Evaluating Scoped Meaning Representations

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, Johan Bos
CLCG, University of Groningen
{r.i.k.van.noord, l.abzianidze, hessel.haagsma, johan.bos}@rug.nl

Abstract

Semantic parsing offers many opportunities to improve natural language understanding. We present a semantically annotated parallel corpus for English, German, Italian, and Dutch where sentences are aligned with scoped meaning representations in order to capture the semantics of negation, modals, quantification, and presupposition triggers. The semantic formalism is based on Discourse Representation Theory, but concepts are represented by WordNet synsets and thematic roles by VerbNet relations. Translating scoped meaning representations to sets of clauses enables us to compare them for the purpose of semantic parser evaluation and checking translations. This is done by computing precision and recall on matching clauses, in a similar way as is done for Abstract Meaning Representations. We show that our matching tool for evaluating scoped meaning representations is both accurate and efficient. Applying this matching tool to three baseline semantic parsers yields F-scores between 43% and 54%. A pilot study is performed to automatically find changes in meaning by comparing meaning representations of translations. This comparison turns out to be an additional way of (i) finding annotation mistakes and (ii) finding instances where our semantic analysis needs to be improved.

Keywords: parallel corpus, semantic annotation, discourse representation structure, evaluation, semantic scope

1 Introduction

Semantic parsing is the task of assigning meaning representations to natural language expressions. Informally speaking, a meaning representation describes who did what to whom, when, and where, and to what extent this is the case or not. The availability of open-domain, wide coverage semantic parsers has the potential to add new functionality, such as detecting contradictions, verifying translations, and getting more accurate search results. Current research on open-domain semantic parsing focuses on supervised learning methods, using large semantically annotated corpora as training data.

However, there are not many annotated corpora available. We present a parallel corpus annotated with formal meaning representations for English, Dutch, German, and Italian, and a way to evaluate the quality of machine-generated meaning representations by comparing them to gold standard annotations. Our work shows many similarities with recent annotation and parsing efforts around Abstract Meaning Representations, (AMR; Banarescu et al., 2013) in that we abstract away from syntax, use first-order meaning representations, and use an adapted version of SMATCH (Cai and Knight, 2013) for evaluation. However, we deviate from AMR on several points: meanings are represented by scoped meaning representations (arriving at a more linguistically motivated treatment of modals, negation, presupposition, and quantification), and the non-logical symbols that we use are grounded in WordNet (concepts) and VerbNet (thematic roles), rather than PropBank (Palmer et al., 2005). We also provide a syntactic analysis in the annotated corpus, in order to derive the semantic analyses in a compositional way.

We make the following contributions:

- A meaning representation with explicit scopes that combines WordNet and VerbNet with elements of formal logic (Section 2).
- A gold standard annotated parallel corpus for formal meaning representations for four languages (Section 3).
- A tool that compares two scoped meaning representations for the purpose of evaluation (Section 4 and Section 5).

2 Scoped Meaning Representations

2.1 Discourse Representation Structures

The backbone of the meaning representations in our annotated corpus is formed by the Discourse Representation Structures (DRS) of Discourse Representation Theory (Kamp and Reyle, 1993). Our version of DRS integrates WordNet senses (Fellbaum, 1998), adopts a neo-Davidsonian analysis of events employing VerbNet roles (Bonial et al., 2011), and includes an extensive set of comparison operators. More formally, a DRS is an ordered pair of a set of variables (discourse referents) and a set of conditions. There are basic and complex conditions. Terms are either variables or constants, where the latter ones are used to account for indexicals (Bos, 2017). Basic conditions are defined as follows:

- If W is a symbol denoting a WordNet concept and x is a term, then W(x) is a basic condition;
- If V is a symbol denoting a thematic role and x and y are terms, then V(x,y) is a basic condition;
- If x and y are terms, then x=y, x≠y, x~y, x≤y, x≰y, x<y, x≤y, x≮y, and x>y are basic conditions formed with comparison operators.

