Semantic Parsing for Textual Entailment

Elisabeth Lien
Department of Informatics
University of Oslo, Norway
elien@ifi.uio.no

Milen Kouylekov
Department of Informatics
University of Oslo, Norway
milen@ifi.uio.no

Abstract

In this paper we gauge the utility of general-purpose, open-domain semantic parsing for textual entailment recognition by combining graph-structured meaning representations with semantic technologies and formal reasoning tools. Our approach achieves high precision, and in two case studies we show that when reasoning over n-best analyses from the parser the performance of our system reaches state-of-the-art for rule-based textual entailment systems.

1 Background and Motivation

There is a growing interest in recent years in general-purpose semantic parsing into graph-based meaning representations, which provide greater expressive power than tree-based structures. Recent efforts in this spirit include, for example, Abstract Meaning Representation (Banarescu et al., 2013), and Semantic Dependency Parsing (SDP) (Oepen et al., 2014; Oepen et al., 2015). Simultaneously, in the Semantic Web community, a range of generic semantic technologies for storing and processing graph-structured data has been made available, but these have not been much used for natural language processing tasks. We propose a flexible, generic framework for precision-oriented Textual Entailment (TE) recognition that combines semantic parsing, graph-based representations of sentence meaning, and semantic technologies.

During the decade since the TE task was defined, (logical) inference-based approaches have made some important contributions to the field. Systems such as Bos and Markert (2006) and Tatu and Moldovan (2006) employ automated proof search over logical representations of the input sentences. Other systems, such as Bar-Haim et al. (2007), apply transformational rules to linguistic representations of the sentence pairs, and determine entailment through graph subsumption. Because inference-based systems are vulnerable to incomplete knowledge in the rule set and errors in the mapping from natural language sentences to logical forms or linguistics representations, and because the definition of the TE task encourages a more relaxed, non-logical notion of entailment, the majority of TE systems have used more robust approaches, however. Our work supports a notion of logical inference for TE by reasoning with formal rules over graph-structured meaning representations, while achieving results that are comparable with robust approaches.

We use a freely available, grammar-driven semantic parser and a well-defined reduction of underspecified logical-form meaning representations into variable-free semantic graphs called Elementary Dependency Structures (EDS) (Oepen and Lønning, 2006). We capitalize on a pre-existing storage and search infrastructure for EDSs using generic semantic technologies. For entailment classification, we create inference rules that enrich the EDS graphs, apply the rules with a generic reasoner, and use graph alignment as a decision tool.

To test our generic setup, we perform two case studies where we replicate well-performing TE systems, one from the Parser Evaluation using Textual Entailments (PETE) task (Yuret et al., 2010), and one from SemEval 2014 Task 1 (Marelli et al., 2014). The best published results for the PETE task, Lien (2014), were obtained through heuristic rules that align meaning representations based on structural similarity. Lien and Kouylekov (2014) extend the same basic approach for SemEval 2014 by including lexical relations and negation handling. We recast the handwritten heuristic rules from these systems as formal Semantic Web Rule Language (SWRL) rules, and run them with a generic reasoning tool over EDS

Proceedings of the 14th International Conference on Parsing Technologies, pages 40–49,
Bilbao, Spain; July 22–24, 2015.

meaning representations. The PETE contribution of Lien (2014) experimented with using n-best analyses from the parser to boost TE recall, and we can easily include n-best reasoning in our setup.

In Sections 2 and 3, we outline our approach and describe the semantic parsing setup and semantic technologies we employ. Sections 4 and 5 detail our replication of the two TE shared tasks. Finally, in Section 6, we sum up our effort and point to directions for future work.

