Cambridge: Parser Evaluation using Textual Entailment by Grammatical Relation Comparison

Laura Rimell and **Stephen Clark**
University of Cambridge
Computer Laboratory
{laura.rimell,stephen.clark}@cl.cam.ac.uk

Abstract

This paper describes the Cambridge submission to the SemEval-2010 Parser Evaluation using Textual Entailment (PETE) task. We used a simple definition of entailment, parsing both T and H with the C&C parser and checking whether the core grammatical relations (subject and object) produced for H were a subset of those for T. This simple system achieved the top score for the task out of those systems submitted. We analyze the errors made by the system and the potential role of the task in parser evaluation.

1 Introduction

SemEval-2010 Task 12, Parser Evaluation using Textual Entailment (PETE) (Yuret et al., 2010), was designed as a new, formalism-independent type of parser evaluation scheme. The task is broadly Recognizing Textual Entailment (RTE), but unlike typical RTE tasks, its intention is to focus on purely syntactic entailments, assuming no background knowledge or reasoning ability. For example, given a text (T) *The man with the hat was tired.*, the hypothesis (H) *The man was tired.* is entailed, but *The hat was tired.* is not. A correct decision on whether H is entailed can be used as a diagnostic for the parser's analysis of (some aspect of) T. By requiring only a binary decision on the entailment, instead of a full syntactic analysis, a parser can be evaluated while its underlying formalism remains a "black box".

Our system had two components: a parser, and an entailment system which decided whether T entails H based on the parser's output. We distinguish two types of evaluation. *Task evaluation*, i.e. the official task scoring, indicates whether the entailment decisions – made by the parser and entailment system together – tally with the gold standard dataset. *Entailment system evaluation*, on the other hand, indicates whether the entailment system is an appropriate parser evaluation tool. In the PETE task the parser is not evaluated directly on the dataset, since the entailment system acts as intermediary. Therefore, for PETE to be a viable parser evaluation scheme, each parser must be coupled with an entailment system which accurately reflects the parser's analysis of the data.

2 System

We used the C&C parser (Clark and Curran, 2007), which can produce output in the form of grammatical relations (GRs), i.e. labelled head-dependencies. For example, `(nsubj tired man)` for the example in Section 1 represents the fact that the NP headed by *man* is the subject of the predicate headed by *tired*. We chose to use the Stanford Dependency GR scheme (de Marneffe et al., 2006), but the same approach should work for other schemes (and other parsers producing GRs).

Our entailment system was very simple, and based on the assumption that H is a simplified version of T (true for this task though not for RTE in general). We parsed both T and H with the C&C parser. Let grs(S) be the GRs the parser produces for a sentence S. In principle, if grs(H) $\subseteq$ grs(T), then we would consider H an entailment. In practice, a few refinements to this rule are necessary.

We identified three exceptional cases. First, syntactic transformations between T and H may change GR labels. The most common transformation in this dataset was passivization, meaning that a direct object in T could be a passive subject in H.

Second, H could contain tokens not present in T. Auxiliary verbs were introduced by passivization. Pronouns such as *somebody* and *something* were introduced into some H sentences to indicate an NP or other phrase not targeted for evaluation. Determiners were sometimes introduced or changed, e.g. *prices* to *the prices*. Expletive subjects were also sometimes introduced.

Third, the parses of T and H might be inconsistent in an incidental way. Consider the pair *I reached into that funny little pocket that is high up on my dress.* ⇒ *The pocket is high up on something.* The intended focus of the evaluation (as indicated by the content word pair supplied as a supplement to the gold standard development data) is *(pocket, high)*. As long as the parser analyzes *pocket* as the subject of *high*, we want to avoid penalizing it for, say, treating the PP *up on X* differently in T and H.

To address these issues we used a small set of heuristics. First, we ignored any GR in grs(H) containing a token not in T. This addressed the passive auxiliaries, pronouns, determiners, and expletive subjects. Second, we equated passive subjects with direct objects. Similar rules could be defined for other transformations, but we implemented only this one based on the prevalence of passivization in the development data. Third, when checking whether grs(H) $\subseteq$ grs(T), we considered only the core relations subject and object. The intention was that incidental differences between the parses of T and H would not be counted as errors. We chose these GR types based on the nature of the entailments in the development data, but the system could easily be reconfigured to focus on other relation types. Finally, we required grs(H) $\cap$ grs(T) to be non-empty (no vacuous positives), but did not restrict this criterion to subjects and objects.

