))$, where R is an accessibility relation over worlds. Each n-ary relation is translated into a relation of $n+1$ arguments, the extra argument playing the role of current possible world (or "state", if you prefer). Action terms are translated into three-place relations, where the first argument is the current world, the second argument the world during execution of the action, and the third argument the resulting world.

information is produced in the model. As a pleasant side-effect, the actions that the robot has to perform can be accessed directly from the model [10]. To illustrate this, consider the example model above: the action the robot has to perform is denoted by `d2` in the current world `d1`, which involves cleaning three objects, `d5`, `d6` and `d7`, the three rooms, resulting in the possible world `d3`, where all rooms are clean. Because it is a minimal model, no irrelevant actions or objects are part of it.

The theorem prover SPASS was still used to check for inconsistencies and uninformative utterances. Hence, given the discourse representation structure of an utterance, it was translated into first-order logic, and given to the model builder MACE. At the same time, the negation of the input was given to SPASS. When SPASS found a proof, the model builder was halted, and when MACE found a model, the theorem prover was stopped. This parallel architecture was able to give meaningful responses in real-time. The robot in action was regularly demonstrated to visitors at Buccleuch Place as well as to the general public at the National Museum of Scotland in Edinburgh.

Of course, there were limitations in using first-order tools. As with the DORIS system, the size of the inference problem was limited and corresponded to roughly two natural language utterances while still keeping response-times of the robot acceptable to human participants. The problems were also slightly harder than those generated by DORIS, as we were using a first-order language with possible worlds to model the semantics of actions [10]. Also, the number of different objects in the domain was limited (given the background axioms of the robot, MACE produced models in reasonable time for models up to domain size 20). Nevertheless, the overall system was impressive, and showed what one could do with general-purpose, off-the-shelf, first-order inference tools in a practical system. In particular, it demonstrated the idea of using a theorem prover and model builder in parallel, and taking decisions on the basis of the content of the produced model.

4. Recognising textual entailment

The rapid developments in statistical parsing in the 1990s were initially only followed from a distance by researchers in computational semantics. Although some of the parsers developed were impressive with respect to coverage, the level of detail in the syntactic derivations that were produced by these parsers were unsuitable as a basis for constructing semantic representations in a systematic, principled way. Moreover, it would require a large number (probably several thousands) of carefully handcrafted rules specifying the compositional semantics of natural language expressions.

It is not an exaggeration to say that the release of statistical wide-coverage parser for CCG (Combinatorial Categorial Grammar) in 2004 gave rise to a little breakthrough in computational semantics. This CCG parser, developed by Stephen Clark and James Curran [14], and trained on a tree-bank (a collection of annotated syntactic analyses) created by Julia Hockenmaier and Mark Steedman [23], had the best of both worlds: it achieved wide coverage on texts, and produced detailed syntactic derivations. Because of the correspondence between syntax and semantic rules in CCG, this framework was the ideal setting for producing semantic representations of the kind employed in the DORIS system.

Because CCG is a heavily "lexicalised" theory, it has a large number of lexical categories (dictionary entries that map words to categories), and only a handful of grammar rules. In order to translate the output of the parser (a CCG derivation) into a DRT representation (which was my main aim, in order to reuse the existing tools that I developed for DRT and automated inference), I coded a lambda-expression for each of the ca. 400 syntactic categories that were known to the parser. Using the lambda-calculus to produce DRT representations, we succeeded in building a system that translated newspaper texts into semantic representations, with a coverage of around 95% [6,8]. This was an excellent starting point to exploit computerised reasoning for natural language on a wider scale.

Producing semantic representations is one thing, but evaluating the quality, or in other words, the semantic adequacy, is something else. It is hard to say what constitutes a good semantic representation, for instance with respect to the level of abstraction. There is also no corpus with annotated gold-standard semantic representations available, so no simple evaluation schemes exist. Meanwhile, in the same year, 2005, the first challenge to recognising textual entailment (RTE) was organised [15]. This is a shared task for implemented systems to detect whether one (small) text entails another (small) text. The RTE campaign seemed to be just right to test wide-coverage computational semantic formalisms.

To give an impression of the task given in the RTE challenge, consider an example of a positive and an example of a negative entailment pair. Here T stands for the text, and H for the hypothesis, using the terminology of the RTE campaign:

Example: 115 (TRUE)
T: The World Bank has also been criticized for its role in financing projects that have been detrimental to human rights and the natural environment.
H: The World Bank is criticized for its activities.