WordNet concepts are represented as word.POS.SenseNum, denoting a unique synset within WordNet. Thematic roles, including the VerbNet roles, always have two arguments and start with an uppercase character. Complex conditions introduce scopes in the meaning representation. They are defined using logical operators as follows:
duced in Venhuizen (2015) and Venhuizen et al. (2018),

This is carried out by using the labels for DRSs as intro-

convert a DRS into a flat clausal form, i.e. a set of clauses.

Figure 1. The examples of DRSs in the box notation are presented in

compact linear format saves space, the box notation in-


Besides basic DRSs, we also have segmented DRSs,

Figure 1: Examples of PMB documents with their scoped meaning representations and the corresponding clausal form.

The first two structures are basic DRSs while the last one is a segmented DRS.

Figure 1:

• If B is a DRS, then ¬B, □B are complex condi-

• If x is a variable, and B is a DRS, then x:B is a complex condi-

• If B and B’ are DRSs, then B⇒B’ and B∨B’ are com-

Besides basic DRSs, we also have segmented DRSs, following [Asher (1993)] and [Asher and Lascarides (2003)].

Hence, DRSs are formally defined as follows:

• If D is a (possibly empty) set of discourse referents, and C a (possibly empty) set of DRS-conditions, then <D,C> is a (basic) DRS;

• If B is a (basic) DRS, and B’ a DRS, then B\{B’ is a (segmented) DRS;

• If U is a set of labelled DRSs, and R a set of discourse re-

DRSs can be visualized in different ways. While the

compact linear format saves space, the box notation in-

creases readability. In this paper we use the latter notation.

The examples of DRSs in the box notation are presented in Figure[1]

However, for evaluation and comparison purposes, we con-

vict a DRS into a flat clausal form, i.e. a set of clauses.

This is carried out by using the labels for DRSs as intro-

duced in [Venhuizen (2015)] and [Venhuizen et al. (2019)],

and breaking down the recursive structure of DRS by as-

signing them a label of the DRS in which they appear. Let
t’, and t” be meta-variables ranging over DRSs or terms. Let C be a set of WordNet concepts, T a set of the them-

atic roles, and O the set of DRS operators (REF, NOT, POS,

NEC, EQU, NEQ, APX, LES, LEQ, TPR, TAB, IMP, DIS, PRP, DRS). The resulting clauses are then of the form t R t’

or t R t” where R ∈ C ∪ T ∪ O. The result of translating

DRSs to sets of clauses is shown in Figure[1] In a clausal

form, it is assumed that different variables are represented

with different variable names and vice versa. Due to this,

before translating a DRS to a clausal form, different dis-

course referents in the DRS must be represented with dif-

ferent variable names. This assumption significantly sim-

plies the matching process between clausal forms (Section

and makes it possible to recover the original box notation

of a DRS from its clausal form.

2.2 Comparing DRSs to AMRs

Since DRSs in a clausal form come close to the triple no-

tation of AMRs ([Cai and Knight, 2013]), and both aim to

model meaning of natural language expressions, it is in-

structive to compare these two meaning representations.

The main difference between AMRs and DRSs is that the

latter ones have explicit scopes (boxes) and scopal oper-

ators such as negation. Due to the presence of scope in

DRSs, their clauses are more complex than AMR triples.

The length of DRS clauses varies from three to four, in

contrast to the constant length of AMR triples. Addition-

ally, DRS clauses contain two different types of variables,

for scopes and discourse referents, whereas AMR triples

have just one type.
Unlike AMRs, DRSs model tense. In general, the tense related information is encoded in a clausal form with three additional clauses, which express a WordNet concept, semantic role and a comparison operator. In order to give an intuition about the diversity of clauses in DRSs, Table 1 shows a distribution of various types of clauses in a corpus of DRSs (see Section 3). Since every logical operator carries a scope, their number represents a lower bound of the number of scopes in the meaning representations. In addition to logical operators, scopes are introduced by presupposition triggers like proper names or pronouns.

