Recognizing Textual Entailment and Computational Semantics

Johan Bos

Abstract Recognizing textual entailment (RTE)—deciding whether one piece of text contains new information with respect to another piece of text—remains a big challenge in natural language processing. One attempt to deal with this problem is combining deep semantic analysis and logical inference, as is done in the Nutcracker RTE system. In doing so, various obstacles will be met on the way: robust semantic analysis, designing interfaces to state-of-the-art theorem provers, and acquiring relevant background knowledge. The coverage of the parser and semantic analysis component is high, yet performance on RTE examples yields high precision but low recall. An empirical study of Nutcracker's output reveals that the true positives are caused by sophisticated linguistic analysis such as coordination, active-passive alternation, pronoun resolution and relative clauses; the small set of false positives are caused by insufficient syntactic and semantic analyses. But most importantly, the false negatives are produced mainly by lack of background knowledge that is only implicit in the RTE examples.

1 Introduction

Textual entailment has long been used as an illustrational device in formal semantics to show or convince scholars that certain natural language inferences hold or don't (as in popular textbooks such as Gamut 1991; Heim and Kratzer 1998, and Chierchia and McConnell-Ginet 1991). This has merely been a theoretical exercise, until the introduction of recognizing textual entailment (RTE) as a shared task in the area of natural language processing Dagan et al. (2006) in 2005, even though the idea of the computational variant was aired much earlier (Cooper et al. 1996; Monz and de Rijke 2001). The RTE challenge consists of predicting whether one (short) text entails another (short) text. The RTE data-sets are a collection of such text–hypothesis pairs, labelled with a gold standard tag. Here are two such examples, one labelled as

J. Bos (✉)
Center for Language and Cognition (CLCG), University of Groningen, Groningen, The Netherlands
e-mail: johan.bos@rug.nl

H. Bunt et al. (eds.), *Computing Meaning*, Text, Speech and Language Technology 47,
DOI 10.1007/978-94-007-7284-7_6,

FALSE (no entailment, i.e. hypothesis H contains new information with respect to text T), and one labelled as TRUE (entailment, i.e. no new information in H given T):

Example 1: FALSE
T: I recently took a round trip from Abuja to Yola, the capital of Adamawa State and back to Abuja, with a fourteen-seater bus.
H: Abuja is located in Adamawa State.

Example 2: TRUE
T: Bountiful arrived after war's end, sailing into San Francisco Bay 21 August 1945. Bountiful was then assigned as hospital ship at Yokosuka, Japan, departing San Francisco 1 November 1945.
H: Bountiful reached San Francisco in August 1945.

It soon became clear that RTE is an extremely difficult task: simple baseline systems based on textual surface features are hard to outperform by more sophisticated systems. Not only does one need a robust and accurate analysis of text, also the use of external resources to inform the inference process are essential.

Various approaches to RTE have been proposed, ranging from surface-oriented techniques to methods using sophisticated semantic analysis. This focus of this chapter is on a method belonging in the latter category, namely determining textual inferences on the basis of logical inference. The idea is simple and rooted in the formal approaches to natural language semantics mentioned before: we translate the texts into logical formulas, and then use (classical) logical inference to find out whether T entails H, whether T and H are consistent or contradictory, and so on.

Even though the idea itself is simple in theory, its practical execution isn't. In this chapter I describe a framework and implementation for textual inference based on first-order logic and formal theory. It comprises a system for RTE, Nutcracker, developed by myself over the years since the start of the RTE challenge (Bos and Markert 2005) and has been briefly described by others in a wider context (Balduccini et al. 2008), but never been the subject of publication itself. The aim is to find an answer to the question whether there is a significant role for computational semantics to play in the current state of RTE. From this "big" question several smaller questions arise, that are probably easier to answer, and I will concentrate on these first:

1. Can we use deep semantic analysis and logical inference, or are we lacking coverage?
2. Are the RTE data-sets suitable for black-box testing of systems claiming to performing natural language understanding?
3. And finally, given the knowledge that RTE is a hard (and yet unsolved) problem: can we identify a bottleneck—is it in semantic analysis, selecting background knowledge, or in theorem proving?

