Is there a place for logic in recognizing textual entailment?

Johan Bos[1]

From a purely theoretical point of view, it makes sense to approach recognizing textual entailment (RTE) with the help of logic. After all, entailment matters are all about logic. In practice, only few RTE systems follow the bumpy road from words to logic. This is probably because it requires a combination of robust, deep semantic analysis and logical inference—and why develop something with this complexity if you perhaps can get away with something simpler? In this article, with the help of an RTE system based on Combinatory Categorial Grammar, Discourse Representation Theory, and first-order theorem proving, we make an empirical assessment of the logic-based approach. High precision paired with low recall is a key characteristic of this system. The bottleneck in achieving high recall is the lack of a systematic way to produce relevant background knowledge. There is a place for logic in RTE, but it is (still) overshadowed by the knowledge acquisition problem.

1 Introduction

Recognizing textual entailment—predicting whether one text entails another—is a task that embraces everything that needs to be accomplished in natural language understanding. In the past, textual entailment was limited to the domain of formal semanticists, who used it as an illustrational device to show that certain natural language inferences hold or not (Gamut 1991; Chierchia and McConnell-Ginet 1991; Kamp and Reyle 1993; Heim and Kratzer 1998). By now, however, recognizing textual entailment (RTE, henceforth) is viewed by many as a key task in the area of natural language processing (Dagan et al. 2006).

[1]University of Groningen

LiLT Volume 9
Perspectives on Semantic Representations for Textual Inference.

In the early developments of approaches to RTE 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 component that gives an accurate analysis of text, the use of external resources to inform the inference process are also essential to achieve a good performance on the standard RTE data sets.

Various approaches to RTE have been proposed, ranging from "shallow" methods working directly on the surface features of texts, to "deep" methods using sophisticated linguistic analyses. The formalism proposed in this article belongs in the latter category, and works by determining textual inferences on the basis of deductive 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 one text entails the other or the other way around, whether they are consistent or contradictory, and so on.

Even though this idea itself sounds simple, its execution is not. In this article we describe a framework 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).[2] The input of this system is a text, and an hypothesis (another text). The output of the system is an entailment prediction for the hypothesis given the text. The system makes use of external theorem provers to calculate its predictions.

Performance on RTE data sets is measured in terms of recall (the number of correctly predicted entailments divided by the total number of text–hypothesis pairs given to a system) and precision (the number of correctly predicted entailments divided by the number of predictions made by the system). RTE systems based on logical inference tend to be low in recall and high in precision. This means that, currently, such systems ideally can play an important role in ensemble-based architectures of RTE systems, because they could complement simpler surface-based systems performing with higher recall and low precision.

The logical inference approach for RTE has been criticized by other RTE practitioners with respect to its low recall. However, in doing so, not always the correct explanation is given. MacCartney et al. (2006), for instance, write "few problem sentences can be accurately translated to logical form" when discussing Bos and Markert (2005), and al-

[2]The Nutcracker system has been briefly described by others (Balduccini et al. 2008), but never been the focus of publication itself. The source code of the system can be downloaded via the website of the C&C tools (Curran et al. 2007).

though one could debate the notion of accurate translation, it is doubtful whether this is the main reason for the lack of recall in RTE systems using deductive inference. In fact, one of the aims of this article is to show that logical inference is a promising approach to RTE, despite its limitations.

The rest of this article is organized as follows. First we explain what we mean by semantic interpretation in the context of RTE, and what formalism is useful for doing so, both from a theoretical and practical perspective. Then we make the link to (modal) first-order logic, in preparation for the inference tasks required for RTE. We then show which inference tasks are useful for the RTE task, and point out that supplementary background knowledge is required to increase recall. Finally we present the details of the Nutcracker system, a complete implementation of an RTE system based on logical reasoning, and will return to address the issue why RTE systems based on logical inference show low recall, and what can be done about it.