2 General-purpose Semantic Parsing

General-purpose, open-domain semantic parsing systems that output logical-form meaning representations are freely available today, but have not yet been widely used in TE systems. For our replication of the PETE and SemEval tasks, we use the English Resource Grammar (ERG) (Flickinger, 2000), a broad-coverage HPSG-based parser. The ERG has been continuously developed since around 1993, and today will typically allow parsing of 90-95% of the sentences in naturally occuring running texts of various domains and genres at average parse times of a couple of seconds per sentence. The ERG includes a Maximum Entropy parse ranking model that is trained on some 50,000 mixed-domain sentences; the parser applies exact inference, i.e., constructs a complete parse forest and facilitates extraction of n-best lists of analyses in globally optimal rank order. In our experiments, we use the ERG in its 1212 release version, together with its standard PET parser (Callmeier, 2002), and off-the-shelf models and settings. The ERG outputs underspecified meaning representations in the Minimal Recursion Semantics (MRS) framework (Copestake et al., 2005). The MRS logical-form meaning representations can be converted to EDSs, which are variable-free semantic dependency graphs. Kouylekov and Oepen (2014) recently showed that the Resource Description Framework (RDF) is suitable for representing various types of semantic graphs, and demonstrated how to embed EDS meaning representations in RDF. We opt for EDS over MRS because its variable-free form integrates more naturally with RDF technologies, while still retaining the semantic information essential to entailment recognition.

In the EDS example in Figure 1, each line depicts a graph node (each corresponding to one elementary predication in the original MRS), with node identifiers prefixed to the node labels (separated by the colon), and a set of outgoing arcs (role-argument pairs) enclosed in parentheses. The semantic arguments to the relation representend by the node are directed arcs to other nodes in the EDS graph. For instance, the node for `_would_v_modal` is connected to the node for `_and_c` through an arc labeled `ARG1`. The node labeled `_and_c` in turn has outgoing arcs to `_wake_v_up` and `_fret_v_about`. The two `pron` nodes do not have outgoing arcs, they are connected to the structure through incoming arcs from the verb nodes. Finally, each of the `pronoun_q` nodes is connected to a `pron` node through a `BV` ("bound variable") arc. A graphical visualization of the same graph is shown in Figure 3 (ignoring nodes and arcs shown in green there, which are added by our entailment processor).

There are two notable examples of logic-based TE systems that have used the ERG parser and MRS meaning representations: Wotzlaw and Coote (2013) present a TE system which combines the results of deep and shallow linguistic analyses into scope-resolved MRS representations. The MRS expressions are translated into another, semantically equivalent first-order logic format, which, enriched with background knowledge, is used for the actual inference. The system of Bergmair (2010) also uses MRS as an intermediate format in constructing meaning representations. Input sentences are parsed with the ERG, and the resulting MRSs are translated into logical formulae that can be prosessed by an inference engine. In contrast to these prior applications of generic semantic parsing using the ERG to the TE task, our work simplifies the scopally underspecified logical forms of MRS into more compact graph-structured representations of core predicate–argument relations, and we define TE-specialized notions of inference over these semantic graphs.

3 Semantic Technologies and Textual Entailment

Kouylekov and Oepen (2014) map different types of meaning representations, including the EDSs used in our work, to RDF graphs, stored in off-the-shelf RDF triple stores, and searched using SPARQL queries. In our work, we build a TE system that utilizes their infrastructure as a basis for reasoning over EDS graphs.

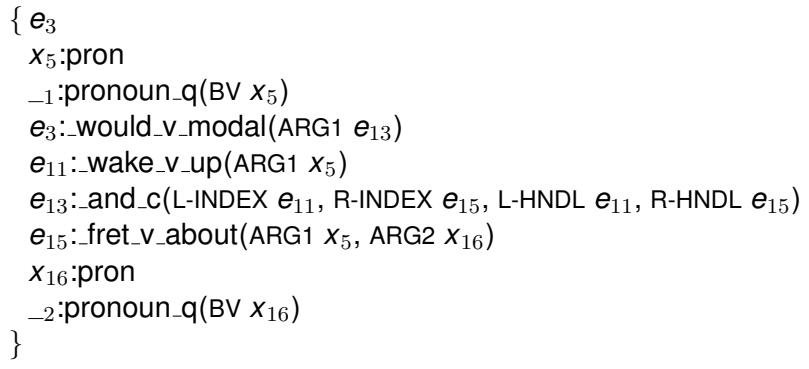

Figure 1: EDS for *He would wake up [...] and fret about it.* (PETE id 5019).

Textual Entailment was defined by Dagan et al. (2006) as the task of recognizing whether, given two text fragments, the meaning of one text entails the meaning of the other text. The text fragments are conventionally referred to as the *text T* and the *hypothesis H*, respectively. The notion of "entailment" used in TE is informal and based at least in part on general human knowledge of language and the world.