This chapter is organized as follows. First the framework and implementation of the logical approach to RTE, is presented in Sect. 2. This includes syntactic and semantic analysis, with a description of the parser (based on categorial grammar) and

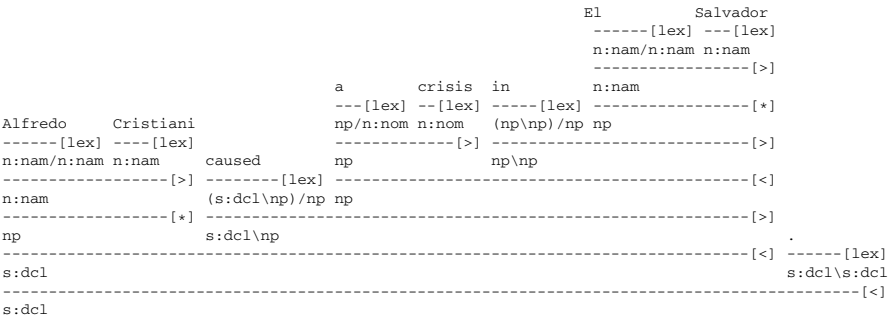

Fig. 1 Output of the C&C parser, a CCG derivation, as displayed by Boxer

the semantic interpretation component (Boxer, implementing a version of Discourse Representation Theory), and the use of off-the-shelf theorem provers to perform inferences on the textual analyses of RTE examples. In Sect. 3 I look critically at the performance of the logical approach to RTE, and show where it acts well and on what examples it fails to deliver the goods.

2 The Logical Method

2.1 Robust Semantic Analysis

With "semantic analysis" I mean the process of mapping text into logical formula. Traditionally, this is performed by a syntactic analysis (with the help of a parser) followed by a semantic analysis that produces a logical form based on the output of the syntactic parser. For the purposes of RTE based on logical inference, the linguistic analysis needs to be reasonably sophisticated and at the same time offer large coverage. It needs to be sophisticated in analysis because a shallow analysis would not support the logical inferences that need to be drawn and hence sacrifice precision in performance. It needs to be robust and offer large coverage to achieve a high recall in performance. As a practical rule of thumb, the loss in coverage should still outweigh the gain in performance using deep linguistic analysis.

Nowadays there are several (statistical) parsers available that offer broad coverage syntactic analysis on news-wire texts. The parser of our choice, the C&C parser (Clark and Curran 2004), combines speed and robustness with detailed syntactic analyses in the form of derivations of combinatory categorial grammar (Fig. 1).

Categorial grammar offers a principled way to construct formal meaning representations with the help of the λ-calculus. In a nutshell, it works as follows. Each basic syntactic category is associated with a basic semantic type, and using the recursive definition of categories and types, this also fixes the semantic types of complex syntactic categories. This results in a strongly lexically-driven approach, where only the semantic representations have to be provided for the lexical categories. Function application will take care of the rest and produce meaning representations for phrases beyond the token level and eventually for the entire sentence.

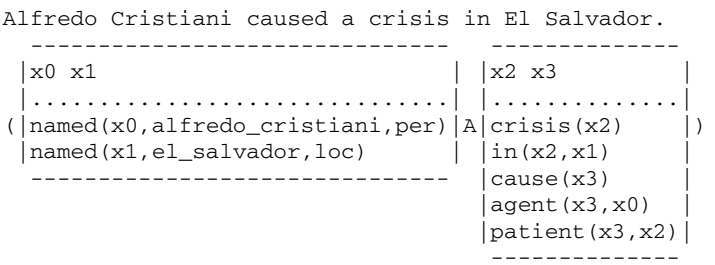

Fig. 2 Boxer output for a simple text, a DRS (Discourse Representation Structure)

As for choice of meaning representation language, it needs to be something that supports logical inference as well as adequately describe natural language meaning. There is an uneasy and unsolved tension here between expressiveness on the one hand and efficiency on the other. The formalisms proposed by linguists and philosophers are usually not computationally attractive—most of them exceed the expressive power of first-order logic, and theorem proving for first-order logic is already undecidable (more precisely, first-order logic is known to be *semi-decidable* (Blackburn and Bos 2005)). Yet, there are powerful theorem provers for first-order logic available developed by the automated deduction research community, and it seems a good compromising choice as language to perform logical inference given the current state-of-the-art.

However, we won't use standard first-order formula syntax, but adopt a variant of Discourse Representation Theory's DRSs, Discourse Representation Structures, graphically visualized as boxes (Fig. 2). DRT (Kamp and Reyle 1993) offers a way to deal with many linguistic phenomena in a principled way, including quantifiers, pronouns, presupposition and events. Diverging slightly from standard DRT, I adopt a neo-Davidsonian way for describing events (rather than the Davidsonian approach employed in classical DRT), because this results in a lower number of background knowledge rules (meaning postulates) required to draw correct inferences. Turning to implementation, the meaning representations are produced by the semantic parser Boxer (Bos 2008), which works on the output of the aforementioned C&C parser.