2 Semantic Interpretation

The challenge of translating ambiguous text into unambiguous logical formulas is usually performed by a detailed 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 high coverage. Its analysis needs to be sophisticated because a shallow analysis would not support the required logical inferences and hence sacrifice precision in performance. It needs to be robust and offer wide coverage to achieve a high recall in performance. As a practical rule of thumb, the loss in coverage should outweigh the gain in performance using deep linguistic analysis.

Due to the development of tree-banks in the past decades, many high-performing statistical parsers are available that offer broad coverage syntactic analysis for open-domain texts. The parser employed in our RTE system, the C&C parser (Clark and Curran 2004), combines speed and robustness with detailed syntactic analyses in the form of derivations of categorial grammar (Steedman 2001). Categorial grammar offers a neat way to construct formal meaning representations with the help of the λ-calculus (Bos 2008). 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 a complete meaning representation for the entire sentence will be produced.

Next we arrive at the choice of meaning representation language. This language needs to be capable of supporting logical inference, as well as being able to adequately describe natural language meaning. There is an uneasy and unsolved tension 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 are based on higher-order formalisms and 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). Nevertheless, there are powerful theorem provers for first-order logic available developed by the automated deduction research community. Hence, given the state-of-the-art in automated reasoning, the choice of first-order logic as representation language seems a good compromise between the ability to perform logical inferences and the expressive power for representing meaning.[3]

The standard first-order formula syntax is not a convenient format for meaning analysis. Instead we use a variant of Discourse Representation Theory's DRSs, Discourse Representation Structures, graphically visualized as boxes (Kamp and Reyle 1993). DRT offers a representational way to deal with many linguistic phenomena in a principled way, including quantifiers, pronouns, negation, presupposition and events. Diverging from standard DRT, we adopt a neo-Davidsonian way for describing events, because this results in a lower number of background knowledge rules (meaning postulates) required to draw correct inferences.

Another issue worth emphasizing is that we work with fully specified logical forms, despite many efforts in the past twenty years to produce underspecified semantic interpretations, in particular with respect to scope of quantifiers and other scope-bearing operators. Semantic underspecification is not a feasible option, because it is unclear how theorem provers would work with underspecified representations — they expect ordinary first-order formulas as input. Even when scope is resolved with a "naive" algorithm following mostly the surface order of scope-bearing

[3]Note that, in our framework, λ-calculus, a higher-order logic, only plays a role in meaning composition, and is not used for the inference tasks required for textual entailment prediction. This is basically the same strategy as put forward by Blackburn and Bos (2005).

operators, no harm seems to be done to the performance of RTE tasks. In fact, we have never encountered an example in the existing RTE data sets where correct scope resolution mattered for making a correct textual entailment prediction.

In sum: categorial grammar gives us a systematic and robust way to produce semantic representations from text; fully resolved first-order representations are a good practical choice for the basis of logical inference. In the next section we present how we produce such logical forms.

3 Semantic Representations and First-Order Logic

The RTE data sets consists of pairs of texts, and once we have established a method to produce semantic representations (DRSs in our case) for such pairs, we arrive at the problem of translating such DRSs into formulas of first-order logic (FOL). The result from this translation, FOL formulas, are given to a theorem prover to perform various inference tasks. One of them, the most important one, is to find out whether the text (T) entails the hypothesis (H). If the theorem prover then succeeds in finding a proof, we predict an entailment for this RTE pair. In this section we will discuss this translation, motivate the choice of theorem provers, and present basic results.

The standard translation from DRS to FOL (Muskens 1996, Kamp and Reyle 1993) is not suitable to RTE because it does not take modalities and embedded propositions into account. We will explain why this is a problem with the help of some examples. In Ex. 1, H is not entailed, because if John *thinks* that Mary smokes, it does not follow from this information that Mary does in fact smoke. Put differently, H contains new information, namely the fact that Mary smokes, which is information that cannot be deduced from T.

Example 1: H is informative wrt T

T: John thinks that Mary smokes.