Our textual entailment system uses graph alignment over EDS structures as the basis for entailment decisions. We extend the approach by enriching the graphs in a forward-chaining spirit using SWRL rules, and the Jena reasoner[1]. After the reasoning step, the actual alignment is performed with a SPARQL query that tries to match the hypothesis graph to the text graph. Along with a classification decision, the system outputs a "proof" by listing every SWRL rule that was used in the reasoning. In a sense, we are following the classical reasoning approach of trying to infer the hypothesis from the text.

3.1 SWRL

Our subsumption approach to entailment recognition requires some rewriting of the EDS graphs produced by the ERG parser. For example, the EDS graph in Figure 1 needs to be rewritten so that dependencies are propagated into the coordinate structure, which will facilitate the subsumption of subgraphs. We use SWRL, a semantic web standard for reasoning over ontologies[2], to encode rewriting rules for EDS graphs. The graph structures are enriched with a set of forward-chaining SWRL rules, and, thus, our graph-rewriting approach can be seen as a form of forward-chaining inference.

The system uses two sets of SWRL rules, one for the text and one for the hypothesis graph. The function of these rules is to further normalize and to add information to both graphs in order to make matching possible. We adapt the rule sets for different data sets to accomodate variation in entailment phenomena. The rule sets contain five types of rules:

- abstraction rules
- predicate simplification rules
- structural rules
- lexical relation rules
- polarity marking rules

Abstraction Rules We employ a number of abstraction rules to allow matching of indefinite and personal *pronouns* in the *H* graph to NPs in the *T* graph. To be able to match the indefinite pronoun *somebody* to the personal pronoun *he* in e.g. *He has a point he wants to make [...]* ⇒ *Somebody wants to make a point* (PETE id 1026), the rules label both pronouns with the same abstraction label, i.e., they add an additional `rdf:type` property to these nodes, which can be used in subsequent testing for node equivalence.

Our rules also abstract over certain *quantifiers*. In the data sets we have examined, the text and hypothesis sentence of an entailment pair often have quantifier variations that are clearly not relevant for recognizing the entailment relationship (e.g., *A woman is cleaning a shrimp* ⇒ *The woman is cleaning a shrimp*, SemEval id 3364). We group these quantifiers into candidate equivalence classes using rules of the form:

[1] `https://jena.apache.org/`
[2] `http://www.w3.org/Submission/SWRL/`

```
[(?a eds:predicate "_a_q") ->
(?a rdf:type eds:equiv_quant)]

[(?a eds:predicate "_the_q") ->
(?a rdf:type eds:equiv_quant)]
```

These rules state that if a node `?a` is labeled with a certain quantifier predicate (`_a_q` or `_the_q`, in this specific example), then the node `?a` is of type `equiv_quant`. This fact is added to the EDS graph, which allows matching of the node with other nodes that have the same type.

Simplified Predicates ERG lexical predicate symbols conjoin information about the lemma, part-of-speech, and sense of the wordform. To increase the robustness of the matching, we add a simplified predicate symbol which contains only the lemma and part-of-speech. This makes matching possible in cases where the ERG has given different predicate symbol interpretations of the same word in text and hypothesis. For instance, `_trade_v_in` and `_trade_v_1` are associated with different usages of the verb *trade*, and for our purposes can be simplified to `_trade_v`.

Structural Rules Certain rules enrich the graph structure without adding new meaning content to the graph. By adding arcs to certain constructions in the text graph, we make matching possible for cases where the hypothesis graph contains a substructure of the text construction. For instance, to make matching possible for the text *He would wake up [...] and fret about it* and the hypothesis *He would wake up* (PETE id 5019), we need to draw additional arcs from the node `_would_v_modal` to its indirect arguments `_wake_v_up` and `_fret_v_about`, i.e., the arguments of the conjunction node `_and_c`. This is done by applying the rules in Figure 2. The first two rules label all modal verb nodes as having type `modal_verb`, and coordinating nodes as being of type `coordination`. The third rule states that if a node is of type `modal_verb`, and it has an `ARG1` arc to a node of type `coordination`, then we add `ARG1` arcs to each of the argument nodes of the coordination. When applied to the EDS in Figure 1, the rules yield the structure shown in Figure 3, where the new arcs are marked in green.