Example: 117 (FALSE)
T: The release of its report led to calls for a complete ivory trade ban, and at the seventh conference in 1989, the African Elephant was moved to appendix one of the treaty.
H: The ban on ivory trade has been effective in protecting the elephant from extinction.

In the RTE challenge a participating system is given a set of entailment pairs and has to decide, for each T–H pair, whether T entails H or not. The baseline (randomly guessing) already gives a score of 50%, as half of the dataset correspond to true entailments, and the other half to false ones. The best systems on the RTE-1 campaign achieved a score approaching 60%. Arguably, the RTE shared task measures the semantic adequacy of a system: in order for a system to perform well, a system needs to have the ability to make inferences.

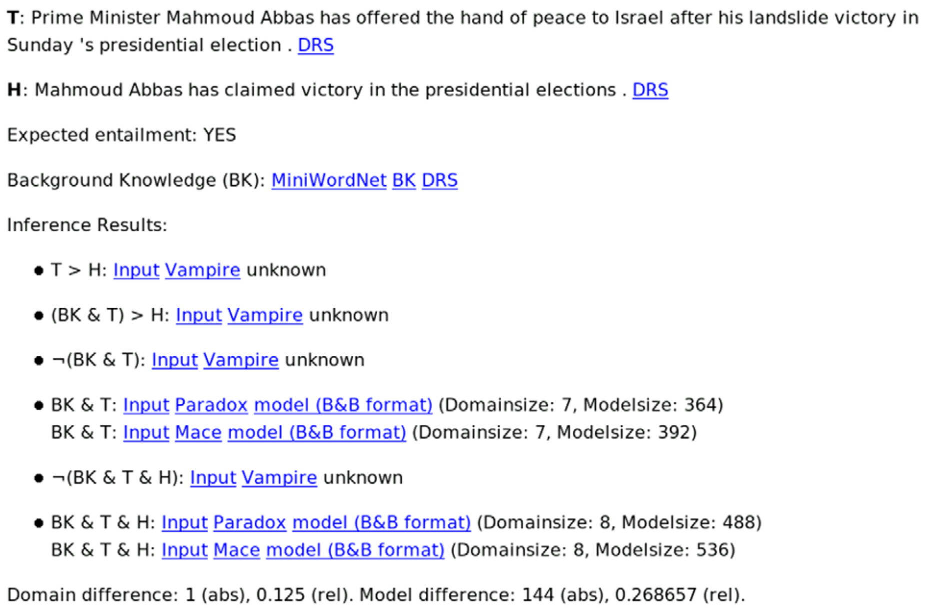
With the CCG parser at our disposal, and the ability to produce semantic representations achieving wide coverage of English texts, we decided to implement a system that used logical inference to approach the RTE challenge. The overall idea, given the available tools, was straightforward: produce a DRT representation for T and H, translate these to first-order logic, and then use an off-the-shelf prover to check whether $T' \rightarrow H'$ (where T' and H' are the translations of T and H, respectively). We installed VAMPIRE [31] for this task, motivated by its good performance at the recent CASCs [33]. As an example, consider the discourse representation structures in (10) and (11) for the T and H of example 115 above:

$$(10)\quad \begin{array}{|c|}\hline \text{x9 x1 x7 x8 x2 x3 x4 x6 x5} \\ \hline \text{named(x9,world\_bank,org)} \\ \text{criticize(x1) patient(x1,x9)} \\ \text{project(x3) nn(x8,x3) financing(x8)} \\ \text{neuter(x7) of(x2,x7) role(x2)} \\ \text{detrimental(x4,x3) to(x4,x5)} \\ \text{human(x6) rights(x6) to(x4,x6)} \\ \text{natural(x5) environment(x5)} \\ \text{in(x2,x3) for(x1,x2) also(x1)} \\ \hline \end{array}$$

$$(11)\quad \begin{array}{|c|}\hline \text{x4 x1 x3 x2} \\ \hline \text{named(x4,world\_bank,org)} \\ \text{criticize(x1) patient(x1,x4) for(x1,x2)} \\ \text{neuter(x3) of(x2,x3) activity(x2)} \\ \hline \end{array}$$

At first, the performance of our RTE system was limited. The system performed quite well on cases such as 115 above, where most of the information needed to make the inference can be derived directly from the information in the texts. However, many of the examples in the RTE data-set require a substantial amount of background knowledge to draw the correct inference. We used WordNet [19] to compute some of these background axioms automatically. WordNet is an electronic dictionary, where content words are connected to each other by linguistic relations. One relation that was useful for our purposes was *hyponymy*, more or less resembling the ISA relation known from knowledge representation approaches in AI. For instance, for the RTE example 115 given earlier, WordNet would tell us that a *role* is a kind of *activity*.