To make a meaningful comparison between AMRs and DRSs in terms of size, we compare the DRSs of 250,000 English sentences from the Parallel Meaning Bank (PMB; Abzianidze et al., 2017) to AMRs of the same sentences, produced by the state-of-the-art AMR parser from van Noord and Bos (2017). Statistics of the comparison are shown in Figure 2. On average, DRSs are about twice as large as AMRs, in terms of the number of clauses as well as the number of unique variables. This is obviously due to the explicit presence of scope in the meaning representation. However, for both meaning representations the number of clauses and variables increase linearly with sentence length.

![Figure 2: Comparison of the number of triples/clauses and variables between AMRs and DRSs for sentences of different length.](http://pmb.let.rug.nl/data.php)

### Table 2: Statistics of the first PMB release.

<table>
<thead>
<tr>
<th>Language</th>
<th>Documents</th>
<th>Sentences</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>2,049</td>
<td>2,057</td>
<td>11,664</td>
</tr>
<tr>
<td>German</td>
<td>641</td>
<td>642</td>
<td>3,430</td>
</tr>
<tr>
<td>Italian</td>
<td>387</td>
<td>387</td>
<td>1,944</td>
</tr>
<tr>
<td>Dutch</td>
<td>394</td>
<td>395</td>
<td>2,268</td>
</tr>
</tbody>
</table>

### 3 The Parallel Meaning Bank

The scoped meaning representations, integrating word senses, thematic roles, and the list of operators, form the final product of our semantically annotated corpus: the Parallel Meaning Bank. The PMB is a semantically annotated corpus of English texts aligned with translations in Dutch, German and Italian (Abzianidze et al., 2017). It uses the same framework as the Groningen Meaning Bank (Bos et al., 2017), but aims to abstract away from language-specific annotation models. There are five annotation layers present in the PMB: segmentation of words, multi-word expressions and sentences (Evang et al., 2013), semantic tagging (Bjerva et al., 2016), semantic analysis based on CCG (Lewis and Steedman, 2014), word senses based on WordNet and, finally, discourse representation structures.

To make a meaningful comparison between AMRs and DRSs in terms of size, we compare the DRSs of 250,000 English sentences from the Parallel Meaning Bank (PMB; Abzianidze et al., 2017) to AMRs of the same sentences, produced by the state-of-the-art AMR parser from van Noord and Bos (2017). Statistics of the comparison are shown in Figure 2. On average, DRSs are about twice as large as AMRs, in terms of the number of clauses as well as the number of unique variables. This is obviously due to the explicit presence of scope in the meaning representation. However, for both meaning representations the number of clauses and variables increase linearly with sentence length.

![Figure 2: Comparison of the number of triples/clauses and variables between AMRs and DRSs for sentences of different length.](http://pmb.let.rug.nl/data.php)

The first public release of the PMB contains gold standard scoped meaning representations for over 3,000 sentences in total (see Table 2). The release includes mainly relatively short sentences involving several semantic scope phenomena. A detailed distribution of clause types in the dataset is given in Table 1. A larger amount of texts and more complex linguistic phenomena will be included in future releases.

In addition to the released data, the PMB documents are publicly accessible through a web interface, called the PMB explorer. In the explorer, visitors can view natural language data and discourse structures.
ral language texts with several layers of annotations and compositionally derived meaning representations, and, after registration, edit the annotations. It is also possible to use a word or a phrase search to find certain words or constructions with their semantic analyses. Figure 3 shows the PMB explorer with the semantic analysis of a sentence in the edit mode.

4 Matching Scoped Representations

4.1 Evaluation by Matching

In the context of the Parallel Meaning Bank there are two main reasons to verify whether two scoped meaning representations capture the same meaning or not: (1) to be able to evaluate semantic parsers that produce scoped meaning representations by comparing gold-standard DRSs to system output; and (2) to check whether translations are meaning-preserving; a discrepancy in meaning between source and target could indicate a mistranslation.

The ideal way to compare two meaning representations would be one based on inference. This can be implemented by translating DRSs to first-order formulas and using an off-the-shelf theorem prover to find out whether the two meanings are logically equivalent (Blackburn and Bos, 2005). This method can compare meaning representation that have different syntactic structures but still are equivalent in meaning. The disadvantage of this approach is that it yields just a binary answer: if a proof is found the meanings are the same, else they are not.