Additional rules for **lexical relations** and **polarity marking** are described in Sections 3.3 and 3.4, respectively.

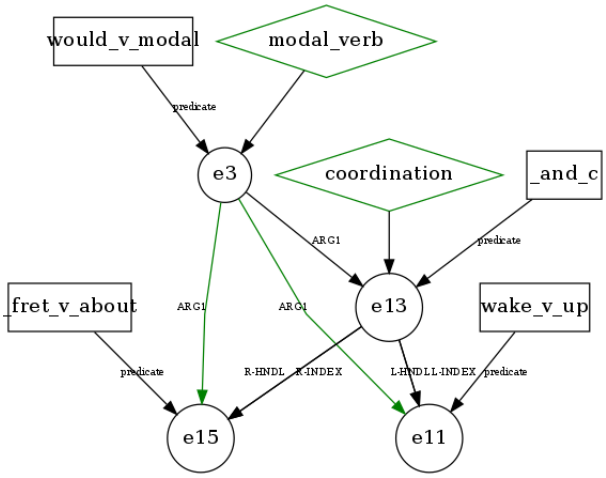

Figure 3: Additional `ARG1` arcs (in green) have been added to directly connect the modal verb (node `e3`) to its subarguments (nodes `e11` and `e15`).

Removing Graph Components To make matching possible, we also need an additional set of rules that remove certain predicates, nodes, and arcs from the hypothesis before the subsumption algorithm is applied. For instance, the sentences *A boy is playing* and *There is a boy playing* have the same meaning content, but receive different analyses from the ERG, where the existential assertion, including its tense and aspect, is reified as a separate relation. Removing the subgraph corresponding to *there is* makes matching possible. Removing graph components is not part of the general SWRL specification, but it is an extension to the rule language provided by the Jena reasoner. To ensure that the bulk of our entailment rules remain SWRL compatible, we keep the removal rules separated.

3.2 Subsumption Algorithm

Our reasoning-based system (**RBS**) processes an entailment pair using an algorithm that has the following steps:

- The text T and hypothesis H are analyzed with the ERG parser.
- The EDSs for T and H are converted into RDF triples.
- H is enriched using the SWRL rules and converted into a SPARQL query $query_h$ in which the query statements are the conjunc-

```
[modal_verb_type: (?a eds:predicate ?p), regex(?p, "^.+modal$")
-> (?a rdf:type eds:modal_verb)]

[coordination_type: (?a eds:predicate ?p), regex(?p, "^.+_c_?.*$")
-> (?a rdf:type eds:coordination)]

[(?a rdf:type eds:modal_verb),  (?b rdf:type eds:coordination),
 (?a eds:arg1 ?b), (?b eds:l-index ?c), (?b eds:r-index ?d)
-> (?a eds:arg1 ?c), (?a eds:arg1 ?d)]
```

Figure 2: SWRL rule for making explicit (inserting) arcs from a modal verb to its indirect arguments

```
_1:_a_q[BV x6]
x6:_man_n_1[]
e3:_walk_v_1[ARG1 x6]

select x1 x2 x3 where{
x1 eds:predicate "_a_q" .
x2 eds:predicate "_man_n_1" .
x3 eds:predicate "_walk_v_1" .
x1 eds:BV x2 .
x3 eds:ARG1 x2
}
```

Figure 4: Converting an EDS structure into a SPARQL query

tion of all of the triples in the RDF representation of *H*.

- The RDF triples of *T* and the SWRL rules for expanding *T* are given as the input to the reasoner.
- If the $query_h$ is matched into the inferred model for *T*, the entailment relation is assigned to the pair.

The algorithm defines textual entailment as a subsumption problem. *T* entails *H* if the (enriched) RDF graph that represents *T* contains the entire graph of *H*.

Converting *H* into a SPARQL query allows us to use the standard RDF technology to perform the graph subsumption. In Figure 4, we see an example of how the EDS for the sentence *A man walks* is converted into a SPARQL query. Using SPARQL automatically computes (and makes available) the correspondence between the predications of *T* and *H*.

