# Parser Evaluation Using Elementary Dependency Matching

**Rebecca Dridan**
NICTA Victoria Research Laboratory
Dept. of Computer Science and Software Engineering
University of Melbourne
rdridan@csse.unimelb.edu.au

**Stephan Oepen**
Department of Informatics
Universitetet i Oslo
oe@ifi.uio.no

## Abstract

We present a perspective on parser evaluation in a context where the goal of parsing is to extract meaning from a sentence. Using this perspective, we show why current parser evaluation metrics are not suitable for evaluating parsers that produce logical-form semantics and present an evaluation metric that is suitable, analysing some of the characteristics of this new metric.

## 1 Introduction

A plethora of parser evaluation metrics exist, which evaluate different types of information, at different levels of granularity, using different methods of calculation. All attempt to measure the syntactic quality of parser output, but that is not the only goal of parsing. The DELPH-IN consortium[1] has produced many grammars of different languages, as well as a number of parsers, all with the aim of extracting meaning from text. In order to drive development, we need a parser evaluation metric that evaluates against that goal. That is, we require a metric that measures semantic rather than syntactic output. In the following section, we reflect on and categorize the semantic information we wish to evaluate, and discuss how current metrics partially overlap with this framework. We then, after describing some of the specifics of the tools we work with, present an evaluation metric that fits within the given framework, and show, using a couple of case studies, some characteristics of the metric.

## 2 Semantic Information

Our primary goal in parsing is to extract meaning from text. To evaluate progress towards this goal in a granular fashion, one needs to break up the semantic information into discrete elements. For this purpose, we distinguish three broad classes of information that contribute to meaning:

- **class 1** core functor–argument structure, whether syntactic or semantic
- **class 2** predicate information, such as the lemma, word category, and sense
- **class 3** properties of events and entities, such as tense, number, and gender

The widely-used PARSEVAL metric (Black et al., 1991) evaluates phrase structure, which covers none of these classes directly. Dependency-based evaluation schemes, such as those used by MaltParser (Nivre et al., 2004) and MSTParser (McDonald et al., 2005) evaluate **class 1** surface information. The annotation used in the Briscoe and Carroll (2006) DepBank for parser evaluation also describes just **class 1** syntactic information, although the relationships are different to those that MaltParser or MSTParser produce. The annotation of the original King et al. (2003) PARC700 DepBank does describe all three classes of information, but again in terms of syntactic rather than semantic properties.

A common element between all the dependency types above is the use of grammatical relations to describe **class 1** information. That is, the dependencies are usually labels like SUBJ, OBJ, MOD, etc. While these grammatical functions allow one to describe the surface linguistic structure, they do not make the underlying deep structure explicit. This deep structure describes semantic rather than syntactic arguments and can be seen in resources such as the Prague Dependency Treebank (Böhmová et al., 2003) and the Redwoods Treebank (Oepen et al., 2004b). Using this semantic argument structure for parser evaluation not only gets closer to the actual sentence meaning that we are trying to extract, but is potentially more general, as there is generally wider agreement on semantic arguments than on, for example, whether the main verb depends on the auxiliary, or

[1] See http://www.delph-in.net for background.



*Proceedings of the 12th International Conference on Parsing Technologies*, pages 225–230,
October 5-7, 2011, Dublin City University. 

vice versa.[2]

## 3 Background

The parser that we will be evaluating in this work encodes its semantic output in the form of Minimal Recursion Semantics (Copestake et al., 2005), although the derivation that we use for the evaluation metric should be compatible with any parser that produces information in classes given in the previous section.

