The ERG at MRP 2019: Radically Compositional Semantic Dependencies

Stephan Oepen[♣] and **Dan Flickinger**[♠]

[♣] University of Oslo, Department of Informatics

[♠] Stanford University, Center for the Study of Language and Information

oe@ifi.uio.no, danf@stanford.edu

Abstract

The English Resource Grammar (ERG) is a broad-coverage computational grammar of English that derives underspecified logical-form representations of meaning. Elementary Dependency Structures (EDS) and DELPH-IN MRS Bi-Lexical Dependencies (DM) are graph-based simplifications of ERG meaning representations. As a point of reference outside the official competition of the 2019 Shared Task on Cross-Framework Meaning Representation Parsing, we evaluate ERG-derived EDS and DM graphs. These graphs yield higher accuracy scores than the purely data-driven parsers in the shared task, suggesting that the general-purpose grammatical knowledge encoded in the ERG aids parsing into these meaning representations.

1 Introduction

Two of the target representations in the 2019 Shared Task on Cross-Framework Meaning Representation Parsing (MRP 2019; Oepen et al., 2019) derive from the framework dubbed English Resource Semantics (ERS; Flickinger et al., 2014; Bender et al., 2015). ERS instantiates the designer logic for scopally underspecified meaning representation called Minimal Recursion Semantics (MRS; Copestake et al., 2005); in and of themselves, ERS terms are logic- rather than graph-based, i.e. require conversion into graph-structured representations of meaning in the context of the MRP shared task. Elementary Dependency Structures (EDS; Oepen and Lønning, 2006) and DELPH-IN MRS Bi-Lexical Dependencies (DM; Ivanova et al., 2012) achieve simplification of ERS into labeled directed graphs by elimination of most of the information regarding scope underspecification and, in the case of DM, further reduction into pure bi-lexical graphs. Oepen et al. (2019) provide additional background on these representations. This paper gives some linguistic and technical background on ERS parsing (§2), summarizes the processes used in deriving EDS and DM graphs for the MRP evaluation data (§3), and puts quantitative ERS parsing results into the perspective of the shared task at large (§4).

2 The LinGO English Resource Grammar and Redwoods Treebank

At the core of this work are two linguistic resources that have been under continuous development for multiple decades now, as part of the world-wide Deep Linguistic Processing with HPSG Initiative (DELPH-IN; http://delph-in.net). First, the LinGO English Resource Grammar (ERG; Flickinger, 2000) is an implementation of the grammatical theory of Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994) for English, i.e. a computational grammar that can be used for parsing and generation. Development of the ERG started in 1993, building conceptually on earlier work on unification-based grammar engineering for English at Hewlett Packard Laboratories (Gawron et al., 1982). The ERG has continuously evolved through a series of R&D projects (and a small handful of commercial applications) and today allows the grammatical analysis of running text across domains and genres. The hand-built ERG lexicon of some 38,000 lemmata (for 27,000 distinct citation forms) aims for complete coverage of function words and open-class words with 'non-standard' syntactic properties (e.g. argument structure). Built-in support for light-weight named entity recognition and an unknown word mechanism combining statistical PoS tagging and on-the-fly lexical instantiation for 'standard' open-class words (e.g. names or non-relational common nouns and adjectives) typically enable the grammar to derive complete syntactico-semantic analyses for 85 – 95 percent of all utterances in standard corpora, including newspaper text, the English Wikipedia, or bio-medical research literature (Flickinger et al., 2017). Parsing times for these data sets measure in seconds per sentence, time comparable to human production or comprehension.

Second, since around 2001 the ERG has been accompanied by a selection of development cor-

Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 CoNLL, pages 40–44

https://doi.org/10.18653/v1/K19-2003

pora, where for each sentence an annotator has selected the intended analysis among the alternatives provided by the grammar (or has recorded that no appropriate analysis is available, in a given version of the grammar). This companion resource is called the LinGO Redwoods Treebank (Oepen et al., 2004). For each release of the ERG, a corresponding version of the treebank has been produced, manually validating and updating existing analyses to reflect changes in the underlying grammar, as well as 'picking up' analyses for previously out-of-scope inputs and new development corpora. Since mid-2016, the current version of Redwoods (dubbed Ninth Growth, corresponding to ERG release 1214) encompasses gold-standard analyses for some 85,400 utterances (or close to 1.3 million tokens) of running text from half a dozen different genres and domains, including the first 22 sections of the venerable Wall Street Journal (WSJ) text in the Penn Treebank (PTB; Marcus et al., 1993).