Boxer performs pronoun resolution, presupposition projection, thematic role labelling and assigns scope to quantifiers, negation and modal operators. It produces one semantic representations for each input, and its logical form is fully disambiguated. Note that semantic underspecification, a technique to pack several meanings into one compact representation, isn't a feasible option here, as it remains unclear how theorem provers would work with underspecified representations.

2.2 Applying Theorem Proving

In the previous section I showed how to produce a DRS for a text and hypothesis of a pair of the RTE data-set. The next step involves translating these DRSs into formulas

of first-order logic, and pass on the result in a suitable way to a theorem prover. If the theorem prover then succeeds in finding a proof, we predict an entailment for this RTE pair. However, the standard translation from DRS to FOL (Muskens 1996; Kamp and Reyle 1993) gives wrong predictions to RTE problems because it doesn't take modalities and embedded propositions into account. The standard translation, for instance, would predict an entailment for the following example pair:

Example 3: FALSE
T: Leakey believed Kenya's wildlife, which underpins a tourist industry worth Dollars 450m a year, could be managed in a profitable and sustainable manner.
H: Kenya's wildlife is managed in a profitable manner.

Why is this? In the standard translation, it is impossible to connect the embedded proposition to a belief report (or other propositional attitude) or modal operator, because first-order terms can't be formulas. The modal translation, that I adopt, is based on a technique called reification. It translates a basic DRS condition with n terms into a first-order formula with $n+1$ arguments, where the added term is a first-order variable ranging over entities. (I won't give the full translation from DRSs to modal FOL here for reasons of space, but instead refer the interested reader to Bos 2004.) One might want to refer to these entities as "possible worlds", "situations", or simply "propositions". Whatever you call them, this way it is possible to connect embedded propositions to attitudinal nouns and verbs or modal operators, and therefore prevents unwanted entailments such as in the example above. In cases with factive constructions (as in the sentence "Bill knows that Mary smokes" or "the fact that Mary smokes"), meaning postulates could specify the project content of the embedded clause to be interpreted as if it were in the main clause.

Theorem proving doesn't just play the role for checking entailment between T and H. We also need to check whether T and H are logically consistent. This is necessary because otherwise we might predict incorrect entailments. If T is inconsistent, anything would follow from that. Logically speaking that would be sound, but for natural language entailment this is (perhaps) an unwanted result. If H is inconsistent, then checking whether T entails H would boil down to checking whether T is consistent. Again, this is something we should be able to detect. And finally, if T and H taken together are inconsistent, then clearly T does not entail H (in fact, H is very informative in such a case!).

This brings us to the basic algorithm for applying first-order theorem proving to an RTE example with text T and hypothesis H. For convenience, we write X′ to designate the FOL translation of natural language text X, derived from a DRS produced for X. The Boolean function *proof* has as input a formula, and returns true if it finds a proof (given certain time and space constraints), false otherwise. Figure 3 shows all the steps of the algorithm.

Note that in Fig. 3 we check for the consistency of a formula ϕ by trying to prove its negation—if we manage to do so, $\neg\phi$ is a theorem, and therefore ϕ has no model, in other words is inconsistent. Steps 1 and 2 apply to cases where—for whatever reason—the text or hypothesis is inconsistent itself. There isn't much point

```
IF proof(not(T')) THEN                              % STEP 1
   OUTPUT "unknown"
ELSE
   IF proof(not(H')) THEN                           % STEP 2
      OUTPUT "unknown"
   ELSE
      IF proof(not(and(T,H))) THEN                  % STEP 3
         OUTPUT "informative"
      ELSE
         IF proof(not(and(T,not(H))) THEN           % STEP 4
            OUTPUT "entailment"
         ELSE
            OUTPUT "informative"                    % STEP 5
         ENDIF
      ENDIF
   ENDIF
ENDIF
```

Fig. 3 Core of the Nutcracker algorithm for recognizing textual entailment

to continue at this stage. Step 3 checks for a contradiction between T and H. If this is the case, there is no entailment. Step 4 is the moment of truth: the check whether T entails H, and applies only when both T and H are consistent. If a proof is found, then entailment is reported for this pair. Step 5, finally, is a fallback clause in case no proof was found in previous steps of the algorithm.