### 3.1 Minimal Recursion Semantics

Minimal Recursion Semantics (MRS) is a flat semantic formalism that represents semantics as a bag of *elementary predications* and a set of underspecified scopal constraints. An elementary predication can be directly related to words in the text, or can reflect a grammatical construction, such as compounding. Each elementary predication has a relation name, a label and a distinguished variable (designated ARG0). Arguments of a predication are identified by 'bleached' ARG*n* roles (which are to be semantically interpreted for classes of predicates). Figure 1 shows the MRS analysis of *He persuaded Kim to leave*. Here we see six elementary predications, four with text referents and two as construction-specific covert quantifiers. The ARG1, ARG2 and ARG3 roles of the verbal predicates describe the predicate–argument relations and demonstrate co-indexation between the ARG2 of *persuade* and the ARG1 of *leave*. Entity and event variables carry properties such as gender or tense. An evaluation scheme based on MRS therefore allows us to evaluate **class 1** information using the roles, **class 2** information through predicate names and **class 3** information from the properties of the distinguished variables.

### 3.2 Setup

We will use the PET parser (Callmeier, 2000) and associated grammars as our test environment to evaluate. The traditional accuracy metric for PET has been sentence accuracy which requires an exact match against the very fine-grained gold analysis, but arguably this harsh metric supplies insufficient information about parser performance on its own. In order to evaluate a parser for its use in an application, we are also interested in knowing how good the top ranked parse is, rather than only whether it is the very best parse possible. Even if the goal of evaluation were just parser development, a nuanced granular evaluation may help reveal what types of mistakes a parser is making.

[2] At the same time, we wish to focus *parser* evaluation on information determined solely by grammatical analysis, i.e. all contributions to interpretation by syntax, and only those. For these reasons, the task of semantic role labeling (SRL) against PropBank-style target representations (Kingsbury et al., 2002) is too far removed from parser evaluation proper; Copestake (2009) elaborates this argument.

## 4 EDM: Elementary Dependency Match

In addition to our focus on semantic information, we considered two other requirements for an effective parser evaluation metric. It should be:

1. understandable not just by parser developers, but also potential users of the parser.
2. configurable to suit the level of detail required for a particular scenario.

### 4.1 Elementary Dependencies

The metric we have devised to satisfy these requirements is Elementary Dependency Match (EDM), based on so-called Elementary Dependencies (EDs), a variable-free reduction of MRS developed by Oepen and Lønning (2006).[3] In our work, we use sub-string character spans (e.g. <3:12>) to identify nodes in the dependency graph, to facilitate alignment of corresponding elements across distinct analyses. In keeping with our information classes, this allows us to separate the evaluation of **class 2** information from **class 1**. Our EDM metric hence consists of three triple types which align with the three information classes:

| | | | |
|---|---|---|---|
| ARGS: | $span_i$ | $role_j$ | $span_k$ |
| NAMES: | $span_i$ | NAME | $relation_i$ |
| PROPS: | $span_i$ | $property_j$ | $value_j$ |

In these forms, $relation$ is the predicate name of an elementary predication from the MRS, $role$ is an argument label such as ARG1, $property$ refers to an attribute such as TENSE or GEND and $value$ is an appropriate instantiation for the respective property. Figure 2 shows the triples produced for the MRS in Figure 1. The text segment associated with each character span is shown for illustrative purposes, but is not part of the triple.

During evaluation, we compare the triples from the gold standard analysis with that ranked top by

[3] In more recent work, Copestake (2009) shows how essentially the same reduction can be augmented with information about the underspecified scope hierarchy, so as to yield so-called Dependency MRS (which unlike EDs facilitates bidirectional conversion from and to the original MRS).

$\langle h_1$,
$h_3$:pron<0:2>(ARG0 $x_4${PERS *3*, NUM *sg*, GEND *m*, PRONTYPE *std_pron*}),
$h_5$:pronoun_q<0:2>(ARG0 $x_4$, RSTR $h_6$, BODY $h_7$),
$h_8$:_persuade_v_of<3:12>(ARG0 $e_2${SF *prop*, TENSE *past*, MOOD *indicative*}, ARG1 $x_4$, ARG2 $x_{10}$, ARG3 $h_9$),
$h_{11}$:proper_q<13:16>(ARG0 $x_{10}${PERS *3*, NUM *sg*}, RSTR $h_{12}$, BODY $h_{13}$),
$h_{14}$:named<13:16>(ARG0 $x_{10}$, CARG *Kim*),
$h_{15}$:_leave_v_1<20:26>(ARG0 $e_{16}${SF *prop-or-ques*, TENSE *untensed*, MOOD *indicative*}, ARG1 $x_{10}$, ARG2 $p_{17}$)
{$h_{12} =_q h_{14}, h_9 =_q h_{15}, h_6 =_q h_3$}$\rangle$