Yet still there were knowledge gaps, and often a little omission of background knowledge would lead to the theorem prover not finding a proof. Ideally, one would like to have an inference engine that would also be able to say that it “almost found a proof” and quantify the amount of missing information, instead of just saying “no proof”. To address this problem, and a way to generate a probability of an entailment between T and H, we used finite model builders.



T: Prime Minister Mahmoud Abbas has offered the hand of peace to Israel after his landslide victory in Sunday 's presidential election . DRS

H: Mahmoud Abbas has claimed victory in the presidential elections . DRS

Expected entailment: YES

Background Knowledge (BK): MiniWordNet BK DRS

Inference Results:

- T > H: Input Vampire unknown
- (BK & T) > H: Input Vampire unknown
- ¬(BK & T): Input Vampire unknown
- BK & T: Input Paradox model (B&B format) (Domainsize: 7, Modelsize: 364)
  BK & T: Input Mace model (B&B format) (Domainsize: 7, Modelsize: 392)
- ¬(BK & T & H): Input Vampire unknown
- BK & T & H: Input Paradox model (B&B format) (Domainsize: 8, Modelsize: 488)
  BK & T & H: Input Mace model (B&B format) (Domainsize: 8, Modelsize: 536)

Domain difference: 1 (abs), 0.125 (rel). Model difference: 144 (abs), 0.268657 (rel).

Fig. 3. Example output screen for recognising textual entailment using the theorem prover VAMPIRE and the model builders MACE and PARADOX.

Model builders, such as MACE [28] produce finite models for satisfiable input (given a certain domain size). Such finite models are the ideal mathematical objects to approximate the "almost found a proof" scenario.

By generating minimal models for T' and for $T' \wedge H'$, we argued that comparing the number of entities of the two models would give us a useful handle on estimating entailment.[7] If the difference is relatively small, it is likely that T entails H—after all, if the difference in sizes is zero, H hasn't introduced any new information. If the difference is relatively large, it is unlikely that T entails H. What counts as relatively small and large has to be determined empirically.

Although this is, obviously, not a sound form of reasoning, in practice it turned out to work well for predicting entailments. We exploited standard machine learning techniques to estimate the thresholds of the model sizes, and used both the size of the domain as well as the number of instantiated relations in the model. We noticed that MACE was quite slow for longer textual entailment examples. We tried PARADOX [13], which is known to be faster, but it does not always return minimal models (with respect to the predicate extensions). To overcome this, we used PARADOX to calculate the domain size, and then called MACE to generate the model given that domain size.

The resulting system scored significantly better than the baseline and ranked among the best performing systems at the RTE-1 challenge [9]. Fig. 3 shows a screenshot of the system.

5. Future directions

It seems that in applying first-order reasoning tools to natural language understanding, we have only scratched the surface of what seems to be an interesting and fruitful application area. A basic proof-of-concept is given, but there is ample space for improvement in performance. Several avenues can be explored that would make the integration of theorem provers and model builders in computational semantics even more successful. I will present them in this section.

The theorem provers and model builders, in the way we used them straight out-of-the-box, were not designed with the application of natural language interpretation in mind. One aspect of natural language understanding is that the linguistic context is continually being updated. In the way we used the inference tools, for each new contribution to the

[7] To deal with negation, one also has to calculate the models for $\neg T'$ and $\neg(T' \wedge H')$ and compare their model sizes. To deal with disjunction, one has to do this for all minimal models for T and H. This hasn't been implemented in the system described here, but this omission has little consequence for the RTE as examples with negation are relative rare, and ones with disjunction even rarer.

dialogue or text, new inference problems for the theorem prover and model builder were computed from scratch. No knowledge of previous inferences was used, and as each new inference problem contains more contextual knowledge, the performance time increases for each new utterance. In practical applications, this forces one to limit the amount of linguistic context taken into consideration. For instance, in the talking robot application, only the last two utterances in the ongoing dialogue are considered, and previous contributions of the dialogue participants are removed from the dialogue history. This is, obviously, not an optimal solution.

