On the Generation of Abstract Meaning Representations using Polynomial-Time Parsable Hyperedge Replacement Grammars

Anna Jonsson

Dept. of Computing Science, Umeå University
901 87, Umeå
aj@cs.umu.se

Abstract

We summarise an evaluation of the usefulness of two types of hyperedge replacement grammar (HRG) for the generation of abstract meaning representations (AMRs). For both types, polynomial recognition algorithms exist, and the evaluation is made with respect to a restricted AMR domain. We conclude that hyperedge replacement is not in itself powerful enough to capture all of the aspects of AMRs in general; however, various polynomial-time recognisable extensions of HRG should be attempted.

1. Introduction

Natural language is often ambiguous and hence not suitable for direct algorithmic processing. Therefore, a meaning representation with well-defined semantics is needed. A formalism that has been proposed for the representation of the semantics of a natural language sentence is the abstract meaning representation (AMR) (Langkilde and Knight, 1998; Banarescu et al., 2013); it has the form of a directed, rooted, acyclic, node and edge labelled graph in which the nodes represent concepts and the edges give the relations between the concepts. The parsing of sentences into AMRs has been extensively studied using various methods (Flanigan et al., 2014; Wang et al., 2015b; Werling et al., 2015; Wang et al., 2015a; Peng et al., 2015; Pust et al., 2015; Artzi et al., 2015). Since none of the methods achieves perfect results, it would be useful to combine them with an algorithm that checks that the resulting AMRs are at least structurally correct (e.g., that a concept referring to a person is not used as a verb concept).

Structural properties of a graph can be expressed using graph generating or recognising devices such as graph grammars or automata; here, we focus on graph grammars. If all of the structural properties required of a correct AMR over a certain domain can be expressed in a single graph grammar, then the grammar can be used for verifying the correctness of any input AMR over that domain. The problem of finding a derivation for an input graph with respect to a given graph grammar (if such a derivation exists) is known as the *parsing problem* for graph grammars. However, there must exist a polynomial-time parsing algorithm for the type of graph grammars considered if using it for describing correct AMRs is to be feasible in practice.

Hyperedge replacement grammar (HRG) (Drewes et al., 1997) is a graph grammar type that makes use of *hyperedge replacement*, i.e., the replacement of a hyperedge (an edge connecting an arbitrary number of nodes) in a hypergraph (a graph containing hyperedges) with another hypergraph. AMRs can easily be represented as hypergraphs since hypergraphs are simply a generalisation of graphs. It has previously been shown that HRGs can generate NP-complete graph languages, hence parsing for HRGs seems to require exponential time in general.

In this short paper, we summarise a study in which two HRG based formalisms with known polynomial-time parsing algorithms are evaluated with respect to AMR generation within the restricted domain of the boy-girl example originating in (Braune et al., 2014). In other words, we compare the ability of HRGs to generate AMRs over the concepts `boy`, `girl`, `want` and `believe` and the relations `arg0` and `arg1` that have at most one occurrence each of `boy` and `girl` (here referred to as *boy-girl AMRs*). In (Chiang et al., 2013), it is claimed that the above described boy-girl AMR language can be generated using hyperedge replacement; this has however not been shown.

The first type of HRG evaluated is predictive top-down (PTD) parsable grammar (Drewes et al., 2015); the second type is restricted directed acyclic graph (rDAG) grammar (Björklund et al., 2016). AMR generation has been used to motivate both types, and they both allow for quadratic parsing; this study investigates their practical feasibility. In this paper, we argue that rDAG grammar is better suited for boy-girl AMR generation, but that neither of the formalisms is well-suited for AMR generation in general because they are unable to generate AMRs that are minimal with respect to the number of nodes (i.e., as compact as possible).

This paper is based on the author's student paper (Jonsson, 2016a) and master's thesis (Jonsson, 2016b).

2. Predictive top-down parsable grammar

Predictive top-down (PTD) parsable grammars (Drewes et al., 2015) are grammars that pass a certain, extensive analysis. Put simply, the analysis checks that the grammar is such that one can always know which rule to match next throughout a top-down parsing procedure. If the analysis succeeds for a grammar, a parser for that particular grammar that runs in at most quadratic time is created. However, the analysis itself is not guaranteed to be performed in polynomial time.

For the first part of the evaluation, we use an implementation that performs both the analysis and the parsing with a PTD parsable boy-girl AMR generating grammar (both provided by Drewes et al.) in order to attempt to parse all of the boy-girl AMRs in the corpus presented in (Braune et al., 2014) with respect to the aforementioned grammar.

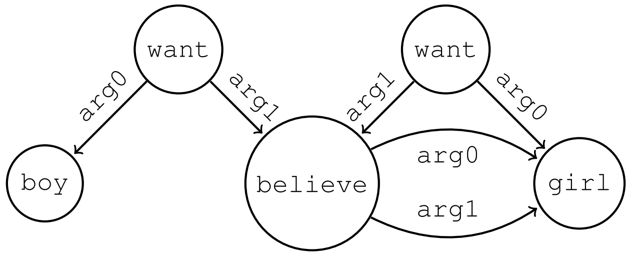

Figure 1: An AMR representing the meaning "The boy wants the girl to believe herself and the girl wants to believe herself" that is minimal with respect to the number of nodes, but cannot be generated by the PTD parsable boy-girl grammar.