Any theorem prover for first-order logic could be used in theory. In practice, there is quite a lot of choice, thanks to the active area of automated deduction that offers various efficient state-of-the-art provers for research purposes, and a lot of variation in performance, too. The theorem prover used in our experiments reported later in this chapter is Vampire (Riazanov and Voronkov 2002), the currently highest ranked prover in CASC, the annual competition for inference engines (Sutcliffe and Suttner 2006). In addition to a theorem prover, we use the model builder Paradox to find counter models (Claessen and Sörensson 2003). Following Blackburn and Bos (2005), for each inference problem called in Fig. 3 the theorem prover and model builder work in parallel, where the model builder gets the negated input of the theorem prover. If a proof is found for problem $\neg\phi$, the model builder is halted because it would never be able to find a model for ϕ—if a model is found for ϕ, the theorem prover is halted because it would never be able to find a proof for $\neg\phi$.

The model builder searches for models up to a specified domain size n, and terminates if it can't construct a model for sizes $1 - n$. In theory, because first-order logic is semi-decidable, this setting always terminates with one of three results: (i) proof found, (ii) no proof found but finite countermodel constructed with domain size n, or (iii) no proof and no model for size n (for instance for inputs that have non-finite models). Case (i) succeeds if we give enough resources (time and space) to the theorem prover, but in practice we use a time-out. For case (ii) by specifying the maximum domain size as high as possible while maintaining reasonable response times. Case (iii) is one that we wish to avoid in practice.

Table 1 Coverage of the C&C parser and Boxer on RTE examples

Data-set	Pairs	Semantics	Coverage
RTE–2 dev	800	784	98.0 %
RTE–2 test	800	782	97.8 %
RTE–3 dev	800	780	97.5 %
RTE–3 test	800	786	98.3 %
Total	3,200	3,132	97.9 %

Table 2 Proofs found on RTE-2 and RTE-3 (3,132 pairs)

Data-set	Proofs	Precision	Recall
RTE-2 Dev	13	100 %	3.3 %
RTE-2 Test	14	86 %	3.0 %
RTE-3 Dev	14	93 %	3.3 %
RTE-3 Test	13	77 %	2.8 %

2.3 Implementation and Results

The approach to RTE as described above is implemented as the Nutcracker RTE system.[1] Nutcracker is basically a wrapper around a pipeline of NLP components, comprising a tokeniser, POS tagger, lemmatiser (Minnen et al. 2001) and named entity recognizer, followed by the C&C parser and Boxer. Nutcracker further coordinates the communication with the external theorem provers and model builders as designed in Fig. 3.

Coming back to one of the key questions posed at the beginning of this chapter, what is the coverage and quality of our NLP pipeline on RTE examples, and is it good enough? The coverage of the pipeline on RTE examples is shown in Table 1, from which we can conclude that the coverage for producing semantic representations is high (around 98 %), and therefore suitable for a task such as RTE, assuming we can recover from the loss of 2 % in recall by achieving a high precision. However, even though producing semantic representations in a robust way is a good start for performing well on the RTE task, it is merely a single step in the NLP pipeline. The ultimate success depends on the number and accuracy of the proofs that are found. As Table 2 shows in terms of precision and recall, the accuracy of proofs is high, but the number of proofs is very low.

As it stands, using simply logical inference would just about outperform the simplest baseline (flipping a coin, assuming an equal distribution between the TRUE and FALSE entailment pairs in the data-set, which is usually the case in RTE exercises). As a matter of fact, the Nutcracker system employs a slightly more sophisticated baseline system based on word overlap in the cases where it fails to find a

[1] The source code of the system can be downloaded via the website of the C&C tools Curran et al. (2007).

proof, a baseline that performs remarkably well. In the next section we try to find out why precision is not 100 %, explain why recall is low, and make suggestions for how to improve on this.

3 A Critical Evaluation of Performance

RTE is measured in terms of recall (how many instances of the total given to a system are correctly predicted) and precision (how many instances attempted by a system are correctly predicted). RTE systems based on logical inference tend to be low in recall and high in precision. Why is this so? In this section we would like to find an answer by inspecting the output of Nutcracker on the RTE-2 and RTE-3 data-sets.

The logical approach to RTE assumes there is no entailment for an T–H pair unless a proof is found. However, not every proof corresponds to an entailment in the RTE data-set, and not every entailment in the RTE data-set triggers a proof. Hence, we can evaluate and verify the performance of the system by dividing the data into four classes:

1. true positives (proofs found for an entailment);
2. false positives (proofs found for a non-entailment);
3. true negatives (no proof found for a non-entailment);
4. false negatives (no proof found for an entailment).