The original motivation for treebanking ERG analyses was to enable training discriminative parse ranking models, i.e. a conditional probability distribution over ERG derivations (Johnson et al., 1999). For this purpose, the treebank must disambiguate at the same level of granularity as is maintained in the grammar, i.e. encode its exact linguistic distinctions. Furthermore, to train discriminative (i.e. conditional) stochastic models, both the intended as well as the dispreferred analyses are needed.

The Redwoods treebank is built exclusively from ERG analyses, i.e. full HPSG syntactico-semantic signs. Annotation in Redwoods amounts to disambiguation among the candidate analyses derived by the grammar (identifying the intended parse) and, of course, analytical validation of the final result. To make this task practical, a specialized tree selection tool extracts a set of what are called *discriminants* from the complete set of analyses. Discriminants encode contrasts among alternate analyses—for example whether to treat a word like *crop* as nominal or verbal, or where to attach a prepositional phrase modifier. While picking one full analysis (among a set of hundreds or thousands of trees) would be daunting (to say the least), the isolated contrasts presented as discriminants are comparatively easy to judge for a human annotator.

Discriminant-based tree selection was first proposed by Carter (1997) and has since been successfully applied to a range of grammatical frameworks. To the best of our knowledge, Redwoods is the most comprehensive such effort, complementing the original proposal by Carter (1997) with the notion of *dynamic* treebanking, in two senses of this term. First, different views can be projected from the multi-stratal HPSG analyses at the core of the treebank, highlighting subsets of the syntactic or semantic properties of each analysis, e.g. HPSG derivations, more conventional phrase structure trees, full logical-form meaning representations, and various variable-free forms of semantic dependency graphs—including EDS and DM.

Second, the dynamic treebank is extended and refined *over time*. As the grammar (the core repository of knowledge about derivation and composition) evolves, dynamic refinement refers to the ability to mostly automatically update the Redwoods treebank, to for example add detail to the linguistic analyses or apply targeted error correction while minimizing any loss of manual input from previous annotation cycles. Although we can by no means quantify precisely the effort devoted to ERG and Redwoods development to date, we estimate that in excess of thirty person years have been accumulated between 1993 and 2019.

3 Parsing with the ERG

There are several highly engineered implementations of the DELPH-IN feature structure reference formalism; for our experiments we used the PET parser of Callmeier (2002), as bundled in the open-source distribution of DELPH-IN resources called LOGON (Lønning and Oepen, 2006).[1] At its core, PET is a classic, agenda-driven chart parser (Kay, 1986), synthesizing a large body of algorithm design for efficient feature structure manipulation and unification-based parsing by among others Tomabechi (1995), Malouf et al. (2000), Erbach (1991), Kiefer et al. (1999), and Oepen and Callmeier (2000). The parser achieves exact inference by constructing the complete *parse forest*, factoring local ambiguity under feature structure subsumption (a technique termed *retroactive packing* by Oepen and Carroll, 2000) and subsequently enumerating n-best full derivations from the forest according to a discriminative parse ranking model in the tradition of Johnson et al. (1999) and Toutanova et al. (2005).

Despite the non-local nature of features (of ERG derivations) used in parse ranking, the *selective unpacking* procedure of Carroll and Oepen (2005)

[1] See http://moin.delph-in.net/LogonTop.