H: Mary smokes.

A more general observation for attitude verbs like *think* and *believe* is: if X thinks/believes that P, then it doesn't mean that P. In contrast, factive verbs like *regret* and *know* result in an entailment of their propositional complement (Ex. 2).

These are hand-crafted examples to illustrate the point, but note that real-world examples of modal contexts are abundant. Ex. 3 below shows an example from the RTE data set in which the modal construc-

Example 2: H is entailed from T

T: John knows that Mary smokes.

H: Mary smokes.

tion in T blocks the inference hypothesized in H. Ex. 4 below shows a T–H pair with a subordinated clause introduced by *when*.

Example 3: H is informative wrt T

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.

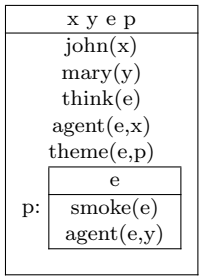
Example 4: H is entailed from T

T: When an earthquake rumbled off the coast of Hokkaido in Japan in July of 1993, the resulting tsunami hit just three to five minutes later, killing 202 people who were trying to flee for higher ground.

H: An earthquake occurred on the coast of Hokkaido, Japan.

In order to predict correct entailments for modal contexts, one needs lexical information about which verbs and adverbs entail their complements and which do not. In addition, one needs an adequate semantic interpretation of modal contexts – an issue to which we turn now.

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 cannot be formulas. The modal translation, that we adopt, is based on a technique called reification, as proposed for DRSs in Bos (2004). 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 (a particular kind of) entities. One could imagine these entities as ranging over "possible worlds" or simply "propositions". This extension in notation makes it possible to connect embedded propositions to attitudinal verbs or modal operators. We will not give the full translation from DRSs to modal FOL here (the interested reader is referred to Bos (2004)), but instead give an example translation of the DRS and first-order logic formulas for Ex. 1 to illustrate the approach.



$\exists w \exists x \exists y \exists e \exists p(\text{john}(w,x) \wedge \text{mary}(w,y) \wedge \text{think}(w,e) \wedge \text{agent}(w,e,x) \wedge \text{theme}(w,e,p) \wedge \text{smoke}(p,e) \wedge \text{agent}(p,e,y))$

The modal first-order translation above does not admit that Mary smokes, because the event where Mary smokes is established in connection with possible world p, which is not necessary the same as w, the actual world. But for certain verbs or other syntactic constructions we will add background knowledge axioms that force to make the actual world identical with a subordinated situation. We show how to do so in Section 5, but first we discuss how first-order theorem proving is integrated in our RTE framework.

4 Theorem Proving

In this section we show how to use off-the-shelf theorem provers for the task of recognizing textual entailment. Apart from checking whether there is an entailment between T and H, they can also be used for checking whether T or H contains contradictions or tautologies, or whether T and H together are contradictory or not. Such tests are also important in RTE, and we will discuss them first. We refer to them as consistency checking.

Consistency checking is important, because without doing so we might predict false entailments. In logic, anything follows from a contradiction. Hence, if T is inconsistent, H would automatically follow. It is questionable whether this is a desired outcome in the context of RTE. Consider the following example:

Example 5: Word Sense Ambiguity

T: A fan is a useful instrument.

H: The workers used a fan to prevent overheating.

In Ex. 5, the text T contains the ambiguous noun *fan*. Word sense disambiguation is a hard task and an RTE system might make the mistake of assigning the sense of *sports fan* or *admirer* to the noun *fan*, instead of the device sense. Together with the knowledge that people (sports fans, admirers) are disjoint from artifacts (instruments, de-

vices), this would lead to an inconsistent T. As a result, the RTE system would predict an entailment for Ex. 5.