Using the RDF reasoner allows us to understand the reason *H* was subsumed by *T* as the Jena reasoner outputs a verbose log on each inference step taken to obtain a specific triple in the inferred model. In the log output example below, the predication $x5$ was recognized to be an indefinite pronoun because it has the predicate `person`, and is the target of a `BV` (bound variable) relation from a predication with the predicate `_some_q`:

```
Added statement
  [x5, indef_pronoun, "true"]
Used rule
  [Rule someone-body_is_indef_pron
    concluded
    (x5 indef_pronoun 'true')
  <-
  Fact (x5 predicate 'person')
  Fact (_1 predicate '_some_q')
  Fact (_1 bv x5)]
```

3.3 Lexical Relations

In our reasoning-based system we have integrated lexical entailment rules extracted from WordNet (Fellbaum, 1998) as proposed in Lien and Kouylekov (2014). For each predication in *T* we dynamically create SWRL rules that expand the RDF graph of *T* by adding new predications for words that are synonyms or hypernyms of the original predication. For example, for the predication `_assistant_n` we expand the *T* graph with the predications `_worker_n` and `_person_n`. Figure 5 shows a simplified version of these rules.

The creation of these rules is done once before the start of the inference. The system queries WordNet for rules that can be used until no rules can be added. If the SWRL rules add predicates after the reasoning step that can be expanded using rules deducted from WordNet then these rules are added to the reasoner and the reasoning is restarted. We used this strategy as we were not able to encode the entire WordNet database as rules in

```
[to-sense-rule: (x eds:predicate "_assistant_n_1")
       -> (x eds:wordnet "assistant_n_1") ]

 [hypernym-rule: (x eds:wordnet "assistant_n_1")
        -> (x eds:wordnet "worker_n_1")]

  [hypernym-rule: (x eds:wordnet "worker_n_1")
        -> (x eds:wordnet "person_n_1")]

[to-predicate-rule: (x eds:wordnet "person_n_1")
       -> (x eds:predicate "_person_n_1")]
```

Figure 5: Automatically generated WordNet rules.

the Jena reasoner.

3.4 Contradiction

The SemEval 2014 task uses a three-way classification of the entailment pairs. Systems were required to assign to each pair one of the three categories ENTAILMENT, CONTRADICTION, or NEUTRAL. To handle three-way classification, we have developed a special rule-based contradiction module. Although the SemEval data display various contradiction phenomena, we focus on negation, which is the most frequent contradiction indicator.

For classification of pairs where event negation or instance negation in one of the sentences creates contradiction, we combine *polarity marking* of nodes with graph matching. The nodes that are in the immediate scope of the negation are marked as negative, and all other nodes as positive. For instance, in the most simple case of event negation, the predicate `neg` negates some event node via an `ARG1` arc (e.g., *not singing*). The following rule marks both the node of the `neg` predicate, and the event node as negative:

```
[(?a eds:predicate "neg"),
 (?a eds:arg1 ?b) ->
 (?a eds:polarity "negative"),
 (?b eds:polarity "negative")]
```

In the parallell case for simple instance negation (e.g., *no woman*), the node of the `_no_q` predicate and its "bound variable", the instance node, are both labeled as negative:

```
[(?q eds:predicate "_no_q"),
 (?q eds:bv ?a) ->
 (?q eds:polarity "negative"),
 (?a eds:polarity "negative")]
```

Since both events and instances can be complex linguistic constructions, our rule set contains rules that handle negation of e.g., compounds, nominalizations, coordination, and nesting of verbs. Broadly speaking, these rules are similar in spirit to the "MRS crawling" process defined by Packard et al. (2014) for the task of negation scope resolution.

In the classification process, we run the system twice on each entailment pair: in the first run the polarity markings are ignored, and in the second run they are considered. If the system finds a subsumption of H in the T graph *without* polarity markings, but no subsumption *with* polarity markings, then the pair is classified as CONTRADICTION.

Polarity marking allows us to use the same structures for both entailment and contradiction testing. Our polarity marking approach is parallell to how negation is represented in AMR[3].