Figure 1: MRS representation of *He persuaded Kim to leave.*

| | | | | |
|---|---|---|---|---|
| "He" | <0:2> | ARG0 | <0:2> | "He" |
| "persuaded" | <3:12> | ARG1 | <0:2> | "He" |
| "persuaded" | <3:12> | ARG2 | <13:16> | "Kim" |
| "persuaded" | <3:12> | ARG3 | <20:26> | "leave." |
| "Kim" | <13:16> | ARG0 | <13:16> | "Kim" |
| "leave." | <20:26> | ARG1 | <13:16> | "Kim" |
| "He" | <0:2> | NAME | pronoun_q | |
| "He" | <0:2> | NAME | pron | |
| "persuaded" | <3:12> | NAME | _persuade_v_of | |
| "Kim" | <13:16> | NAME | proper_q | |
| "Kim" | <13:16> | NAME | named | |
| "leave." | <20:26> | NAME | _leave_v_1 | |
| "He" | <0:2> | GEND | *m* | |
| "He" | <0:2> | NUM | *sg* | |
| "He" | <0:2> | PERS | *3* | |
| "He" | <0:2> | PRONTYPE | *std_pron* | |
| "persuaded" | <3:12> | MOOD | *indicative* | |
| "persuaded" | <3:12> | SF | *prop* | |
| "persuaded" | <3:12> | TENSE | *past* | |
| "Kim" | <13:16> | NUM | *sg* | |
| "Kim" | <13:16> | PERS | *3* | |
| "leave." | <20:26> | MOOD | *indicative* | |
| "leave." | <20:26> | SF | *prop-or-ques* | |
| "leave." | <20:26> | TENSE | *untensed* | |

Figure 2: Gold triples for *He persuaded Kim to leave.*

the parser, and calculate precision, recall and $F_1$-score across all triples, as well as across the three separate triple types (NAME, ARG and PROP).

### 4.2 Alternate Configurations

The full EDM metric weights each triple equally which may not be ideal for all scenarios. The division by triple type gives one alternative view that provides a more complete picture of what sort of mistakes are being made by the parser. For particular applications, it might be that only **class 1** information will be used, and in that case just measuring ARGs might be a better metric. Further fine-tuning is possible by assigning weights to individual predicate types via a configuration file similar to the parameter files used with the EvalB PARSEVAL script (Sekine and Collins, 1997). This will allow a user to, for example, assign lower weight to entity properties, or only evaluate ARG1 and ARG2 roles. One particular configuration we have found useful is to assign zero weight to the PROP triples and only evaluate ARGs and NAMEs. While the **class 3** information is useful for applications such as machine translation, and ideally would be evaluated, some applications don't make use of this information, and so, in certain scenarios, it makes sense to ignore these triples in evaluation. This configuration produces a metric broadly similar to the CCG dependencies used by Clark and Curran (2007) and also to the predicate argument structures produced by the Enju parser (Miyao and Tsujii, 2008), in terms of the information classes included, although the CCG dependencies again encode syntactic rather than semantic structure.

## 5 Analysis

To get some idea of the numeric range of the different EDM configurations, we parsed a section of the SemCor corpus (Miller et al., 1994) using the English Resource Grammar (ERG: (Flickinger, 2000)), and then calculated the average $F_1$-score for each rank, as ranked by the statistical model packaged with the ERG. Figure 3 shows the relative differences between five configurations: all triples together (EDM), the NAME, ARG and PROP triple types separately ($EDM_N$, $EDM_A$ and $EDM_P$, respectively) and measuring just the NAME and ARG types together ($EDM_{NA}$).