An alternative way of comparing meaning representations is comparing the corresponding clausal forms by computing precision and recall over matched clauses (Allen et al., 2008). The advantage of this approach is that it returns a score between 0 and 1, preferring meaning representations that better approximate the gold standard over those that are completely different. Since the variables of different clausal forms are independent from each other, the comparison of two clausal forms boils down to finding a (partial) one-to-one variable mapping that maximizes intersection of the clausal forms. For example, the maximal matching for the clausal forms in Figure 4 is achieved by the following partial mapping from the variables of the left form into the variables of the right one: \{k0→b0, e1→v1\}.

For AMRs, finding a maximal matching is done using a hill-climbing algorithm called SMATCH (Cai and Knight, 2013). This algorithm is based on a simple principle: it checks if a single change in the current mapping results in a better matching mapping. If this is the case, it continues with the new mapping. Otherwise, the algorithm stops and has arrived at the final mapping. This means that it can easily get stuck in local optima. To avoid this, SMATCH does a predefined number of restarts of this process, where each restart starts with a new and random initial mapping. The first restart always uses a ‘smart’ initial mapping, based on matching concepts.

Our evaluation system, called COUNTER is a modified version of SMATCH. Even though clausal forms do not form a graph and clauses consist of either three or four components, the principle behind the variable matching is the same. The actual implementation differs, mainly because SMATCH was not designed to handle clauses with three variables, e.g. \(⟨k0\ Agent\ e1\ x1⟩\).

In contrast to SMATCH, COUNTER takes a set of clauses directly as input. COUNTER also uses two smart initial map-

http://github.com/RikVN/DRS_parsing/
our data, since only a single REF-clause was preserved in the
operator, the REF-clause is kept. This is very infrequent in
is declared outside the scope of negation or an other scope
not all REF-clauses are redundant: if a discourse referent
clausal forms in Figure 4 demonstrates this fact. Note that
responding REF-clause will also match. Comparison of the
ure 4 shows some examples of redundant REF-clauses. Not
ocurs in some basic condition of the same DRS
b1

needed to ensure a reliable score, again increasing runtime.
over, if there are more variables, more restarts might be
increases the average runtime for comparing DRSs. More-
AMRs. This increase in size might be problematic, since it
As we showed in Figure 2, DRSs are about twice as large as
balanced data set. We take 1,000 DRSs produced by the se-
mantic parser Boxer for each sentence length from 2 to 20
(punctuation excluded), resulting in a set of 19,000 DRSs.

\[
\begin{align*}
& b_1 \text{ REF } x_1 \\
& b_1 \text{ male } n.02 \ x_1 \\
& b_2 \text{ REF } x_1 \\
& b_3 \text{ TPR } x_1 \ "now" \\
& b_3 \text{ time } n.08 \ t_1 \\
& k_0 \text{ Agent } e_1 \ x_1 \\
& k_0 \text{ smile } v.01 \ e_1 \\
& k_0 \text{ smile } v.01 \ e_1 \\
& b_2 \text{ Name } x_2 \ "australia" \\
& b_0 \text{ country } n.02 \ x_2
\end{align*}
\]

Figure 4: The SPAR DRS (Section 5.1) matches the DRS
of 00/3514 PMB document with an F-score of 54.5%. If
redundant REF-clauses are ignored, the F-score drops to
40%. These results are achieved with the help of the mapping
\{x0→b0, e1→v1\}.

pings, based on either role-clauses, like \{k0 Agent e1
x1\}, or concept-clauses, like \{k0 smile v.01 e1\}.

Also specific to this method is the treatment of REF-
clauses in the matching process. Before matching two
DRSs, redundant REF-clauses are removed. A REF-clause
\{b1 REF x1\} is redundant if its discourse referent x1 occurs
in some basic condition of the same DRS b1. Figure
4 shows some examples of redundant REF-clauses. Not
removing these redundant clauses would lead to inflated
matching scores since for each matched variable the corre-
responding REF-clause will also match. Comparison of the
clausal forms in Figure 4 demonstrates this fact. Note that
not all REF-clauses are redundant: if a discourse referent
is declared outside the scope of negation or an other scope
operator, the REF-clause is kept. This is very infrequent in
our data, since only a single REF-clause was preserved in
2,049 examples.