Clearly, it would help an RTE system if such situations could be detected automatically. For instance, detection of a contradiction in T could give the RTE system reason to revise its background knowledge, even though as far as we know such systems have not been realized yet. Similarly, a clever RTE system would detect that the semantically ill-formed T in Ex. 6 is inconsistent, because an event cannot happen in the past as well as in the future. Examples of this kind do not occur in the current RTE data sets, but in real-world applications noisy data could yield such ill-formed texts.

Example 6: Inconsistent T

T: David Beckham had a tendon rupture tomorrow.
.....
H: David Beckham was fortunate.

For similar, logical reasons, we need to verify whether H is consistent or not. Because if H turns out to be inconsistent, checking whether T entails H boils down to verifying whether T is inconsistent, which is not the original goal of the inference task. Furthermore, we want to check whether T and H taken together are inconsistent. If this is so, we want to predict a non-entailment (for a two-way classification of entailment prediction), or report a contradiction between T and H (in the case of a three-way classification of entailment prediction).

In sum, we need to check whether T is consistent, H is consistent, and T∧H is consistent. We do this by translating them to modal first-order logic, and trying to prove their negation. At the same time we attempt to find a counter-model by using a finite model builder. If a counter-model is found, the theorem prover can be halted, which is a way to save valuable resources (time and memory). In addition, we try to find a proof for $T \rightarrow H$ (or the logically equivalent $\neg(T \wedge \neg H)$. Table 1 summarizes the situation.[4]

5 Adding Background Knowledge

For a good performance on RTE examples not only translations of T and H in (modal) first-order logic are required—what is crucial for an increase in recall is a set of background knowledge axioms. Such axioms

[4]One could also extend the inference tasks by explicitly verifying whether T and H are tautologies or not. If T is logically valid (i.e. a tautology), then it would not make sense to test whether T entails H. Similarly, if H is a validity, T would always entail it.

TABLE 1: Inference Tasks for RTE and corresponding predictions based on proofs or countermodels.

Input	**Output**				
¬ T	proof	model	model	model	model
¬ H	–	proof	model	model	model
¬(T ∧ H)	–	–	proof	model	model
¬(T ∧ ¬H)	–	–	–	proof	model
Prediction	unknown	unknown	contradiction	entailment	informative

need to be stated in modal first-order logic too, and can be added to the inference requests, simply as additional background theory. In the inference examples above, this can be achieved by replacing T by (BK ∧ T). This is one of the attractive sides of a logic-based approach: background knowledge can be supplied in a modular way.

Axioms are generally of the form $\forall w \forall x(\phi(w,x) \rightarrow \psi(w,x))$, where ϕ and ψ denote first-order formulas. Here we discuss three types of background knowledge axioms:

1. Axioms automatically derived from synonym and hyponym relations between WordNet synsets;
2. Manually encoded axioms for propositional embeddings;
3. Complex axioms automatically derived from positive RTE pairs.

The number of general background knowledge axioms can be very large. But given a textual entailment problem, we do not want to give irrelevant background knowledge to the theorem prover and waste its resources. It remains an interesting research challenge to select appropriate axioms—axioms that are likely to increase the chance of finding a proof.

A simple way to solve this problem is to associate *triggers* with axioms (Blackburn and Bos 2005). The non-logical symbols in meaning representations are useful triggers for many types of axioms, as long as axioms themselves are not able to initiate the triggering of new axioms, thereby risking a chain reaction resulting in the selection of the entire knowledge base. Following this approach, each type of axiom is illustrated by a T–H pair that triggers it.

Axioms derived from WordNet

Let us start with axioms derived from the WordNet relations. Consider Ex. 7. In WordNet (Fellbaum 1998), the first sense of the noun *role* is a hyponym of the second sense of *duty*, which in turn is a hyponym

of the first sense of *activity*. Similarly, *murder* is a hyponym of *kill* in WordNet, enabling a proof for Ex. 8. We note in passing that this example also demonstrates the benefits from a deep linguistic analysis that assigns syntactically equivalent meaning representations to active and passive forms, as our system does.