3.5 N-best Matching

The ERG parser can output a ranked list of candidate analyses for a sentence. We extended our system with n-best matching to facilitate entailment recognition when the top-ranked analysis does not correspond to the perceived meaning of the sentence, i.e., to reduce the impact of errors in parse ranking. Such errors include prepositional phrase attachments, noun compounds, coordinate structures, and other interpretation variants. For example in the sentence:

who invented the light bulb?

[3] https://github.com/amrisi/amr-guidelines/blob/master/amr.md#negation

the parser creates two valid (in principle, if not equally likely) analyses based on the semantic interpretation of the word *light* as 1) an adjective; 2) part of a noun–noun compound. If the same phrase occurs in *T* and *H*, but their contexts are different, the top-ranked analyses from the parser ranker for *T* and *H* may contain different interpretations of the phrase. Our default assumption is that such misalignment is the cause of many unwarranted mismatches between the *T* and *H* graphs.

For each entailment pair i ($pair_i$) we iterate over all analyses of *T* and *H*. If the n-th analysis of *T* entails the k-th analysis of *H* we assign the ENTAILMENT relation to the entailment pair. This definition is valid as each analysis of *T* and *H* corresponds to a valid interpretation.

To determine the number of analyses for *T* and *H* we need to consider [4] we have employed an optimization strategy. We have gradually increased the number of considered analyses of *T* and *H*, and measured the system performance on the training set. The best n-m combination, where n are the analyses considered for *T* and m are the analyses considered for *H*, is used on the test set.

4 First Case Study: PETE

In our first case study, we recast the Lien (2014) heuristic for the PETE shared task data as SWRL rules. The objective of the PETE task was to propose an alternative method for parser evaluation: instead of comparing parser output to gold annotated treebank data, parsers can be evaluated indirectly by examining how well the parser output supports the task of entailment recognition. The data provided for the task was constructed so that syntactic analysis of the sentence pairs would be sufficient to determine whether the text entails the hypothesis. The PETE development and test sets contain 66 and 301 sentence pairs, respectively. Characteristically, the hypothesis sentence of the positive entailment pairs is shorter that the text sentence, and is a substructure of the text, frequently with some minor changes (e.g., active-to-passive conversion, a noun phrase in the text is replaced by a underspecified pronoun in the hypothesis). In the negative entailment pairs the hypothesis usually contains elements from the text that are structured differently and thus give the hypothesis a different meaning from the text.

[4] The ERG can return all the possible grammatical analyses up to a user-supplied maximum rank n.

The best scoring system in the shared task was the Cambridge system (Rimell and Clark, 2010), with an accuracy of 72.4%.

Table 1 presents our 1-best and 10-best results on the PETE test data, and compares them to the results reported by Lien (2014), and the shared task winner Rimell and Clark (2010). Our **RBS** system outperforms the system developed by Lien (2014), establishing a new state-of-the-art. The two systems have close results on both single analysis input and n-best. This demonstrates that our system correctly implements the approach proposed in Lien (2014).

The main advantage of our system is the high **precision**. The PETE data focus on entailments that can be recognized using structural analysis alone (allowing for the substitution of noun phrases with generalized pronouns), which fits nicely with our strict graph subsumption algorithm over meaning representations. When we examine the system's output for the PETE development data, we see that two-thirds of the true positives in the ENTAILMENT category concern sentence pairs where *H* is a substructure of *T*. In these cases, enriching the RDF graphs with arcs connecting predicates to their indirect arguments, and allowing noun phrases to match generalized pronouns, is sufficient for entailment recognition. In the remaining one-third of the true positives, there are syntactic differences from *T* to *H*, but the ERG abstracts from these differences and assigns the same analysis to both (the relevant substring of) *T* and *H*. For instance, the *T* noun phrase *steamed, whole-wheat grains* and the *H* sentence *Grains are steamed* (PETE id 3081.N) receive the same EDS analysis, with *grains* as the passive `ARG2` of the verb *steam*. In another example below (PETE id 2004), the relative pronoun *which* is ignored at the level of ERG semantics, which instead directly identifies *the stream* as the `ARG2` of the seeing event:

> *[...] the stream which he had seen [...].*
> ⇒ *Someone had seen the stream.*

In the cases where our system fails to recognize the entailment relationship, it is often the case that one of the sentences is assigned an incorrect analysis from the ERG parser. An incorrect assignment of an argument role, or an incorrect attachment of a prepositional phrase prevents our strict subsumption algorithm from classifying the relationship as entailment.