4.2 Evaluating Matching
As we showed in Figure 2, DRSs are about twice as large as
AMRs. This increase in size might be problematic, since it
increases the average runtime for comparing DRSs. More-
over, if there are more variables, more restarts might be
needed to ensure a reliable score, again increasing runtime.

Therefore, our goal is that COUNTER gets close to op-
timal performance in reasonable time. Since we want to
be sure that this also holds for longer sentences, we use a
balanced data set. We take 1,000 DRSs produced by the se-
matic parser Boxer for each sentence length from 2 to 20
(punctuation excluded), resulting in a set of 19,000 DRSs.

To test COUNTER in a realistic setting, we cannot com-
pare the DRSs to themselves or to a DRS of the translation,
since those are too similar. Therefore, the 19,000 English
sentences of the DRS are parsed by an existing AMR parser
and Bos (2017) yields an F-score of 55.0 for 10 restarts, but
motivated by Bos (2016). An example of translating an AMR
to a clausal form of a DRS is shown in Figure 5. We con-
vert AMR relations to DRS roles by employing a manually
created translation dictionary, including rules for semantic
roles (e.g. :ARG0↦Agent and :ARG1↦Patient) and pronouns (e.g. she↦female.n.02). Since AMRs do not contain tense information, past tense clauses are
produced for the first verb in the AMR (see four tense related
clauses in Figure 5). Also, since AMRs do not use Word-
Net synsets, all concepts get a default first sense, except for
concepts that are added by concept-specific rules, such as
female.n.02 and time.n.08.

We compare the sets of DRSs using different numbers
of restarts to find the best trade-off between speed and ac-
curacy. The results are shown in Table 3. The optimal
scores are obtained using a Prolog script that performs an
exhaustive search for the optimal matching. As expected,
increasing the number of restarts benefits performance. Cai
and Knight (2013) consider four restarts the optimal trade-
off between accuracy and speed, showing no improvement
in F-score when using more than ten restarts. Contrary to
SMATCH, performance for COUNTER still increases with
more than 4 restarts. In our case, it is a bit harder to select
an optimal number of restarts, since this number depends
on the length of the sentence, as shown in Figure 6. We see
that for long sentences, 5 and 10 restarts are not sufficient to
get close to the optimal, while for short sentences 5 restarts
might be considered enough. In general, the best trade-off
between speed and accuracy is approximately 20 restarts.

01/3445: He smiled. 00/3514: She fled Australia.

\[
\begin{align*}
& x_1 \ e_1 \ t_1 \\
& \text{male.n.02}(x_1) \\
& \text{smile.v.01}(e_1) \\
& \text{Time}(e_1, t_1) \\
& \text{Agent}(e_1, x_1) \\
& \text{time.n.08}(t_1) \\
& t_1 \prec \text{now}
\end{align*}
\]

\[
\begin{align*}
& x_1 \ x_2 \ v_1 \ t_1 \\
& \text{female.n.02}(x_1) \\
& \text{flee.v.01}(v_1) \\
& \text{Time}(v_1, t_1) \\
& \text{Source}(v_1, x_2) \\
& \text{Theme}(v_1, x_1) \\
& \text{time.n.08}(t_1) \\
& t_1 \prec \text{now}
\end{align*}
\]

\[
\begin{align*}
& \text{country.n.02}(x_2) \\
& \text{Name}(x_2, \text{australia})
\end{align*}
\]

\[
\begin{align*}
& \text{b0 REF } x_1 \\
& \text{b0 remove v.01 } x_1 \\
& \text{b4 REF } x_5 \\
& \text{b4 TPR } x_5 \ "now" \\
& \text{b4 time } n.08 \ x_5 \\
& \text{b0 Time } x_1 \ x_5 \\
& \text{b0 Agent } x_1 \ x_2 \\
& \text{b1 REF } x_2 \\
& \text{b1 female.n.02 } x_2 \\
& \text{b0 Patient } x_1 \ x_3 \\
& \text{b2 REF } x_3 \\
& \text{b2 dish.n.01 } x_3 \\
& \text{b0 Theme } x_1 \ x_4 \\
& \text{b3 REF } x_4 \\
& \text{b3 table.n.01 } x_4
\end{align*}
\]

