Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment

Johan Bos
School of Informatics
University of Edinburgh
jbos@inf.ed.ac.uk

Katja Markert
School of Computing
University of Leeds
markert@comp.leeds.ac.uk

Abstract

We combine two methods to tackle the textual entailment challenge: a shallow method based on word overlap and a deep method using theorem proving techniques. We use a machine learning technique to combine features derived from both methods. We submitted two runs, one using all features, yielding an accuracy of 0.5625, and one using only the shallow feature, with an accuracy of 0.5550. Our method currently suffers from a lack of background knowledge and future work will be focussed on that area.

1 Introduction

In this paper we summarise our submission to the 2004/5 Recognising Textual Entailment (RTE) challenge. In this task, given a pair of text fragments—a text (T) and an hypothesis (H)—the system has to decide whether the hypothesis is entailed by the text. The system we developed is a hybrid system, using both shallow and deep semantic analysis methods.

The shallow techniques establish a baseline performance, but also complement the deep semantic analysis. In the hybrid system, each T/H-pair is represented by feature-value vectors that are derived from either shallow or deep semantic analysis. The features used are domain-independent to increase scalability. An off-the-shelf machine learning tool was then used to derive a decision tree model from the RTE development set.

2 Shallow Semantic Analysis

The shallow semantic analysis measures only word overlap between text and hypothesis. Both text and hypothesis are tokenised and lemmatised. Each lemma in the hypothesis is assigned its inverse document frequency, using the Web as corpus, as its weight. This standard procedure allows us to assign more importance to less frequent words.

The word overlap `overlap` between text and hypothesis is initialised as zero. Should a lemma in the hypothesis also occur in the text, its weight is added to `overlap`, otherwise it is substracted. In the end `overlap` is normalised by dividing it by the sum of all weights of the lemmas in the hypothesis. This ensures that `overlap` is always a real number between 1 and -1 and also ensures independence of the length of the hypothesis.[1]

Training a decision tree on the development set with this feature alone yielded the following tree for entailment, where TRUE associates with entailment, and FALSE does not:[2]

```
overlap <= 0.161146: FALSE
overlap > 0.161146: TRUE
```

Accuracy on the development set (using 10-fold cross-validation) was 0.594 and therefore clearly beat the baseline of 0.50. In general this method overestimates the number of true entailments in the development set and achieved an F-measure of 0.672 for the class TRUE and only 0.474 for the class

[1] This word overlap measure is similar to the method used in (Monz and de Rijke, 2003) and (Saggion et al., 2004) —however, they do not substract from the overlap measure a token in the hypothesis which does not appear in the text. Hence, their scores are within 0 and 1. We experimented with this variation on the development set, but achieved slightly better performance with the scores that used substraction as well.

[2] We used Weka's J48 classifier (`http://www.cs.waikato.ac.nz/~ml/weka/`) for all experiments in this paper. We also used Weka's confidence values for confidence weighting scores.

FALSE. We submitted this baseline as Run2 and the performance on the RTE test set was as follows:

```
cws:       0.5864
accuracy:  0.5550
precision: 0.5375
recall:    0.7875
f:         0.6389
```

Although the performance is still significantly better than the baseline (5% level), it is worse than on the development set, because the level of word overlap in the test set was lower overall than in the development set. This seems to be an indicator of a different design of development and test set—using 10-fold cross-validation on the test set indicates that an `overlap` value of between -0.20 and 0.92 already indicates a TRUE value in the test set, whereas a value of over 0.92 indicates a FALSE value. The latter anomaly, which indicates that if text and hypothesis are very similar then the entailment is false, is due to the fact that there are many examples in the test set that are deliberately constructed to have a high word overlap but nevertheless be FALSE.

3 Deep Semantic Analysis

We use a robust wide-coverage CCG-parser (Bos et al., 2004) to generate fine-grained semantic representations for each T/H-pair. The semantic representation language is a first-order fragment of the DRS-language used in Discourse Representation Theory (Kamp and Reyle, 1993). To check whether an entailment holds or not, we used Vampire, a theorem prover for first-order logic (Riazanov and Voronkov, 2002), and Paradox, a finite model builder (Claessen and Sörensson, 2003).

To support the proofs we calculated background knowledge using three kinds of sources:

- Generic axioms for, for instance, the semantics of possessives, active-passives, and locations.
- Lexical knowledge that was created on the fly with an algorithm that takes as input the DRSs for the text and hypothesis, and outputs first-order axioms based on WordNet hypernyms. This algorithm also performs simple word sense disambiguation and analysis of complex concepts.
- Geographical knowledge from the CIA factbook was translated into first-order axioms.