A moment of reflection informs us that it is not very interesting to discuss the true negatives, because, in a way, this can be viewed as a default behaviour of the system. It is however interesting to have a closer look at the remaining three classes of system output, and we will do so here.

3.1 Proofs Found for Entailment Pairs (True Positives)

In this class we can distinguish several semantic phenomena whose analyses in DRT correctly predict entailments. I will group them into various categories: conjunction elimination, coordination, active-passive altnernation, pronoun resolution, relative clauses, appositives, and control constructions.

3.1.1 Conjunction Elimination

The largest set of true positives is caught by conjunction elimination, a basic inference rule that says that from a conjunctive statement $\phi \wedge \psi$ one can infer ϕ and ψ. We encountered thirteen cases that were correctly classified by Nutcracker as entailment due to conjunction elimination in the RTE data. Some examples are shown

below, where the relevant phrases in T and H are set in bold face. (In these and following examples, some are abbreviated versions of the original entry in the RTE data-set to save space. The gold-standard judgments (TRUE/FALSE) are taken from the RTE data-set.)

Example 4: TRUE
T: The Gurkhas come **from mountainous Nepal** and are extremely tenacious...
H: The Gurkhas come **from Nepal**.

Example 5: TRUE
T: At least eight people have been killed in a **suicide bomb attack** on Sri Lanka's...
H: People were killed in **suicide attacks**.

Example 6: TRUE
T: A male rabbit is called a buck **and** a female rabbit is called a doe, just like deer.
H: A female rabbit is called a doe.

Example 7: TRUE
T: Tom Cruise is **married to actress Nicole Kidman** and the couple has...
H: Tom Cruise is **married to Nicole Kidman**.

Example 8: TRUE
T: Spirou was **created in 1938 by Rob-Vel**, who sold the rights to...
H: Spirou was **created by Rob-Vel**.

As these examples illustrate, several syntactic constructions fall into this category of valid inferences: intersective adjectives ("X is from mountainous Y" entails "X is from Y"), noun-noun compounds (a suicide bomb attack is also a bomb attack), appositives ("X is the actress Nicole Kidman" entails that "X is Nicole Kidman"), and clauses ("X and Y" entails Y). The last example shows that the order of event modifiers is not sensitive to entailment ("created in X by Y" entails "created by'Y"). Nutcracker makes corrected predictions for this type of examples, although, as we will see below, in some cases conjunction elimination doesn't always yield the desired result.

3.1.2 Verb Phrase Coordination

This category of examples shows that a correct syntactic and semantic analysis for verb phrase coordination can contribute to finding an entailment. Approaches based on surface features will likely run into problems for this class of examples. Consider the following examples that Nutcracker handled correctly:

Example 9: TRUE
T: Pibul was anti-communist as well as nationalistic.
H: Pibul was nationalistic.

Example 10: TRUE
T: Bush withheld judgment Monday on… Iraq, and said angry protests in Indonesia…
H: Bush said that protests in Indonesia…

Both examples are cases of verb phrase coordination ("X was P as well as Q" entails "X was Q", and "X did P and did Q" entails "X did Q"). These are non-trivial RTE examples because they require a sophisticated linguistic analysis; shallow RTE approach based on surface forms would have a hard time predicting these inferences. We found three cases of VP coordination entailment in the RTE data-set.

3.1.3 Active-Passive Alternation

In Boxer, verb phrases in passive form are semantically represented as their active paraphrase. Put differently, the Boxer system will produce the same semantic representation for the active sentence "Batman chased the Joker" and "The Joker was chased by Batman." This enables Nutcracker to make inferences of the following kind:

Example 11: TRUE
T: Initially the Bundesbank opposed the introduction of the euro but was…
H: The introduction of the euro has been opposed.

Example 12: TRUE
T: In India, carpets are made mostly in Uttar Pradesh, which adopted a "Child Labour Abolition and Regulation Act" in 1986.
H: The Child Labour Abolition and Regulation Act was adopted in 1986.

The Nutcracker system is able to deal with this because the C&C parser is able to detect verb phrases in passive mood, for which Boxer select the correct thematic roles. For instance, the active-passive alternation example above translates as "X opposed Y" entails "Y has been opposed". There were four cases of active-passive alternation in the studied RTE data-sets.

3.1.4 Past and Present Participles

Past and present participles that modify nouns are analysed in Boxer like events introduced by ordinary verb phrases. This ensures that Nutcracker predicts entailments as in the following examples:

Example 13: TRUE
T: Another factor in the **rising cost of paper** is the increased cost of wood pulp, from which paper is made.
H: The cost of paper is rising.