	RBS	RBS n-best	Lien	Lien n-best	Rimell & Clark
Accuracy	**72.1**	**77.1**	70.7	76.4	72.4
Precision	89.0	81.1	88.6	81.4	79.6
Recall	52.6	72.7	50.0	70.5	62.8
F-Measure	66.1	76.6	63.9	75.5	70.2

Table 1: Performance of our reasoning-based system on the PETE test data.

The influence of imperfect parse ranking on the system performance can be alleviated by running it on n-best parser outputs. Considering multiple analyses of *T* and *H* from the ERG parser increases the performance of our system by adding a significant boost to the recall without damaging the precision. Using 1-best analyses for *T* and *H*, our system has a performance compatible with the previously best performing system on the PETE task.

5 Second Case Study: SemEval 2014 Task 1

RBS	RBS n-best	UIO-Lien	Illinois-LH
77.4	80.4	77.1	84.6

Table 2: Comparison of accuracy of RBS on the SemEval test data.

	Precision	Recall	F-Measure
Contradiction	95.9	66.1	78.3
Entailment	95.6	52.4	67.7

Table 3: Precision, recall and F-measure of RBS n-best on the SemEval test data.

In our second case study, we revisit our contribution to the SemEval 2014 task 1. The focus of this task was evaluation of compositional distributional semantic models through entailment decision (and semantic relatedness) on sentence pairs, in order to remedy the lack of benchmarks for such models. The 10,000 sentence pair data set released for the task (50% training, 50% test) reflects this goal by targeting phenomena that compositional distributional semantic models are meant to account for, e.g., lexical variation phenomena such as contextual synonymy, active-passive and other syntactic alternation, negation, and operator scope. The data do not require encyclopedic knowledge about instances of concepts, only generic semantic knowledge about general concept categories. Unlike in the PETE data set, the text and hypothesis sentences are usually similar in length, and either paraphrase or contradict each other, or are more or less unrelated in meaning.

In the entailment subtask, systems were required to assign one of the categories ENTAILMENT, CONTRADICTION, or NEUTRAL to each sentence pair. The best scoring system was the Illinois-LH system (Lai and Hockenmaier, 2014), with an accuracy of 84.6%.

Table 2 presents our 1-best and 10-best results on the SemEval test data, and compares them to the results for the UIO-Lien system (Lien and Kouylekov, 2014) and the shared task winner Illinois-LH.

The results obtained on the SemEval data set are encouraging. As with the PETE data set we have improved over the results we achieved with the UIO-Lien system. This demonstrates the adaptability of our approach to new data sets. When we participated in the SemEval task with the UIO-Lien system, we did not submit a run using the n-best analyses from the ERG parser, so we are not able to make a comparison for n-best results. Our current n-best RBS system obtains a high accuracy which makes it the 6th ranked system on the SemEval data. With this result it is the top ranked unsupervised rule based system.

It is worth noticing that our system achieves a similar result as another task participant, Bestgen (2014), which employs a similarity-based algorithm and latents semantic analysis to recognize entailment. Our advantage versus such approaches is that we are able to create a reasoning chain that motivates the system decision instead of presenting a simple similarity number. Still in the future development of our system we can investigate the possibility of using probabilistic rules to guide our reasoner.

Similar to the PETE dataset results our system obtained a high precision (more than 95.0% precision on both ENTAILMENT and CONTRADIC-

TION), maintaining a decent recall as shown in Table 3.

The SemEval data display more variation in entailment phenomena than the PETE data, and require the use of external knowledge sources. We use WordNet to generate lexical inference rules. This allows us to capture the same types of "syntactic" entailments as in the PETE data, augmented with synonymy and hypernymy relations between predicates in T and H, as examplified by the following entailment pair (SemEval id 4176):

An eggplant is being sliced by a woman
$\Rightarrow$ *A woman is cutting a vegetable*

We did not focus on capturing entailment phenomena that were aimed specifically at evaluation of compositional distributional semantic models, and that require contextual information or equating structurally diverse phrases. In many cases, it would require formulating specific rules that would do little to improve the coverage of our system.