We used a PTB tokenizer[1] for consistency with the parser's training data. We used the morpha lemmatizer (Minnen et al., 2000), which is built into the C&C tools, to match tokens across T and H; and we converted all tokens to lowercase. If the parser failed to find a spanning analysis for either T or H, the entailment decision was NO. The full pipeline is shown in Figure 1.

3 Results

A total of 19 systems were submitted. The baseline score for "always YES" was 51.8% accuracy. Our system achieved 72.4% accuracy, which was the highest score among the submitted systems. Table 1 shows the results for our system, as well as SCHWA (University of Sydney), also based on the C&C parser and the next-highest scorer (see Section 6 for a comparison), and the median and lowest scores. The parser found an analysis for 99.0% of T sentences and 99.7% of H sentences in the test data.

[1] `http://www.cis.upenn.edu/~treebank/tokenizer.sed`.

Tokenize T and H with PTB tokenizer
⇓
Parse T and H with C&C parser
⇓
Lowercase and lemmatize all tokens
⇓
Discard any GR in grs(H) containing a token not in T
⇓
YES if core(H) $\subseteq$ core(T) and grs(H) $\cap$ grs(T) $\neq \emptyset$,
NO otherwise

Figure 1: Full pipeline for parser and entailment system. core(S): the set of core (subject and object) GRs in grs(S).

4 Error Analysis

Table 2 shows the results for our system on the development data (66 sentences). The parser found an analysis for 100% of sentences and the overall accuracy was 66.7%. In the majority of cases the parser and entailment system worked together to find the correct answer as expected. For example, for *Trading in AMR shares was suspended shortly after 3 p.m. EDT Friday and didn't resume.* ⇒ *Trading didn't resume.*, the parser produced three GRs for H (tokens are shown lemmatized and lowercase): `(nsubj resume trading)`, `(neg do n't)`, and `(aux resume do)`. All of these were also in grs(T), and the correct YES decision was made. For *Moreland sat brooding for a full minute, during which I made each of us a new drink.* ⇒ *Minute is made.*, the parser produced two GRs for H. One, `(auxpass make be)`, was ignored because the passive auxiliary *be* is not in T. The second, passive subject GR `(nsubjpass make minute)` was equated with a direct object `(dobj make minute)`. This GR was not in grs(T), so the correct NO decision was made.

In some cases a correct YES answer was reached via arguably insufficient positive evidence. For *He would wake up in the middle of the night and fret about it.* ⇒ *He would wake up.*, the parser produces incorrect analyses for the VP *would wake up* for both T and H. However, these GRs are ignored since they are non-core (not subject or object), and a YES decision is based on the single GR match `(nsubj would he)`. This

System	Score on YES entailments			Score on NO entailments			Overall
	correct	incorrect	accuracy (%)	correct	incorrect	accuracy (%)	accuracy (%)
Cambridge	98	58	62.8	120	25	82.8	72.4
SCHWA	125	31	80.1	87	58	60.0	70.4
Median	71	85	45.5	88	57	60.7	52.8
Low	68	88	43.6	76	69	52.4	47.8

Table 1: Results on the test data.

System	Score on YES entailments			Score on NO entailments			Overall
	correct	incorrect	accuracy (%)	correct	incorrect	accuracy (%)	accuracy (%)
Cambridge	22	16	57.9	22	6	78.6	66.7

Table 2: Results on the development data.

Type	FN	FP	Total
Unbounded dependency	8	1	9
Other parser error	6	2	8
Entailment system	1	3	4
Difficult entailment	1	0	1
Total	16	6	22

Table 3: Error breakdown on the development data. FN: false negative, FP: false positive.

is not entirely a lucky guess, since the entailment system has correctly ignored the odd analyses of *would wake up* and focused on the role of *he* as the subject of the sentence. However, especially since the target content word pair was *(he, wake)*, more positive evidence would be desirable. Of the 22 correct YES decisions, only two were truly lucky guesses in that the single match was a determiner; others had at least one core match.

Table 3 shows the breakdown of errors. The largest category was false negatives due to unbounded dependencies not recovered by the parser, for example *It required an energy he no longer possessed to be satirical about his father.* ⇒ *Somebody no longer possessed the energy.*. Here the parser fails to recover the direct object relation between *possess* and *energy* in T. It is known that parsers have difficulty with unbounded dependencies (Rimell et al., 2009, from which the unbounded examples in this dataset were obtained), so this result is not surprising.

The next category was other parser errors. This is a miscellaneous category including e.g. errors on coordination, parenthetical elements, identifying the head of a clausal subject, and one due to the POS tagger. For example, for *Then at least he would have a place to hang his tools and something to work on.* ⇒ *He would have something to work on.*, the parser incorrectly coordinated *tools* and *something* for T. As a result `(dobj have something)` was in grs(H) but not grs(T), yielding an incorrect NO.