		Tops			Labels			Properties			Anchors			Edges			Attributes			All		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F
DM	ERG	.92	.92	.918	.99	.99	**.987**	.96	.96	**.956**	.99	.99	**.994**	.91	.91	.912	–	–	–	.96	.96	**.961**
		.95	.95	.950	.99	.99	**.987**	.98	.98	**.978**	.99	.00	**.995**	.93	.93	.927	–	–	–	.97	.97	**.973**
	SJTU–NICT	.93	.93	**.933**	.95	.95	.949	.96	.95	.955	.99	.99	.993	.93	.92	.924	–	–	–	.96	.95	.955
		.97	.96	**.965**	.93	.93	.933	.94	.94	.944	.99	.99	.990	.93	.93	.933	–	–	–	.95	.95	.949
	HIT-SCIR	.93	.93	.926	.93	.93	.930	.95	.95	.953	.99	.99	.993	.93	.92	**.925**	–	–	–	.95	.95	.951
		.95	.95	.950	.93	.93	.928	.95	.95	.947	.99	.99	.990	.94	.94	**.935**	–	–	–	.95	.95	.950
	SUDA–Alibaba	.91	.91	.911	.90	.91	.903	.91	.92	.915	.97	.99	.982	.89	.91	.898	–	–	–	.91	.93	.923
		.91	.88	.893	.86	.89	.872	.88	.91	.895	.96	.99	.979	.88	.92	.896	–	–	–	.89	.92	.907
	Peking	.93	.93	.927	.92	.91	.915	.95	.94	.945	.99	.99	.991	.92	.92	.924	–	–	–	.94	.94	.944
		.96	.96	.960	.88	.88	.882	.91	.92	.914	.99	.99	.989	.92	.92	.921	–	–	–	.92	.93	.925
EDS	ERG	.90	.90	**.902**	.97	.96	**.965**	.96	.96	**.960**	.96	.96	**.963**	.93	.93	.929	–	–	–	.95	.95	**.952**
		.93	.93	.930	.96	.97	**.964**	.85	.88	**.863**	.98	.99	**.983**	.93	.94	**.932**	–	–	–	.96	.96	**.959**
	SUDA–Alibaba	.90	.90	.899	.91	.91	.912	.89	.91	.897	.95	.95	.949	.90	.90	.897	–	–	–	.92	.92	.918
		.94	.94	**.940**	.91	.92	.913	.72	.84	.778	.95	.96	.953	.91	.91	.911	–	–	–	.92	.93	.925
	HIT-SCIR	.88	.82	.852	.90	.89	.894	.89	.91	.895	.95	.94	.943	.89	.88	.888	–	–	–	.91	.90	.907
		.92	.91	.915	.85	.86	.854	.76	.88	.815	.95	.96	.950	.89	.89	.890	–	–	–	.89	.90	.898
	SJTU–NICT	.91	.85	.877	.93	.86	.894	.79	.76	.775	.97	.90	.934	.95	.82	.878	–	–	–	.95	.86	.899
		.97	.89	.927	.93	.88	.904	.27	.24	.255	.97	.93	.949	.94	.86	.894	–	–	–	.94	.88	.912
	Peking	.83	.83	.829	.95	.94	.946	.91	.96	.936	.96	.96	.961	.94	.93	**.933**	–	–	–	.95	.94	.945
		.89	.89	.890	.91	.92	.918	.49	.88	.629	.95	.96	.959	.92	.92	.918	–	–	–	.92	.93	.928

Table 1: MRP results for DM (top) and EDS (bottom), with precision (P), recall (R), and F_1 for different types of graph components: top nodes, node labels, other node properties, anchoring into the surface string, labeled edges, and all of these combined (neither DM nor EDS use edge attributes). Best F_1 scores in each category are in bold. The pair of rows per submission indicate the full MRP evaluation data vs. the 100-sentence *Little Prince* subset.

guarantees n-best enumeration from the parse forest in globally correct rank order. At its core, this is a specialized search procedure on a weighted and–or graph (the forest), where for packed (i.e. disjunctive) nodes local contexts of optimization are established on demand. Although worst-case complexity for both forest construction and unpacking is in principle exponential, parsing times (for small values of n) with the ERG in practice mostly grow polynomially in input length. For example, parser throughput for the sentences from the *Little Prince* subset of the MRP evaluation data (see Oepen et al., 2019) averages at two sentences per second, whereas average parse times for the much longer 100-sentence MRP sample of WSJ text lie around four seconds per sentence.