Example 7: T entails H, hyponymy

T: The World Bank has also been criticized for its role in financing projects.

H: The World Bank is criticized for its activities.

Example 8: T entails H, active-passive alternation

T: Lennon was murdered by Mark David Chapman outside the Dakota on Dec. 8, 1980.

H: Mark David Chapman killed Lennon.

Example 9: T entails H, synonymy

T: The two presidents, Bush and Chirac, were honored with a 21-gun salute.

H: The two presidents, Bush and Chirac, were honoured with a 21-gun salute.

The WordNet hyponym relation is translated into first-order logic as an implication. As we use the modal translation, we need to include possible worlds in the generated background knowledge. As a consequence, we end up with the following set of axioms for the examples above:

$\forall$w(possible-world(w)$\rightarrow$ $\forall$x(n1role(w,x)$\rightarrow$n2duty(w,x)))
$\forall$w(possible-world(w)$\rightarrow$ $\forall$x(n2duty(w,x)$\rightarrow$n1work(w,x)))
$\forall$w(possible-world(w)$\rightarrow$ $\forall$x(n1work(w,x)$\rightarrow$n1activity(w,x)))
$\forall$w(possible-world(w)$\rightarrow$ $\forall$x(v1murder(w,x)$\rightarrow$v1kill(w,x)))

It is easy to see that such axioms can be systematically generated from WordNet.[5] Apart from hyponyms, we can also explore synonyms

[5]Note that the non-logical symbols are composed using part-of-speech information (noun, verb, modifier) and a sense number, to avoid unwanted clashes of symbols derived from the same words with different meanings. That is, we want to have different non-logical symbols for the noun *fly*, the verb *fly*, and the adjective *fly*, because they mean different things. Similarly, we would like to distinguish

stored in WordNet. In WordNet, synset members are considered to correspond to equivalent concepts. A case in point is Ex. 9, where we can observe that *honor* and *honour* are members of the same synset in WordNet. Members of the same synset are translated into axioms with a bi-implication. Returning to Ex. 9, we trigger the following axiom:

$\forall$w(possible-world(w)$\rightarrow$ $\forall$x(v1honor(w,x)$\leftrightarrow$v1honour(w,x)))

There is more information in WordNet that could form the basis for background knowledge axioms. The antonymy relation found between adjectives is a good candidate. But other lexical resources could supply useful information too. The NomLex database (Meyers et al. 1998) provides information about normalizations, thereby making it possible to compute background knowledge axioms that relate concepts and events.

Axioms for embedded contexts

The axioms for embedded contexts all follow the same pattern. They are manually picked for sentential complement verbs like *know*, *regret*, *say*, *report*, *tell*, *reveal*, as well as for sentential adverbs such as *because*, *although* and *when*, that presuppose their subordinated sentential argument. They are manually selected because existing lexical resources such as WordNet do not contain this information. Ex. 10 illustrates the idea behind this type of axioms:

Example 10: T entails H, sentential complement

T: Authorities say Monica Meadows, who has appeared in catalogs and magazines, is in stable condition.

H: Monica Meadows is in stable condition.

The required background knowledge is that the information of the theme of a *saying* event also holds in the world in which this event was expressed. The relevant axiom is the following.

$\forall$w(possible-world(w) $\rightarrow$ $\forall$x$\forall$y(v1say(w,x) $\wedge$ r1theme(w,x,y) $\leftrightarrow$ w = y))

This axiom template is accurate for factive verbs, but in general not for reporting verbs.[6] All what is said is not necessarily true, and we would like to exclude, for instance, liars. In Ex. 10, it is the source of

between the different senses of words. For example, n2duty is the symbol for the second noun sense of the word *duty*.

[6]There is also a connection with presupposition projection here (Beaver 1997). Factive verbs such as *regret* presuppose their propositional complement. If presupposition projection is implemented by the semantic formalism, then these axioms would not be needed.

the information, the *authorities*, that cause the textual entailment of H with respect to T. In fact, in most newspaper examples reporting verbs entail the content of their propositional complement. In general however, one wants to strengthen the axioms of reporting verbs, by including a constraint on the reliability of the agent of the reporting event.