To perform the actual search for a proof, the DRSs for T and H were translated into first-order logic. The theorem prover and model builder were used in all tasks as complementary inference engines, where the theorem prover attempts to prove the input, and the model builder tries to find a model for the negation of the input. First we checked whether the background knowledge (BK) was consistent with the text, by giving $\neg$(BK $\wedge$ T) to the theorem prover. If there is a proof, indicating that the background knowledge is inconsistent, we proceed with checking for entailment without background knowledge, by giving (T $\rightarrow$ H) to the theorem prover. Otherwise we attempt to prove (BK $\wedge$ T $\rightarrow$ H).

Although in theory the method of finding proofs should work, in practice it does not work that well. This is mostly due to the lack of appropriate background knowledge without which many true entailments cannot be found. To overcome this problem we also used a novel way of measuring approximate entailments, relying on the model sizes computed by the model builder. Using Paradox, we computed the model size of (BK$\wedge$T) and that of (BK$\wedge$T$\wedge$H). The underlying idea was that if the difference of these two numbers is small, it is likely to be an entailment. (In other words, the hypothesis does not introduce any or little new information.)

This deep semantic analysis proposes a number of features to describe the T/H-pairs:

`entailed`	{proof,unknown}
`inconsistent`	{proof,unknown}
`domainsize`	numeric
`domainsizeabsdif`	numeric
`domainsizereldif`	numeric
`modelsize`	numeric
`modelsizeabsdif`	numeric
`modelsizereldif`	numeric
`negation`	{yes,no}
`negationtext`	{yes,no}
`negationhypo`	{yes,no}

The features `entailed` and `inconsistent` have been discussed above. `domainsize` is the value of the domainsize of the model for both T and H, `domainsizeabsdif` is the absolute difference between the domain sizes of T and H, and `domainsizereldif` the difference relative to the model size. The `modelsize` is computed by multiplying the domain size with the number of all positive two-place predicates in the model. The features `negation`, `negationtext`, and `negationhypo` are determined by inspecting the DRSs for the presence of negation operators.

4 Combining the Methods

For the combined run we used all shallow and deep features for training a decision tree on the development set. The tree generated for the development data is displayed below:

```
entailed = proof: TRUE
entailed = unknown
|  negationhypo = yes: FALSE
|  negationhypo = no
|  |  overlap <= 0.161146: FALSE
|  |  overlap > 0.161146
|  |  |  inconsistent = proof: TRUE
|  |  |  inconsistent = unknown
|  |  |  |  domainsize <= 8
|  |  |  |  |  negation = yes: FALSE
|  |  |  |  |  negation = no
|  |  |  |  |  |  domainsize <= 6
|  |  |  |  |  |  |  domainsizeabsdif <= 0: TRUE
|  |  |  |  |  |  |  domainsizeabsdif > 0
|  |  |  |  |  |  |  |  modelsizereldif <= 0.595556: TRUE
|  |  |  |  |  |  |  |  modelsizereldif > 0.595556: FALSE
|  |  |  |  |  |  domainsize > 6: FALSE
|  |  |  |  domainsize > 8: TRUE
```

Note that not all features were used (negation in the text, relative domain size difference, model size, and absolute model size difference were not used).

We did not expect good results, as experiments using cross-validation on the development data yielded around 60% accuracy (depending on the decision tree parameters). However, on the test set, this run performed better than the baseline at the 1% level and slightly better than the shallow feature alone. The actual results on the test set are detailed below.

```
cws:       0.5931
accuracy:  0.5625
precision: 0.5530
recall:    0.6525
f:         0.5986
```

5 Error Analysis

The hybrid system was able to create semantic representations and then search for proofs for 774 of all 800 T/H-pairs in the test data, achieving a coverage of 96.8%. Only 30 proofs were found by the system, of which 23 were annotated as entailments in the gold standard. These include adequately analysed phenomena such as apposition (5x: 760, 929, 995, 1903, 1905), relative clauses (3x: 142, 1060, 1900), coordination and attachment(3x: 898, 807, 893), active-passive alternation (2x: 1007, 1897), possessives (1x: 1010), the use of background knowledge (6x: 236, 836, 1944, 1952, 1987, 1994) and more or less straightforward cases (3x: 833, 1076, 741). Note that two examples are included that were annotated as entailment, but strictly speaking they are not (Examples 893 and 236, see also Section 6).

Incorrect proofs were found for seven cases. Some of these are due to the lexical semantics of certain linguistic categories, others to a lack of background knowledge. As an example, the current system does not deal adequately with ordinals and thus finds proofs for 1617 (see below) and 2040.

Example: 1617

T: In 1782 Martin Van Buren, the first US president who was a native citizen of the United States, was born in Kinderhook, N.Y.

H: The first US president was born in Kinderhook, N.Y.

It also found a proof for 2025, where the text contained the hypothesis in an if-clause. Again this was due to an incorrect lexical semantics, and is easy to fix. More complex cases involving modifiers were 2030 and 2082 (see below). It is hard to see what kind of background knowledge can preclude proofs for such cases. (For 2030, the knowledge that Paris is the capital of France, and that each country has at most one capital, would suffice. Unfortunately our system does not select this as background knowledge because the trigger Paris is mentioned neither in the text nor in the hypothesis.)