Example 14: TRUE
T: The provincial veterinarian with the Department of Forest Resources and Agri-foods, Dr. Hugh Whitney, confirmed today another case of rabies in Labrador, bringing the total number of **confirmed rabies cases** to nine in Labrador since November 2000.
H: A case of rabies was confirmed.

The two examples above were the only ones that I found in the RTE data. In the first example the present participle "rising" is analysed by Boxer as an event with the thematic roles for the corresponding intransitive verb. In the second example, the past participle "confirmed" is treated by Boxer as a passive verb.

3.1.5 Relative Clauses and Control Constructions

Relative clauses and control constructions invoke interesting semantic dependencies. Nutcracker is able to correctly predict entailments for the following examples, covering standard relative clauses, a reduced relative clause, and a control construction.

Example 15: TRUE
T: **Franz Liszt**, a Hungarian composer **who lived from 1811 to 1886** was the equivalent of a rock star in his day. His piano compositions were extremely popular and he often gave concerts to his multitude of fans. Liszt was also the pioneer of many musical techniques, including the symphonic poem and the technique of transforming themes.
H: Franz Liszt lived from 1811 to 1886.

Example 16: TRUE
T: The prize is named after **Alfred Nobel**, a pacifist and entrepreneur **who invented dynamite in 1866**. Nobel left much of his wealth to establish the award, which has honoured achievements in physics, chemistry, medicine, literature and efforts to promote peace since 1901.
H: Alfred Nobel invented dynamite in 1866.

Example 17: TRUE
T: **The Pharos**, a monumental **lighthouse built around 280 BC** and standing 330 ft high, lit the entrance to Alexandria harbour for centuries, but archeologists have never been able to identify positively any remains.
H: The Pharos Lighthouse was built around 280 BC.

Example 18: TRUE
T: **The 84-year-old pope** was wheeled to a hospital window, and blessed the crowd **by making the sign of the cross** in clear gestures, as a Vatican photographer snapped pictures.
H: The pope made the sign of the cross.

Note that the first three examples also interact with a proper analysis of appositives. Example 18 is correctly predicted thanks to the lexical semantics of "by", a VP modifiers sub-categorizing for a present participle, ensuring that the subject of the participle is the same as the subject of the VP it is modifying.

3.1.6 Pronouns

The current version of Boxer performs pronoun resolution using a simple rule-based algorithm that emphasizes precision at the cost of recall. This mean that not all pronouns are resolved, but when they are, they are usually associated with a correct antecedent. Consider the following examples that Boxer got correct and caused a correct entailment when running Nutcracker:

Example 19: TRUE
T: **Aeschylus** was born in 525 BC, and spent his youth as a soldier in the Athenian army. **He** wrote The Persians when he was 53 years old, but it is his earliest surviving work.
H: "The Persians" was written by Aeschylus.

Example 20: TRUE
T: **Yunus**, who shared the 1.4 million prize Friday with the Grameen Bank that **he** founded 30 years ago, pioneered the concept of "microcredit"...
H: Yunus founded the Grameen Bank 30 years ago.

Both examples demonstrate the need of pronoun resolution for RTE. We note in passing that the first example also shows active-passive alternation, and that the second example requires a proper treatment of object relative clauses, underlining that in real-world RTE there is often more than one complex phenomenon that one needs to get right to correctly predict entailments.

3.2 Incorrect Proofs Found (False Positives)

Assuming that the theorem prover that one uses is sound, any proof that is produced by it is mathematically correct. But in the RTE setting that we are examining, finding a proof doesn't automatically mean predicting a correct entailment. These cases,

the *false positives*, are usually caused by a wrong semantic analysis of Boxer. Fortunately, this doesn't happen often. But when it does happen, it is interesting to find out why, because it informs you where semantic analysis (sometimes unexpectedly) failed and points out weak points in the semantic analysis. We will look at these cases in this section.

3.2.1 Incorrect Syntactic Analyses

An incorrect syntactic analysis automatically yields an incorrect semantic analysis. Sometimes this leads to incorrect entailment predictions. For the examples below Nutcracker predicted entailments, although they were tagged as non-entailments in the gold standard annotation:

Example 21: FALSE
T: Yunus, who was nominated for the peace prize at least twice before, is the first person from Bangladesh, a country of 147 million, to win a Nobel Prize.
H: Yunus is the first person to win a Nobel Prize.

Example 22: FALSE
T: Germany will pay more into the EU coffers than Britain had originally proposed—but still less than it had been prepared to pay at the last summit six months ago.
H: Germany will pay more into the EU coffers than Britain.