Automatically learned axioms

The third type of axiom can be automatically learned from positive T–H pairs of the available RTE data sets (Ihsani 2012). The idea here is to identify a pattern between two entities that appear both in T and H. If the same pattern is observed in different T–H pairs, then this indicates that it might be a valid and useful background knowledge axiom. In Ex. 11, the complex relations between *Tilda Swinton* and *White Witch* in T and H suggest the axiom that "X playing a role as Y implies that X plays the part of Y".

Example 11: T entails H, complex axiom

T: Tilda Swinton has a prominent role as the White Witch in The Chronicles of Narnia: The Lion, The Witch and The Wardrobe, coming out in December.

H: Tilda Swinton plays the part of the White Witch.

Ihsani (2012) presents a method to automatically generate such axioms from positive T–H pairs, and tested on negative T–H pairs (inclusion of a learned axiom should not result in a proof). The axiom automatically generated for the above example is:

$$\forall \text{w}(\text{possible-world}(\text{w}) \rightarrow$$
$$\forall \text{x} \forall \text{y} \forall \text{z}(\exists \text{e}(\text{h}\textit{ave}(\text{w,e}) \wedge \text{agent}(\text{w,e,x}) \wedge \text{theme}(\text{w,e,y}) \wedge$$
$$\text{role}(\text{w,y}) \wedge \text{as}(\text{w,y,z})) \rightarrow$$
$$\exists \text{e}(\text{play}(\text{w,e}) \wedge \text{agent}(\text{w,e,x}) \wedge \text{theme}(\text{w,e,y}) \wedge \text{part}(\text{w,y}) \wedge \text{of}(\text{w,y,z}))))$$

Lin and Pantel (2001) present an unsupervised algorithm, DIRT, for discovering inference rules, such as *X is the author of Y* $\approx$ *X writes Y*, by applying the distributional hypothesis to syntactic dependency analysis. The method of Ihsani (2012) could be viewed as a variation of this, but differs in the level of supervision during learning (DIRT is unsupervised). The level of linguistic analysis is also different, as DIRT produces (non-directional) surface string paraphrases, and Ihsani's method yields (directional) first-order axioms. In general, Ihsani's method produces

axioms with high precision and low recall, while DIRT tends to yield opposite results (Szpektor et al. 2007).

6 Implementation and Evaluation

The framework presented before has been implemented in a complete RTE system known as Nutcracker. The system (including source code) is distributed as part of the C&C tools (Clark and Curran 2004). A description of the most important components of this complex system follows below.

The Nutcracker system has a traditional pipeline architecture of components, starting with a tokenizer, POS tagger, lemmatizer (Minnen et al. 2001) and named entity recognizer. This is followed by syntactic and semantic parsing. The meaning representations are produced by the semantic interpreter Boxer (Bos 2008), which works on the output of the C&C parser, based on Combinatory Categorial Grammar. Boxer performs pronoun resolution, presupposition projection, thematic role labeling and assigns scope to quantifiers, negation and modal operators.

The coverage of the pipeline—meaning the percentage of examples for which a semantic representation could be produced—on RTE examples is high, reaching nearly 98% on the examples of the RTE data sets. Remember that the parser's statistical model is not specifically trained on examples of these data sets. The NLP pipeline formed by the C&C tools and Boxer is therefore suitable for a task such as RTE, contrary to what MacCartney et al. (2006) suggest. Note however, that high coverage does not always mean high correctness, but at present no corpora with gold-standard annotated semantic representations are available to measure accuracy.