We can see that all configurations show approximately the same trends, and maintain their relative order. $EDM_P$ is consistently higher, which follows from the fact that many of the properties are inter-dependent, and that the parser enforces agreement. Most difficult to identify correctly is the ARG type, which represent the core semantic arguments. All of the scores are quite high, even at the 40th rank parse, which is due to using a highly constrained grammar with fine-grained analyses that can vary in only small details.

To get a different view of the information that EDM provides, we looked at different scenarios

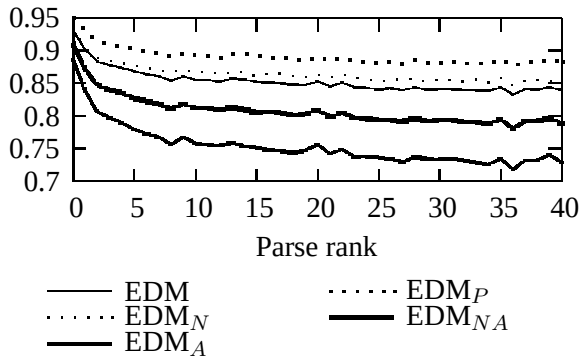


Figure 3: Average $F_1$ score at each rank (up to 40).

| | Config 1 | | | Config 2 | | |
|---|---|---|---|---|---|---|
| **Sent. Acc.** | 0.095 | | | 0.093 | | |
| | **P** | **R** | **F** | **P** | **R** | **F** |
| **EDM** | 0.847 | 0.677 | 0.753 | 0.847 | 0.693 | 0.763 |
| **EDM$_{NA}$** | 0.796 | 0.635 | 0.707 | 0.798 | 0.652 | 0.717 |
| **EDM$_A$** | 0.778 | 0.620 | 0.690 | 0.780 | 0.637 | 0.701 |
| **EDM$_N$** | 0.815 | 0.651 | 0.724 | 0.815 | 0.668 | 0.734 |
| **EDM$_P$** | 0.890 | 0.714 | 0.792 | 0.890 | 0.729 | 0.801 |

Table 1: Comparing unknown word handling configurations.

that allow us to see relative differences in parser performance, measured using EDM and variants, as well as the traditional sentence accuracy.

### 5.1 Cross-Configuration

One possible evaluation scenario involves changing a parameter in the parser and measuring the effect. The results in Table 1 come from parsing a single corpus using a variant of the ERG with a much smaller lexicon in order to test two unknown word handling configurations.

The sentence accuracy figures are very low, since the grammar has been limited, and show no real difference between the two configurations. In the EDM results, we can see that, while the precision between the two configurations is very similar, recall is consistently lower for Config 1 (which had a slightly better sentence accuracy).

### 5.2 Cross-Grammar

In this comparison, we look at two different grammars, over parallel test data.[4] The Spanish Resource Grammar (SRG: (Marimon et al., 2007)) also produces MRS, although properties are treated differently, so we leave out the EDM and EDM$_P$ metric for compatibility and compare to ERG performance over the same small test set.

[4]The MRS test suite was constructed to represent a range of phenomena and consists of 107 short sentences which have been translated into multiple languages, maintaining parallel MRS analyses as far as possible.

| | SRG | | | ERG | | |
|---|---|---|---|---|---|---|
| **Sent. Acc.** | 0.95 | | | 0.85 | | |
| | **P** | **R** | **F** | **P** | **R** | **F** |
| **EDM$_{NA}$** | 0.97 | 0.97 | 0.97 | 0.92 | 0.93 | 0.92 |
| **EDM$_A$** | 0.96 | 0.95 | 0.95 | 0.90 | 0.91 | 0.90 |
| **EDM$_N$** | 0.98 | 0.98 | 0.98 | 0.93 | 0.95 | 0.94 |

Table 2: Comparing between the SRG and ERG grammars over a parallel test suite. PROP type triples are excluded for compatibility.