Figure 5: A clausal form obtained from an automatically
generated AMR of the document 14/0849.
Table 3: Results of comparing 19,000 Boxer-produced DRSs to DRSs produced by AMR2DRS, for different number of restarts. For three or more restarts, we always use the smart role and concept mapping.

<table>
<thead>
<tr>
<th>Restarts</th>
<th>P%</th>
<th>R%</th>
<th>F1%</th>
<th>Time (h:m:s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(random) 1</td>
<td>27.20</td>
<td>22.71</td>
<td>24.75</td>
<td>4:19</td>
</tr>
<tr>
<td>(smart concepts) 1</td>
<td>27.45</td>
<td>22.92</td>
<td>24.98</td>
<td>4:35</td>
</tr>
<tr>
<td>(smart roles) 1</td>
<td>27.27</td>
<td>22.76</td>
<td>24.81</td>
<td>4:37</td>
</tr>
<tr>
<td>5</td>
<td>30.25</td>
<td>25.25</td>
<td>27.53</td>
<td>19:33</td>
</tr>
<tr>
<td>10</td>
<td>30.65</td>
<td>25.59</td>
<td>27.89</td>
<td>37:08</td>
</tr>
<tr>
<td>20</td>
<td>30.84</td>
<td>25.75</td>
<td>28.07</td>
<td>1:10:13</td>
</tr>
<tr>
<td>30</td>
<td>30.90</td>
<td>25.80</td>
<td>28.12</td>
<td>1:41:43</td>
</tr>
<tr>
<td>50</td>
<td>30.94</td>
<td>25.83</td>
<td>28.16</td>
<td>2:41:38</td>
</tr>
<tr>
<td>75</td>
<td>30.96</td>
<td>25.85</td>
<td>28.17</td>
<td>3:53:01</td>
</tr>
<tr>
<td>100</td>
<td>30.97</td>
<td>25.85</td>
<td>28.18</td>
<td>5:01:25</td>
</tr>
<tr>
<td>Optimal</td>
<td>30.98</td>
<td>25.86</td>
<td>28.19</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Comparison of the differences to the optimal F-score per sentence length for different number of restarts.

5 COUNTER in Action

5.1 Semantic Parsing

The first purpose of COUNTER is to evaluate semantic parsers for DRSs. Since this is a new task, there are no existing systems that are able to do this. Therefore, we show the results of three baseline systems PMB PIPELINE, SPAR, and AMR2DRS (Subsection 4.2). The PMB PIPELINE produces a DRS via the pipeline of the tools used for automatic annotation of the PMB. This means that it has no access to manual corrections, and hence it uses the most frequent word senses and default VerbNet roles. SPAR is a trivial semantic ‘parser’ which always outputs the DRS that is most similar to all other DRSs in the most recent PMB release (the left-hand DRS in Figure 4).

The results of the three baseline parsers are shown in Table 4. The surprisingly high score of SPAR is explained by the fact that the first PMB release mainly contains relatively short sentences with little structural diversity. The average number of clauses per clausal form (excluding redundant REF-clauses) is 8.7, where a substantial share (approximately 3) comes from tense related clauses. Due to this fact, guessing temporal clauses for short sentences has a big impact on F-score. This is illustrated by the comparison of the clausal forms in Figure 4, where matching only temporal clauses results in an F-score of 40%.

AMR2DRS outperforms SPAR by a considerable margin, but is still far from optimal. This is also the case for PMB PIPELINE, which shows that, within the PMB, manual annotation is still required to obtain gold standard meaning representations.