The end of the pipeline is formed by a theorem provers and model builders. 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. The Nutcracker systems allows us to plug in several different provers, among them Vampire (Riazanov and Voronkov 2002), Otter (McCune and Padmanabhan 1996), and Bliksem (De Nivelle 1998). Vampire is currently the highest-ranked prover in CASC, the annual competition for inference engines (Sutcliffe and Suttner 1997), and it also gives the best results on RTE examples.

In addition to a theorem prover, a model builder is needed to find counter-models. Again, various model builders can be used with Nutcracker, including Mace (McCune 1998) and Paradox (Claessen and Sörensson 2003). Following Blackburn and Bos (2005), for each in-

ference problem 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 cannot construct a model for sizes $1 - n$. In theory, because first-order logic is semi-decidable, the combination of theorem proving and finite model building always terminates with one of three results: (i) proof found, (ii) no proof but finite counter-model found of size n, or (iii) no proof and no model for size n (for instance for inputs that have non-finite counter-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.

The performance of Nutcracker, without supplying background knowledge axioms, on the RTE data sets shows that only few proofs are found (61 for all the 3,200 examples RTE-2 and RTE-3 data sets) but with high precision (54 correct, yielding 88.5%). This shows that, without appealing to further background knowledge, a high-precision performance paired with a low recall is achieved. This is not a big surprise. Many of the examples from the RTE data sets require additional information to draw the wanted inferences. Ihsani (2012) shows that some of these background knowledge axioms can be retrieved using supervised learning. Axioms based on synonym and hyponym relations extracted from WordNet give only a small increase of recall (12 extra proofs found, of which 11 correct). WordNet relations combined with modality axioms gives a further increase in recall (21 extra proofs found, of which 18 correct). Adding automatically generated axioms based on positive T–H pairs yields 52 extra proofs, of which 46 correct (Ihsani 2012). These numbers indicate that recall can be increased without a loss of precision, when appropriate background knowledge can be selected.

7 Related Work

Compared to other RTE approaches, closely related to the "logical approach" are systems based on Natural Logic. The Natural Logic approach is an interesting alternative to logical inference because it is more flexible (resulting in more robust systems) yet based on local log-

ical inferences. The best known example (and implementation) in this tradition is NatLog (MacCartney 2009), which we will compare to our Nutcracker system.

Given an RTE pair T–H, NatLog works by a sequence of components, to wit (1) parsing T and H; (2) aligning T and H with a sequence of local edit operations turning T into H; (3) predicting entailment relations for each of these local edit operations; (4) joining the local entailment relations to produce an entailment prediction for the entire T–H pair. The NatLog system uses lexical resources (including WordNet and NomBank) and also information on string similarity to predict local entailments, with the help of a statistical classifier.

The Natural Logic approach is interesting because it does not use the full power of FOL (in fact, as MacCartney (2009) shows, it is incomplete), yet it makes use of (local) logical inference and performs well on tasks such as RTE, with a lower precision than Nutcracker, but with a much higher recall (MacCartney 2009). A disadvantage of the approach is that the alignment procedure excludes texts consisting out of more than one sentence.

Like the background knowledge axioms for the Nutcracker system, the NatLog system has to get the local entailment predictions from external resources, and for an informative comparison it would be interesting to see how well NatLog would perform without appealing to lexical resources and similarity measurements. Equally interesting, it would be an informative exercise to translate the local inference rules obtained by the NatLog system, transform them into first-order axioms, and feed them into the Nutcracker system, and measure performance differences.

8 Conclusion

The logical approach to RTE is costly—one needs to perform all steps of linguistic analysis ranging including detailed syntactic and semantic analysis. Current semantic parsers reach high coverage and are able to produce reasonably adequate semantic representations for RTE. This is at least what the available data sets for RTE suggest. Translating T–H pairs into first-order formulas result in input that state-of-the-art theorem provers can easily digest most of the time, reaching high precision. Nonetheless, without additional background knowledge, recall is low. Such background knowledge can be provided as additional first-order axioms, but they are hard to generate in a domain-independent manner. Experiments shows however that such additional background knowledge raises recall without a (big) loss in precision. The bottle-

neck of logical inference in RTE is not the inability to translate text to logical formulas as; it is not the performance of theorem provers; but it is the lack of a systematic way to produce relevant background knowledge.