Four errors were due to the entailment system rather than the parser; these will be dicsussed in Section 5. We also identified one sentence where the gold standard entailment appears to rely on extra-syntactic information, or at least information that is difficult for a parser to recover. This is *Index-arbitrage trading is "something we want to watch closely," an official at London's Stock Exchange said.* ⇒ *We want to watch index-arbitrage trading.* Recovering the entailment would require resolving the reference of *something*, arguably the role of a semantic rather than syntactic module.

5 Entailment System Evaluation

We now consider whether our entailment system was an appropriate tool for evaluating the C&C parser on the PETE dataset. It is easy to imagine a poor entailment system that makes incorrect guesses in spite of good parser output, or conversely one that uses additional reasoning to supplement the parser's analysis. To be an appropriate *parser evaluation tool*, the entailment system must decide whether the information in H is also contained in the parse of T, without "introducing" or "correcting" any errors.

Assuming our GR-based approach is valid, then given gold-standard GRs for T and H, we expect an appropriate entailment system to result in 100% accuracy on the task evaluation. To perform this oracle experiment we annotated the development

data with gold-standard GRs. Using our entailment system with the gold GRs we achieved 90.9% task accuracy. Six incorrect entailment decisions were made, of which one was on the arguably extra-syntactic entailment discussed in Section 4.

Three errors were due to transformations between T and H which changed the GR label or head. For example, consider *Occasionally, the children find steamed, whole-wheat grains for cereal which they call "buckshot". ⇒ Grains are steamed.*. In T, *steamed* is a prenominal adjective, with *grains* as its head; while in H, it is a passive, with *grains* as its subject. The entailment system did not account for this transformation, although in principle it could have. The other two errors occurred because GRs involving a non-core relation or a pronoun introduced in H, both of which our system ignored, were crucial for the correct entailment decision.

Table 3 shows that with automatically-generated GRs, four errors on the task evaluation were attributable to the entailment system. Three of these were also found in the oracle experiment. The fourth resulted from a POS change between T and H for *There was the revolution in Tibet which we pretended did not exist. ⇒ The pretended did not exist.*. The crucial GR was `(nsubj exist pretended)` in grs(H), but the entailment system ignored it because the lemmatizer did not give *pretend* as the lemma for *pretended* as a noun. This type of error might be prevented by answering NO if the POS of any word changes between T and H, but the implementation is non-trivial since word indices may also change. There were eight POS changes in the development data, most of which did not result in errors. We also observed two cases where the entailment system "corrected" parser errors, yielding a correct entailment decision despite the parser's incorrect analysis of T. When compared with a manual analysis of whether T entailed H based on automatically-generated GRs, the entailment system achieved 89.4% overall accuracy.

6 Conclusion

We achieved a successful result on the PETE task using a state-of-the-art parser and a simple entailment system, which tested syntactic entailments by comparing the GRs produced by the parser for T and H. We also showed that our entailment system had accuracy of approximately 90% as a tool for evaluating the C&C parser (or potentially any parser producing GR-style output) on the PETE development data. This latter result is perhaps even more important than the task score since it suggests that PETE is worth pursuing as a viable approach to parser evaluation.

The second-highest scoring system, SCHWA (University of Sydney), was also based on the C&C parser and used a similar approach (though using CCG dependency output rather than GRs). It achieved almost identical task accuracy to the Cambridge system, but interestingly with higher accuracy on YES entailments, while our system was more accurate on NO entailments (Table 1). We attribute this difference to the decision criteria: both systems required at least one matching relation between T and H for a YES answer; but we additionally answered NO if any core GR in grs(H) was not in grs(T). This difference shows that a GR-based entailment system can be tuned to favour precision or recall.

Finally, we note that although this was a simple entailment system with some dataset-specific characteristics – such as a focus on subject and object relations rather than, say, PP-attachment – these aspects should be amenable to customization or generalization for other related tasks.

Acknowledgments

The authors were supported by EPSRC grant EP/E035698/1. We thank Matthew Honnibal for his help in producing the gold-standard GRs.

References

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, Genoa, Italy.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of INLG*, Mitzpe Ramon, Israel.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of EMNLP*, Singapore.

Deniz Yuret, Aydın Han, and Zehra Turgut. 2010. Semeval-2010 task 12: Parser evaluation using textual entailments. In *Proceedings of SemEval-2010*, Uppsala, Sweden.