While the SRG is a less mature grammar, and does not analyse the full range of constructions that the ERG parses, EDM allows us to compare over items and information types that both grammars cover, and in Table 2 we can see that the SRG ranking model performs better over this data.

## 6 Conclusion

The current range of parser evaluation metrics all evaluate the syntactic quality of parser output, which makes them unsuitable to evaluate parsers which aim to output semantic analysis. The EDM metric we describe here allows us to evaluate the semantic output of any parser that can encode information in the Minimal Recursion Semantic framework, and indeed, the derivation that we use should be generalisable to any logical-form semantic output. This metric can measure three different classes of deep semantic information, and can be configured to evaluate whatever level is suitable for the potential application, or for the parser being evaluated. We have demonstrated that EDM and its variants, together with sentence accuracy, can give a detailed picture of how accurately a parser can extract meaning from text, allowing useful comparisons in a variety of circumstances. Furthermore, since MRS is used in applications and other semantic research (Oepen et al., 2004a; Dridan, 2007; Schlangen and Lascarides, 2003; Fuchss et al., 2004), the metric we have described here may prove useful in other areas where semantic comparison is required.

## Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

## References


Ezra Black, Steve Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Phil Harrison, Don Hindle, Robert Ingria, Fred Jelinek, J. Klavans, Mark Liberman, Mitch Marcus, S. Roukos, B. Santorini, and Tomek Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the Workshop on Speech and Natural Language*, pages 306–311, Pacific Grove, USA.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three level annotation scenario. In Anne Abeill, editor, *Treebanks: building and using parsed corpora*. Springer.

Ted Briscoe and John Carroll. 2006. Evaluating the accuracy of an unlexicalised statistical parser on the PARC DepBank. In *Proceedings of the 44th Annual Meeting of the ACL and the 21st International Conference on Computational Linguistics*, pages 41–48, Sydney, Australia.

Ulrich Callmeier. 2000. PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–107.

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

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

Ann Copestake. 2009. Invited talk: Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens, Greece.

Rebecca Dridan. 2007. Using Minimal Recursion Semantics in Japanese question answering. Master's thesis, The University of Melbourne.

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

Ruth Fuchss, Alexander Koller, Joachim Niehren, and Stefan Thater. 2004. Minimal recursion semantics as dominance constraints: Translation, evaluation, and analysis. In *Proceedings of the 42nd Annual Meeting of the ACL*, pages 247–254, Barcelona, Spain.

Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the LINC-03 Workshop*, pages 1–8, Budapest, Hungary.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the Penn treebank. In *Proceedings of the Human Language Technology 2002 Conference*, pages 252–256, San Diego, California.

Montserrat Marimon, Núria Bel, and Natalia Seghezzi. 2007. Test-suite construction for a Spanish grammar. In *Proceedings of the GEAF 2007 Workshop*, pages 224–237, Stanford, USA.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 523–530, Vancouver, Canada.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of ARPA Human Language Technology Workshop*, pages 240–243, Plainsboro, USA.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the 8th Conference on Natural Language Learning (CoNLL-2004)*, pages 49–56, Boston, USA.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference*

*on Language Resources and Evaluation (LREC 2006)*, pages 1250–1255, Genoa, Italy.

Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. 2004a. Små kapp-ete med trollet? Towards MRS-based Norwegian—English machine translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 11–20, Baltimore, USA.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004b. LinGO redwoods: a rich and dynamic treebank for HPSG. *Journal of Research in Language and Computation*, 2(4):575–596.

David Schlangen and Alex Lascarides. 2003. A compositional and constraint-based approach to non-sentential utterances. In *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar*, pages 380–390, East Lansing, USA.

Satoshi Sekine and Michael Collins. 1997. EvalB: a bracket scoring program. `http://nlp.cs.nyu.edu/evalb/`.