References

Balduccini, M., C. Baral, and Y. Lierler. 2008. Knowledge representation and question answering. In V. Lifschitz, F. van Harmelen, and B. Porter, eds., *Handbook of Knowledge Representation*, pages 779–819. Elsevier.

Beaver, David Ian. 1997. Presupposition. In J. Van Benthem and A. Ter Meulen, eds., *Handbook of Logic and Language*, chap. 17, pages 939–1008. Elsevier, MIT.

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

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

Bos, Johan. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, eds., *Semantics in Text Processing. STEP 2008 Conference Proceedings*, vol. 1 of *Research in Computational Semantics*, pages 277–286. College Publications.

Bos, Johan and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 628–635.

Chierchia, Gennaro and Sally McConnell-Ginet. 1991. *Meaning and Grammar. An Introduction to Semantics*. The MIT Press.

Claessen, K. and N. Sörensson. 2003. New techniques that improve mace-style model finding. In P. Baumgartner and C. Fermüller, eds., *Model Computation – Principles, Algorithms, Applications (Cade-19 Workshop)*, pages 11–27. Miami, Florida, USA.

Clark, Stephen and James R. Curran. 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)*, pages 104–111. Barcelona, Spain.

Curran, James, Stephen Clark, and Johan Bos. 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*, pages 33–36. Prague, Czech Republic.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Lecture Notes in Computer Science*, vol. 3944, pages 177–190.

De Nivelle, Hans. 1998. A Resolution Decision Procedure for the Guarded Fragment. In *Automated Deduction - CADE-15. 15th International Conference on Automated Deduction*, pages 191–204. Springer-Verlag Berlin Heidelberg.

Fellbaum, Christiane, ed. 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

Gamut, L.T.F. 1991. *Logic, Language, and Meaning. Volume II. Intensional Logic and Logical Grammar*. Chicago and London: The University of Chicago Press.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Malden and Oxford: Blackwell.

Ihsani, Annisa. 2012. *Automatic Induction of Background Knowledge Axioms for Recognising Textual Entailment*. Master's thesis, University of Groningen.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Dordrecht: Kluwer.

Lin, Dekang and Patrick Pantel. 2001. DIRT—discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328.

MacCartney, Bill. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford University.

MacCartney, Bill, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 41–48. Stroudsburg, PA, USA: Association for Computational Linguistics.

McCune, W. 1998. Automatic Proofs and Counterexamples for Some Ortholattice Identities. *Information Processing Letters* 65(6):285–291.

McCune, W. and R. Padmanabhan. 1996. *Automated Deduction in Equational Logic and Cubic Curves*. No. 1095 in Lecture Notes in Computer Science (AI subseries). Springer-Verlag.

Meyers, A., C. Macleod, R. Yangarber, R. Grishman, L. Barrett, and R. Reeves. 1998. Using nomlex to produce nominalization patterns for information extraction. In *Coling-ACL98 workshop Proceedings, The Computational Treatment of Nominals*, pages 25–32. Montreal, Canada.

Minnen, Guido, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Journal of Natural Language Engineering* 7(3):207–223.

Muskens, Reinhard. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy* 19:143–186.

Riazanov, A. and A. Voronkov. 2002. The Design and Implementation of Vampire. *AI Communications* 15(2–3):91–110.

Steedman, Mark. 2001. *The Syntactic Process*. The MIT Press.

Sutcliffe, Geoff and Christian Suttner. 1997. The results of the cade-13 atp system competition. *Journal of Automated Reasoning* 18(2):259–264. Special Issue on the CADE-13 Automated Theorem Proving System Competition.

Szpektor, Idan, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463. Prague, Czech Republic: Association for Computational Linguistics.