The system's high precision on ENTAILMENT shows that the graph subsumption of semantic structures is a reliable indicator of the entailment relation. To further improve recall, the system must incorporate more sources of knowledge and semantic variation.

6 Conclusions and Future Work

In this paper we have described an approach to TE which leans heavily on generic semantic parsing technologies, combining the off-the-shelf ERG parser with formats and tools developed for the Semantic Web and a custom-built notion of inference over graph-structured meaning representations. We have replicated our two previous TE shared task contributions, and using n-best analyses reached state-of-the-art for rule-based TE systems. These results demonstrate the utility of general-purpose, off-the-shelf semantic parsing systems for textual entailment, in particular when reasoning over ranked n-best lists can be applied to compensate for parse ranking limitations. Our system architecture rests on a comparatively small number of reasonably generic rules, i.e., there is very little task-specific engineering and tuning in our approach (as a large part of the work in done in the parser). Our 95 percent precision results demonstrate that subsumption of semantic representations is a strong indication for textual entailment. Our work contributes to moving the TE field towards logical reasoning.

One of the main strength of the system is its versatility. We reduce the amount of task-specific engineering by using generic off-the-shelf tools.

Future Work Our approach is useful for precision-critical applications like information retrieval and particularly Question Answering. In future work we plan to combine it with a shallow information retrieval approach and use its evaluation power to pick the correct answer. The system also provides a detailed account of the reasoning behind each entailment decision. This strength can be used in an answer presentation module which motivates why the system has chosen a particular answer.

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, page 178 – 186, Sofia, Bulgaria, August.

Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch. 2007. Semantic inference and the lexical-syntactic level. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 871–876.

Richard Bergmair. 2010. *Monte Carlo Semantics: Robust Inference and Logical Pattern Processing with Natural Language Text*. Ph.D. thesis, University of Cambridge.

Yves Bestgen. 2014. CECL: a new baseline and a noncompositional approach for the sick benchmark. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Johan Bos and Katja Markert. 2006. When logical inference helps determining textual entailment (and when it doesn't). In Bernardo Magnini and Ido Dagan, editors, *The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Challenges Workshop*, pages 98–103, Venice, Italy.

Ulrich Callmeier. 2002. Preprocessing and encoding techniques in PET. In Stephan Oepen, Daniel Flickinger, J. Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, page 127 – 140. CSLI Publications, Stanford, CA.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, 3(4):281 – 332.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, page 177 – 190. Springer Berlin Heidelberg.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1):15 – 28.

Milen Kouylekov and Stephan Oepen. 2014. RDF Triple Stores and a Custom SPARQL Front-End for Indexing and Searching (Very) Large Semantic Networks. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, August 23-29, 2014, Dublin, Ireland*, pages 90–94.

Alice Lai and Julia Hockenmaier. 2014. Illinois-LH: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Elisabeth Lien and Milen Kouylekov. 2014. UIO-Lien: Entailment recognition using minimal recursion semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 699–703, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Elisabeth Lien. 2014. Using minimal recursion semantics for entailment recognition. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 76–84, Gothenburg, Sweden, April. Association for Computational Linguistics.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, page 1250 – 1255, Genoa, Italy.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8. Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, Dublin, Ireland.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado, June. Association for Computational Linguistics.

Woodley Packard, Emily M. Bender, Jonathon Read, Stephan Oepen, and Rebecca Dridan. 2014. Simple negation scope resolution through deep parsing: A semantic solution to a semantic problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–78, Baltimore, Maryland, June. Association for Computational Linguistics.

Laura Rimell and Stephen Clark. 2010. Cambridge: Parser Evaluation using Textual Entailment by Grammatical Relation Comparison. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*.

Marta Tatu and Dan Moldovan. 2006. A logic-based semantic approach to recognizing textual entailment. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 819–826, Sydney, Australia, July. Association for Computational Linguistics.

Andreas Wotzlaw and Ravi Coote. 2013. A Logic-based Approach for Recognizing Textual Entailment Supported by Ontological Background Knowledge. *CoRR*, abs/1310.4938.

Deniz Yuret, Aydin Han, and Zehra Turgut. 2010. SemEval-2010 Task 12: Parser Evaluation using Textual Entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.