In Example 21, the set of alternatives for the superlative expression "first" comprises *persons from Bangladesh* (not just persons), blocking the entailment. Computing the alternative set of superlatives was recognized as an important problem in Bos and Nissim (2006), and the example above supports this once more. In Example 22 a wrong syntactic analysis of the comparative caused a false positive. The best remedy to deal with these kinds of problems is train the parser on more data or on revised gold-standard data, as in Honnibal et al. (2010).

3.2.2 Incorrect Semantic Analysis

This is a motley crew of examples, including unjustified conjunction elimination (Example 23), not covering certain downward entailing quantifiers (Example 24), wrongly resolved pronouns (Example 25), and not taking care of intensional adjectives (Example 26):

Example 23: FALSE
T: Boys and girls will be segregated during sex education in junior high school.
H: Boys and girls will be segregated in junior high school.

Example 24: FALSE
T: There are approximately 3.7 million European Citizens with intellectual disability.
H: There are approximately 3.7 million European Citizens.

Example 25: FALSE
T: The rhinestone-studded Nudie suit was invented by Nudie Cohn in the 1940s, an Americanization of the matador's "suit of lights".
H: The matador's "suit of lights" was invented by Nudie Cohn.

Example 26: FALSE
T: Belknap was impeached by a unanimous vote of the House of Representatives for allegedly having received money in return for post tradership appointments.
H: Belknap received money in return for post tradership appointments.

For some of these phenomena there are relatively easy fixes thinkable: equip Nutcracker with a better anaphora resolution component, and extend Boxer with a proper analysis of intensional adjectives. But such fixes probably won't have a high impact on the overall results of Nutcracker on the RTE data-sets, because they represent a long tail of various rare cases.

3.3 Missing Proofs (False Negatives)

A large class of predictions is formed by the false negatives: no proof was found by Nutcracker, but it should have. A great setting for the blame game to commence. Can you blame the parser? Boxer? The theorem prover? Where is the bottleneck?

As far as the RTE data-sets is concerned, none of the traditional pipeline components in Nutcracker is responsible for the majority of errors. To illustrate this point, I randomly picked a sequence of examples that had "missing proofs" (i.e. RTE examples that were labelled as TRUE but for which Nutcracker predicted no entailment) and examined them closely:

Example 27: TRUE
T: The Pentagon is rejecting demands by Kyrgyzstan to pay for the past use of Manas air base, a key military facility for US aircraft flying missions to Afghanistan.
H: Manas air base is located in Kyrgyzstan.

Example 28: TRUE
T: He also referred to the "illegal" arrest on 31 May of Mexican Professor Maria Eugenia Ochoa Garcia, whom the Salvadoran government accused of having connections with the Salvadoran guerrillas.
H: Maria Eugenia Ochoa Garcia was arrested in May.

Example 29: TRUE
T: Mercedes-Benz USA (MBUSA), headquartered in Montvale, New Jersey, is responsible for the sales, marketing and service of all Mercedes-Benz and Maybach products in the United States.
H: MBUSA is based in New Yersey.

Example 30: TRUE
T: Since joining the Key to the Cure campaign three years ago, Mercedes-Benz has donated over million toward finding new detection methods, treatments and cures for women's cancers.
H: Mercedez-Benz supports the Key to the Cure campaign.

Example 31: TRUE
T: ASCAP is a membership association of more than 200,000 U.S. composers, songwriters, lyricists and music publishers of every kind of music.
H: More than 200,000 U.S. composers, songwriters, lyricists and music publishers are members of ASCAP.

The inference in Example 27 can only be made with the knowledge that if a government of X demands someone to pay for the use of a facility Y, then Y is located in X. Similarly, the inference in Example 28 can only be made with the knowledge that if an arrest on time T of a person X takes place, then X was arrested on T. Likewise, the inference in Example 29 can only be made with the knowledge that if X is headquarted in Y, then X is based in Y. The inference in Example 30 can only be made with the knowledge that if X joins Y, then X supports Y. And finally, the inference in Example 31 can only be made with the knowledge that if X is a membership association of Y, then Y are members of X.

These aren't special cases, but are representative for a problem for logical approaches to RTE, and perhaps even for purely statistical approaches. These examples make clear that for the majority of the cases there is implicit knowledge required to make the requested inference. I don't think that this is knowledge that should be supplied by a dedicated component, rather than by the semantic analyzer (Boxer, in the case of the Nutcracker RTE system).