For parsing the MRP evaluation data, we applied ERG release 1214 with its bundled WSJ parse ranking model, which uses the feature configuration of Zhang et al. (2007) and was trained on Sections 00–20 of the Redwoods Ninth Growth using the Maximum Entropy estimation toolkit of Malouf (2002). We use the LOGON distribution as of August 2019 to parse in one-best mode the 'raw' strings for the MRP evaluation data whose target representations were indicated as DM or EDS. The resulting HPSG derivations each uniquely determine an ERS meaning representation in underspecified logic, which we subsequently convert to EDS and DM.[2]

Given the formal nature of this process, the resulting graphs are guaranteed to reflect the composition algebra of the ERG, recursively building larger fragments of meaning from smaller parts.

4 Experimental Results

Parsing accuracies for PET and the ERG are summarized in Table 1, for both the DM (top) and EDS (bottom) evaluation graphs. The table compares ERG parsing results to a selection of 'real' submissions to the shared task, viz. the top performers within each framework and for the task

[2]The ERS-to-EDS converter of Oepen and Lønning (2006) is part of the LOGON distribution, as is the converter of Ivanova et al. (2012) for further simplification to bi-lexical DM. Exact command-line incantations for all tools and their parameterization are specified as part of the submission archive in the MRP 2019 data release.

overall: HIT-SCIR (Che et al., 2019), Peking (Chen et al., 2019)[3], SJTU–NICT (Bai and Zhao, 2019), and SUDA–Alibaba (Zhang et al., 2019). In contrast to the ERG parser, all of these systems are purely *data-driven*, in the sense that they do not incorporate manually curated linguistic knowledge (beyond finite-state tokenization rules, maybe) but rather learn all their parameters exclusively from the shared task training data.

By and large, the data-driven parsers are competitive to the ERG, in particular the SJTU–NICT and HIT-SCIR systems for DM, and the Peking parser for EDS. For some structural types of graph components (tops and edges), the ERG is in fact outperformed by some submissions, whereas it holds at times commanding leads on node-local types of information, e.g. labels, properties, and achors. It could be argued that comparison for some of these graph components favors the ERG, seeing as it embodies the exact principles of deriving these values that were used in creating the Redwoods annotations. However, for DM at least, node labels are essentially lemmas, and it is prima facie surprising that none of the data-driven parsers succeeds very well in replicating ERG-style lemmatization.

Likewise, anchoring for EDS is a many-to-many relation between graph nodes and (arbitrary) input sub-strings, where one can speculate that at least some of the conventions used in the ERG may be linguistically idiosyncratic. Inasmuch as that may (or may not) be the case, the Peking parser shows anchoring accuracies comparable to the ERG.

The *Little Prince* subset of the evaluation data is comprised of much shorter sentences, and observed accuracies for some types of graph components may appear to correlate with input complexity, notably top node and (to a lesser) degree edge prediction. At the same time, the novelistic style of this subset most likely makes it least similar to the WSJ-derived training data for the data-driven parsers, hence some submissions can seem to suffer from detrimental cross-domain effects.

5 Reflections

As long-term co-developers of the ERG and its PET parser, we are impressed by the overall performance levels of the data-driven submissions to the MRP 2019 shared task. We hope to conduct more contrastive error-analysis, possibly in collaboration with other parser developers, to further isolate effects of domain variation, for example, and generally gauge the contributions (if any) of the explicit body of linguistic knowledge in the ERG.

[3]The Peking submission is not considered in the *primary* ranking of the official shared task results, because the team inadvertently used tokenization training data beyond the 'white-listed' resources for task participants

References

Hongxiao Bai and Hai Zhao. 2019. SJTU at MRP 2019: A transition-based multi-task parser for cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 86 – 94, Hong Kong, China.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation. On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239 – 249, London, UK.

Ulrich Callmeier. 2002. Preprocessing and encoding techniques in PET. In *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing*, pages 127 – 140, Stanford, CA. CSLI Publications.