Example: 2030

T: Lyon is actually the gastronomic capital of France.

H: Lyon is the capital of France.

Example: 2082

T: Microsoft was established in Italy in 1985.

H: Microsoft was established in 1985.

For 2055, the system correctly associated Einstein to be the subject of being the president of Israel, but it incorrectly assumed that begin invited to X is being X. A restriction on this class of modal verbs could fix this problem. (In the development data, however, there were similar cases that were annotated as entailments.)

Example: 2055

T: The fact that Einstein was invited to be the president of Israel is critical to an accurate understanding of one of the greatest individuals in modern history.

H: Einstein is the president of Israel.

Finally, background knowledge that if X is in Y, then X is located in Y, wrongly predicted an entailment for 2079. A more sophisticated lexical analysis of prepositions could improve on such examples.

Example: 2079

T: US presence puts Qatar in a delicate spot.

H: Qatar is located in a delicate spot.

In sum, the backbone of the deep semantic analysis, trying to find proofs, has a small coverage, but is reasonably accurate. Selecting more appropriate background knowledge and revising some of the lexical semantics will improve its precision. We already improved its recall by incorporating the features concerning model size differences.

6 Discussion of the entailment task

We will now discuss some observations we made on the task definition and the annotated data sets.

Task definition The current RTE dataset classified entailment as binary TRUE and FALSE. Following FRACAS, the semantic test suite in (Coopper et al., 1996), a classification that respects three values (yes, don't know, inconsistent), is probably more in its place. For instance, not only are examples 1301 and 1310 below not entailments, the hypotheses are inconsistent with the corresponding texts as well:

Example: 1301

T: The former wife of the South African president did not ask for amnesty, and her activities were not listed in the political reports submitted by the African National Congress to the Truth and Reconciliation Commission in 1996 and 1997.

H: Winny Mandela, the President's ex-wife, is requesting amnesty.

Example: 1310

T: Although the hospital insists that King Hussein is not fully free of the cancer, they are hopeful that he will recover.

H: The statement added that King Hussein has been cured completely.

In the current RTE task definition FALSE subsumes both the "don't know" and "inconsistent" values used in the FRACAS test suite.

Annotated datasets We found several cases where entailments were incorrectly annotated in our opinion. Example 236 (see below), for instance, was judged as entailment. But taking tense into account (which, incidentally, our system is currently not able to do), it is strictly speaking not a textual entailment.

Example: 236

T: Yasir Arafat has agreed to appoint a longtime loyalist as interior minister to take charge of the country's security.

H: Yasir Arafat nominated a loyalist as interior minister.

Another example is 893: the adverb *perhaps* in the text clearly expresses doubt on the date of establishment of settlements on Jakarta, and the hypothesis establishes it as a fact. This clearly is not entailment.

Example: 893

T: The first settlements on the site of Jakarta were established at the mouth of the Ciliwung, perhaps as early as the 5th century AD.

H: The first settlements on the site of Jakarta were established as early as the 5th century AD.

It would also be helpful if human agreement figures and explicit guidelines for annotation could be released for the task. For a small test, one of the authors annotated all 800 examples of the test set for entailment, using the short rules that were indicated on the entailment web page (for example, disregarding tense). Comparing to the final gold standard, now released, we had 38 differences, yielding an agreement of 95.25%. This indicated good agreement, but one has to take into account that both annotations used the indicated simplified guidelines.

References

J. Bos, S. Clark, M. Steedman, J. Curran, and J. Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proc of the 20^{th} International Conference on Computational Linguistics; Geneva, Switzerland; 2004*.

K. Claessen and N. Sörensson. 2003. New techniques that improve mace-style model finding. In *Model Computationa - Principles, Algorithms, Applications (Cade-19 Workshop)*, Miami, Florida.

R. Coopper, R. Cropuch, J. VanEijck, C. Fox, J. Van Genabith, J. Jaspars, H. Kamp, M. Pinkal, D. Milward, M. Poesio, and S. Pulman. 1996. Using the framework. fracas: A framework for computationla semantics. Technical report, Fracas Deliverable D16.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht, Netherlands.

C. Monz and M. de Rijke. 2003. Light-weight entailment checking for computational semantics. In *Proc. of the 3^{rd} Workshop on Inference in Computational Semantics; 2003*.

A. Riazanov and A. Voronkov. 2002. The design and implementation of Vampire. *AI Communications*, 15(2-3).

H. Saggion, R. Gaizauskas, M. Hepple, I. Roberts, and M Greenwood. 2004. Exploring the performance of boolean retrieval strategies for open domain question answering. In *Proc. of the Information Retrieval for Question Answering (IR4QA) Workshop at SIGIR 2004*.