One way to overcome this situation is to make theorem provers and model builders *incremental*. Rather than starting from scratch each time, an inference engine might be used in a client-server situation, where the application client provides new information to the inference engine server, which integrates it with its current state, and gives a signal when a proof is found (in the case of a theorem prover), or extends the model (in the case of the model builder). I believe that especially model builders would benefit from such an architecture. It is perhaps also thinkable that model builders extend an existing pre-calculated model, perhaps a model representing a fixed context (going back to the talking robot application, this could consist of the static objects in the environment). Furthermore, in a dialogue or text, not all of the previous linguistic information is relevant. If some of the past conversation topics are dealt with by the dialogue participants, they do not need to be part of future inference problems generated for the inference engines. One way to decide what information is relevant or not is to take the discourse structure into account [36].

Finally, finite model builders such as MACE and PARADOX construct models by iteration over domain size. When the domain size is known, or when the lower or upper bound of the domain is known, this information can be useful for speeding up a model builder. Various ways of dealing with this problem can be thought of. One is a brute-force method of farming out a model generation task over a large set of model builders in a distributed environment, each trying to build a model for the input for a different domain size. Another idea is to estimate the domain size using standard machine learning techniques. A further consideration is to use a sorted first-order language, since typically, a large part of what the background knowledge does is specifying a simple hierarchy of concepts. Model builders that are tailored to sorted languages would be able to cut down the search space drastically and hence be more efficient.[8]

6. Conclusion

First-order inference tools, such as automated theorem provers and model builders, can be successfully used in natural language understanding applications. Obviously, there are limitations, but in many interesting tasks these limitations play a subordinate role. Whether computerised (logical) reasoning will ever become part of mainstream research in natural language processing is questionable, though. We will have to see to what extent statistical approaches (currently dominating computational linguistics) can be applied to natural language interpretation tasks. Meanwhile, collaboration between researchers working in automated deduction and computational linguistics should be stimulated to get a better understanding of the boundaries of applying automated reasoning to natural language understanding and push its potential forward.

References

[1] P. Baumgartner, FDPLL—A first-order Davis–Putnam–Logeman–Loveland procedure, in: D. McAllester (Ed.), CADE-17—The 17th International Conference on Automated Deduction, in: Lecture Notes in Artificial Intelligence, vol. 1831, Springer, Berlin, 2000, pp. 200–219.

[2] F. Black, A deductive question-answering system, PhD thesis, Harvard University, 1964.

[3] P. Blackburn, J. Bos, Representation and Inference for Natural Language. A First Course in Computational Semantics, CSLI, 2005.

[4] P. Blackburn, J. Bos, M. Kohlhase, H. de Nivelle, Inference and computational semantics, in: H. Bunt, R. Muskens, E. Thijsse (Eds.), Computing Meaning, vol. 2, Kluwer, Dordrecht, 2001, pp. 11–28.

[5] J. Bos, Implementing the binding and accommodation theory for anaphora resolution and presupposition projection, Computational Linguistics (2003).

[6] J. Bos, S. Clark, M. Steedman, J.R. Curran, J. Hockenmaier, Wide-coverage semantic representations from a CCG parser, in: Proceedings of the 20th International Conference on Computational Linguistics (COLING '04), Geneva, Switzerland, 2004.

[7] J. Bos, DORIS 2001: Underspecification, resolution and inference for discourse representation structures, in: P. Blackburn, M. Kohlhase, (Eds.), ICoS-3, Inference in Computational Semantics, 2001, pp. 117–124.

[8] In fact, one of the first versions of the MACE model builder supported sorted languages, but this feature seems not to be supported anymore in recent versions.

[8] J. Bos, Towards wide-coverage semantic interpretation, in: Proceedings of Sixth International Workshop on Computational Semantics IWCS-6, 2005, pp. 42–53.

[9] J. Bos, K. Markert, Recognising textual entailment with logical inference techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2005), 2005.

[10] J. Bos, T. Oka, An inference-based approach to dialogue system design, in: S.-C. Tseng (Ed.), COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan, 2002, pp. 113–119.

[11] E. Charniak, Towards a model of children's story comprehension, PhD thesis, Massachusetts Institute of Technology, 1972.

[12] E. Charniak, D. McDermott, Introduction to Artificial Intelligence, Addison-Wesley, 1985.