5.2 Comparing Translations

The second purpose of COUNTER is checking whether translations are meaning-preserving. As a pilot study, we compare the gold standard meaning representations of German, Italian and Dutch translations in the release to their English counterparts. The results are shown in Table 5. The high F-scores indicate that the meaning representations are often syntactically very similar, if not identical. However, there is a considerable subset of meaning representations which are different from the English ones, indicating that there is at least a slight discrepancy in meaning for those translations.

Table 5: Comparing meaning representations of English texts to those of German, Italian and Dutch translations.

<table>
<thead>
<tr>
<th>Language</th>
<th>F-score%</th>
<th>Docs</th>
<th>F&lt;1.0</th>
<th>% total</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>98.4</td>
<td>579</td>
<td>61</td>
<td>10.5</td>
</tr>
<tr>
<td>Italian</td>
<td>97.6</td>
<td>341</td>
<td>46</td>
<td>13.5</td>
</tr>
<tr>
<td>Dutch</td>
<td>98.3</td>
<td>355</td>
<td>37</td>
<td>10.4</td>
</tr>
</tbody>
</table>

Manual analysis of these discrepancies showed that there are several different causes for a discrepancy to arise. In most of the cases (38%), a human annotation error was made. In 34% of cases, a definite description was used in one language but not in the other. Examples are ‘has long hair’ with the Italian translation ‘ha i capelli lunghi’, and ‘escape from prison’ with the Dutch translation ‘vluchte uit de gevangenis’. In 15% of cases proper names were translated (e.g. ‘United States’ and ‘Stati Uniti’). This is not accounted for, since we do not currently make use of grounding proper names to a unique identifier, for instance by wikification (Cucerzan, 2007), or by using a language-independent transliteration of names. In 13% of cases the translation was either non-literal or incorrect. Examples are ‘Tom lacks experience’ with the Dutch translation ‘Tom
She removed the dishes from the table. Ze ruimde de tafel af.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$e_1$</th>
<th>$x_3$</th>
<th>$t_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>female.n.02($x_1$)</td>
<td>remove.v.01($e_1$)</td>
<td>Time($e_1$, $t_1$)</td>
<td>Source($e_1$, $x_3$)</td>
<td>Theme($e_1$, $x_2$)</td>
</tr>
<tr>
<td>$t_1 &lt; \text{now}$</td>
<td>dish.n.01($x_2$)</td>
<td>table.n.03($x_3$)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As a consequence, semantic parsing for DRSs is a challenging task. Compared to Abstract Meaning Representations, the number of clauses and variables in a DRS is about two times larger on average. Moreover, compared to AMRs, DRSs rarely contain clauses with single variables. All non-logical symbols used in DRSs are grounded in WordNet and VerbNet (with a few extensions). This makes evaluation using matching computationally challenging, in particular for long sentences, but our matching system COUNTER achieves a reasonable trade-off between speed and accuracy.

Several extensions to the annotation scheme are possible. Currently, the DRSs for the non-English languages contain references to synsets of the English WordNet. Conceptually, there is nothing wrong with this (as synsets can be viewed as identifiers for concepts that are language-independent), but for practical reasons it makes more sense to provide links to synsets of the original language. In a similar vein, we plan to experiment in case they do not fully match.

As for other future work, we plan to include a more fine-grained matching regarding WordNet synsets, since the current evaluation of concepts is purely string-based, with only identical strings resulting in a matching clause. For many synsets, however, it is possible to refer to them with more than one word.PDS.Sen@ium triple, and this should be accounted for (e.g. fox.n.02 and dodger.n.01 both refer to the same synset). In a similar vein, we plan to experiment with including WordNet concept similarity techniques in COUNTER to compute semantic distances between synsets, in case they do not fully match.

Finally, we would like to stimulate research on semantic parsing with scoped meaning representations. Not only are we planning to extend the coverage of phenomena and the number of texts with gold-standard meaning representations for the four languages, we also aim to organize a shared task on DRS parsing for English, German, Dutch and Italian in the near future.

7 Bibliographical References