I am not aware of any resource that makes available background knowledge rules of the kind required by the class of false negatives represented above. Current lexical resources, such as WordNet and VerbNet, certainly do not offer such detailed information. Unsupervised knowledge mining approaches could be a future, partial answer to this problem; the current state-of-the-art in this area (Lin and Pantel 2001) shows interesting high-recall results, albeit with relatively low precision.

4 Discussion and Conclusion

Coverage for producing semantic representations is high (around 98 %), and therefore suitable for a task such as RTE. The number of proofs found is small. But when

a proof is found, it is usually correct in playing a role for predicting entailment. The rare, incorrect proofs are due to insufficient syntactic and semantic analysis and usually of complex linguistic nature. However, the bottleneck for the logical approach to RTE isn't the current state of automated semantic analysis or theorem proving, but the lack of supporting background knowledge. The question is whether resources such as WordNet and VerbNet could play a role in filling this gap, or whether more elaborated knowledge bases are needed. This is an important question for future research on RTE and computational semantics.

References

Balduccini, M., Baral, C., & Lierler, Y. (2008). Knowledge representation and question answering. In V. Lifschitz, F. van Harmelen, & B. Porter (Eds.), *Handbook of knowledge representation* (pp. 779–819). Amsterdam: Elsevier.

Blackburn, P., & Bos, J. (2005). *Representation and inference for natural language. A first course in computational semantics*. Stanford: CSLI.

Bos, J. (2004). Computational semantics in discourse: Underspecification, resolution, and inference. *Journal of Logic, Language and Information*, *13*(2), 139–157.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In J. Bos & R. Delmonte (Eds.), *Research in computational semantics: Vol. 1. Semantics in text processing. STEP 2008 conference proceedings* (pp. 277–286). London: College Publications.

Bos, J., & Markert, K. (2005). Recognising textual entailment with logical inference. In *Proceedings of the 2005 conference on empirical methods in natural language processing* (pp. 628–635).

Bos, J., & Nissim, M. (2006). An empirical approach to the interpretation of superlatives. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, Sydney, Australia (pp. 9–17).

Chierchia, G., & McConnell-Ginet, S. (1991). *Meaning and grammar. An introduction to semantics*. Cambridge: MIT Press.

Claessen, K., & Sörensson, N. (2003). New techniques that improve mace-style model finding. In P. Baumgartner & C. Fermüller (Eds.), *Model computation—principles, algorithms, applications* (Cade-19 Workshop), Miami, Florida, USA (pp. 11–27).

Clark, S., & Curran, J. R. (2004). 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 (pp. 104–111).

Cooper, R., Crouch, D., Van Eijck, J., Fox, C., Van Genabith, J., Jaspars, J., Kamp, H., Pinkal, M., Milward, D., Poesio, M., & Pulman, S. (1996). *Using the framework* (Technical report). FraCaS: A framework for computational semantics. FraCaS deliverable D16.

Curran, J., Clark, S., & Bos, J. (2007). Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, Prague, Czech Republic (pp. 33–36).

Dagan, I., Glickman, O., & Magnini, B. (2006). The Pascal recognising textual entailment challenge. In *Lecture notes in computer science* (Vol. 3944, pp. 177–190).

Gamut, L. (1991). *Logic, language, and meaning. Volume II. Intensional logic and logical grammar*. Chicago: University of Chicago Press.

Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Oxford: Blackwell Sci.

Honnibal, M., Curran, J. R., & Bos, J. (2010). Rebanking ccgbank for improved np interpretation. In *Proceedings of the 48th meeting of the association for computational linguistics* (ACL 2010), Uppsala, Sweden (pp. 207–215).

Kamp, H., & Reyle, U. (1993). *From discourse to logic; An introduction to modeltheoretic semantics of natural language, formal logic and DRT*. Dordrecht: Kluwer Academic.

Lin, D., & Pantel, P. (2001). DIRT—discovery of inference rules from text. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining* (pp. 323–328).

Minnen, G., Carroll, J., & Pearce, D. (2001). Applied morphological processing of English. *Journal of Natural Language Engineering*, *7*(3), 207–223.

Monz, C., & de Rijke, M. (2001). Light-weight entailment checking for computational semantics. In P. Blackburn & M. Kohlhase (Eds.), *Workshop proceedings ICoS-3* (pp. 59–72).

Muskens, R. (1996). Combining Montague semantics and discourse representation. *Linguistics and Philosophy*, *19*, 143–186.

Riazanov, A., & Voronkov, A. (2002). The design and implementation of vampire. *AI Communications*, *15*(2–3), 91–110.

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