[13] K. Claessen, N. Sörensson, New techniques that improve mace-style model finding, in: Model Computation—Principles, Algorithms, Applications (Cade-19 Workshop), Miami, Florida, USA, 2003.

[14] S. Clark, J.R. Curran, Parsing the WSJ using CCG and log–linear models, in: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04), Barcelona, Spain, 2004.

[15] I. Dagan, O. Glickman, B. Magnini, The pascal recognising textual entailment challenge, in: Lecture Notes in Computer Science, vol. 3944, Springer, Berlin, 2006, pp. 171–190.

[16] K. Dahlgren, J. McDowell, E.P. Stabler, Knowledge representation for commonsense reasoning with text, Computational Linguistics 15 (3) (1989) 149–170.

[17] H. De Nivelle, A resolution decision procedure for the guarded fragment, in: Automated Deduction—CADE-15, 15th International Conference on Automated Deduction, Springer, Berlin, 1998, pp. 191–204.

[18] D.R. Dowty, R.E. Wall, S. Peters, Introduction to Montague Semantics, in: Studies in Linguistics and Philosophy, D. Reidel Publishing Company, 1981.

[19] C. Fellbaum (Ed.), WordNet. An Electronic Lexical Database, The MIT Press, 1998.

[20] A. Franke, M. Kohlhase, System description: Mathweb, an agent-based communication layer for distributed automated theorem proving, in: CADE-16, 16th International Conference on Automated Deduction, 1999.

[21] C.C. Green, B. Raphael, The use of theorem-proving techniques in question-answering systems, in: Proceedings of the 1968 23rd ACM National Conference, ACM Press, New York, 1968, pp. 169–181.

[22] J.R. Hobbs, M.E. Stickel, D. Appelt, P. Martin, Interpretation as abduction, Technical Report 499, AI Center, SRI International, Menlo Park, California, 1990.

[23] J. Hockenmaier, M. Steedman, Generative models for statistical parsing with combinatory categorial grammar, in: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, 2002.

[24] D. Jurafsky, J.H. Martin, Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, 2000.

[25] H. Kamp, U. Reyle, From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT, Kluwer, Dordrecht, 1993.

[26] H. Kamp, A theory of truth and semantic representation, in: J. Groenendijk, T.M.V. Janssen, M. Stokhof (Eds.), Formal Methods in the Study of Language, Mathematical Centre, Amsterdam, 1981, pp. 277–322.

[27] K. Konrad, D.A. Wolfram, System description: Kimba, a model generator for many-valued first-order logics, in: 16th International Conference on Automated Deduction CADE-16, 1999.

[28] W. McCune, Automatic proofs and counterexamples for some ortholattice identities, Information Processing Letters 65 (6) (1998) 285–291.

[29] W. McCune, R. Padmanabhan, Automated Deduction in Equational Logic and Cubic Curves, Lecture Notes in Computer Science (AI subseries), Springer, Berlin, 1996.

[30] S. Pulman, Formal and computational semantics: A case study, in: Proceedings of Seventh International Workshop on Computational Semantics IWCS-7, 2007.

[31] A. Riazanov, A. Voronkov, The design and implementation of Vampire, AI Communications 15 (2–3) (2002).

[32] J. Robinson, A machine oriented logic based on the resolution principle, Journal of the ACM 12 (1965) 23–61.

[33] G. Sutcliffe, C. Suttner, The state of CASC, AI Communications 19 (1) (2006) 35–48.

[34] G. Sutcliffe, C. Suttner, The tptp problem library, J. Autom. Reason. (ISSN 0168-7433) 21 (2) (1998) 177–203.

[35] R.A. Van der Sandt, Presupposition projection as anaphora resolution, Journal of Semantics 9 (1992) 333–377.

[36] B.L. Webber, Structure and ostension in the interpretation of discourse deixis, Language and Cognitive Processes 6 (2) (1991) 107–135.

[37] C. Weidenbach, B. Afshordel, U. Brahm, C. Cohrs, T. Engel, E. Keen, C. Theobalt, D. Topic, System description: Spass version 1.0.0, in: H. Ganzinger (Ed.), 16th International Conference on Automated Deduction, CADE-16, in: Lecture Notes in Artificial Intelligence, vol. 1632, Springer-Verlag, Berlin, 1999, pp. 314–318.

[38] T. Winograd, Procedures as a representation for data in a computer program for understanding natural language, PhD thesis, Massachusetts Institute of Technology, 1971.