John Carroll and Stephan Oepen. 2005. High-efficiency realization for a wide-coverage unification grammar. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing*, pages 165 – 176, Jeju, Korea.

David Carter. 1997. The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 9 – 15, Madrid, Spain.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76 – 85, Hong Kong, China.

Yufei Chen, Yajie Ye, and Weiwei Sun. 2019. Peking at MRP 2019: Factorization- and composition-based parsing for Elementary Dependency Structures. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 166 – 176, Hong Kong, China.

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.

Gregor Erbach. 1991. A flexible parser for a linguistic development environment. In Otthein Herzog and

Claus-Rainer Rollinger, editors, *Text Understanding in LILOG*, pages 74 – 87. Springer, Berlin, Germany.

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

Dan Flickinger, Emily M. Bender, and Stephan Oepen. 2014. Towards an encyclopedia of compositional semantics. Documenting the interface of the English Resource Grammar. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 875 – 881, Reykjavik, Iceland.

Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. Sustainable development and refinement of complex linguistic annotations at scale. In *Handbook of Linguistic Annotation*, pages 353 – 377, Dordrecht, The Netherlands. Springer.

Jean Mark Gawron, Jonathan King, John Lamping, Egon Loebner, E. Anne Paulson, Geoffrey K. Pullum, Ivan A. Sag, and Thomas Wasow. 1982. Processing English with a Generalized Phrase Structure Grammar. In *Proceedings of the 20th Meeting of the Association for Computational Linguistics*, pages 74 – 81, Toronto, Canada.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop*, pages 2 – 11, Jeju, Republic of Korea.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic 'unification-based' grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics*, pages 535 – 541, College Park, MD, USA.

Martin Kay. 1986. Algorithm schemata and data structures in syntactic processing. In In Barbara J. Grosz, Karen Spärck Jones, and Bonnie Lynn Weber, editors, *Readings in Natural Language Processing*, pages 35 – 70. Morgan Kaufmann, San Francisco, CA, USA.

Bernd Kiefer, Hans-Ulrich Krieger, John Carroll, and Robert Malouf. 1999. A bag of useful techniques for efficient and robust parsing. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics*, pages 473 – 480, College Park, MD, USA.

Jan Tore Lønning and Stephan Oepen. 2006. Re-usable tools for precision machine translation. In *Proceedings of the COLING|ACL 2006 Interactive Presentation Sessions*, pages 53 – 56, Sydney, Australia.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 49 – 55, Taipei, Taiwan.

Robert Malouf, John Carroll, and Ann Copestake. 2000. Efficient feature structure operations without compilation. *Natural Language Engineering*, 6(1):29 – 46.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics*, 19:313 – 330.

Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdeňka Urešová. 2019. MRP 2019: Cross-framework Meaning Representation Parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1 – 27, Hong Kong, China.

Stephan Oepen and Ulrich Callmeier. 2000. Measure for measure: Parser cross-fertilization. Towards increased component comparability and exchange. In *Proceedings of the 6th International Conference on Parsing Technologies*, pages 183 – 194, Trento, Italy.

Stephan Oepen and John Carroll. 2000. Ambiguity packing in constraint-based parsing. Practical results. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 162 – 169, Seattle, WA, USA.

Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575 – 596.

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

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press, Chicago, USA.

Hideto Tomabechi. 1995. Design of efficient unification for natural language. *Journal of Natural Language Processing*, 2(2):23 – 58.

Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. 2005. Stochastic HPSG Parse Disambiguation using the Redwoods Corpus. *Research on Language and Computation*, 3:83 – 105.

Yi Zhang, Stephan Oepen, and John Carroll. 2007. Efficiency in unification-based n-best parsing. In *Proceedings of the 10th International Conference on Parsing Technologies*, pages 48 – 59, Prague, Czech Republic.

Yue Zhang, Wei Jiang, Qingrong Xia, Junjie Cao, Rui Wang, Zhenghua Li, and Min Zhang. 2019. Suda–alibaba at MRP 2019: Graph-based models with BERT. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 149 – 157, Hong Kong, China.