Out of the 10 000 AMRs in the corpus, only 8 880 turned out to be generatable by the grammar. All of the AMRs that could not be generated contain at least one leaf node labelled `want` or `believe`. An attempt to generalise the grammar in order to enable it to generate the missing AMRs resulted in a grammar for which the PTD analysis failed as it was unable to distinguish between the cases where a `want` or `believe` node is a leaf and where it is not. In order to see why this is the case, we would need a much more detailed description of the PTD analysis, which is out of scope of this paper.

As a second and final step, we characterise the language generated by the PTD parsable boy-girl grammar and and prove that the characterisation is correct. Although these parts are left out here, it should be mentioned that the characterisation makes it clear that the grammar only generates graphs in which only the leaf nodes can have several incoming edges. In particular, this means that the grammar does not necessarily generate the AMR that is minimal with respect to the number of nodes for a particular meaning, but might instead generate a larger AMR to represent the meaning; see Figure 1 for a minimal AMR and Figure 2 for the AMR that is actually generated for the same meaning. In Figure 2, we see that both `believe` concepts (which are in themselves abstract concepts, and therefore indistinguishable as opposed to e.g. two `boy` concepts representing two different persons) have exactly the same outgoing edges ending at the same `girl` concept. In other words, the two subgraphs rooted at the `believe` concepts are equal, and can therefore be merged, as seen in Figure 1. The problem of not being able to generate minimal AMRs for all possible meanings over the boy-girl domain seems unavoidable (rather than being a weakness of this particular grammar).

3. Restricted DAG grammar

Restricted DAG (rDAG) grammars (Björklund et al., 2016) and their polynomial-time parsing algorithm were developed with AMRs in mind. An HRG is an rDAG grammar if its rules satisfy a few easily verifiable structural conditions. Thus, it can easily be seen (or automatically checked without an expensive analysis) whether a given HRG is indeed an rDAG grammar. The rDAG grammar parsing algorithm works bottom-up, and it does not need to know what rule to match next, but instead exploits that the few allowed rule formats are all known. The algorithm works very much like the well-known forward algorithm for regular languages. An rDAG boy-girl grammar can be found in (Björklund et al., 2015); it uses the same graph-building mechanisms and therefore generates the same connected boy-girl AMRs as the PTD parsable one. Thus, the AMRs generated in this case are also not necessarily minimal with respect to the number of nodes. The difference is, however, that the rDAG boy-girl grammar can in fact be extended to include the concepts `want` and `believe` as leaf nodes while continuing to be an rDAG grammar.

4. AMR validity and minimality

Although no formal proofs have been attempted so far, we believe that the observations made above are not just due to weaknesses of the particular constructions. Neither HRG formalism seems to be able to generate AMRs that are minimal with respect to the number of nodes for every meaning, i.e., they seemingly cannot handle the generation of minimal AMRs in general. Since the entire idea of AMRs is based on the desire to have a unique semantic representation for each possible meaning, all different-sized AMRs with the same meaning cannot be valid. Thus, one must be picked, and from a computational point of view, it makes sense for it to be a minimal one since this is the only choice for which an AMR can potentially be uniquely determined. (We do not yet know if there exists a unique minimal AMR for every meaning.)

Should we want to generate minimal AMRs, each concept generated by any rule application needs to be kept track of from the point it is generated until the end of the derivation. This is due to the necessity of leaving open the possibility of adding edges connecting it to another concept generated at a later stage throughout the derivation. To achieve this using hyperedge replacement, an infinite number of rules would probably be needed since an AMR can be made indefinitely large, and it would take at least one extra rule for each number of nodes the intermediate graphs will have during the derivation. Thus, not only may PTD parsable HRGs and rDAG grammars be unable to generate minimal AMRs, but this is likely to be a limitation of hyperedge replacement in general.

5. Conclusion and future work

We have seen that the rDAG boy-girl grammar can generate boy-girl AMRs with `want` and `believe` as leaf node labels, which the PTD parsable one cannot (conjecturing that this is a general limitation rather than a flaw of the particular construction used). Moreover, we know that the PTD parsing algorithm requires an initial analysis which is not guaranteed to be done in polynomial time. Thus, we conclude that rDAG grammar is better suited for boy-girl AMR generation. However, neither PTD parsable grammar nor restricted DAG grammar can be used for generating minimal AMRs, and we have motivated why it is likely that this is due to the hyperedge replacement mechanism not being powerful enough. Therefore, we want to attempt the usage of extensions of the two HRG formalisms seen here as well as contextual HRG (Drewes and Hoffmann, 2015)

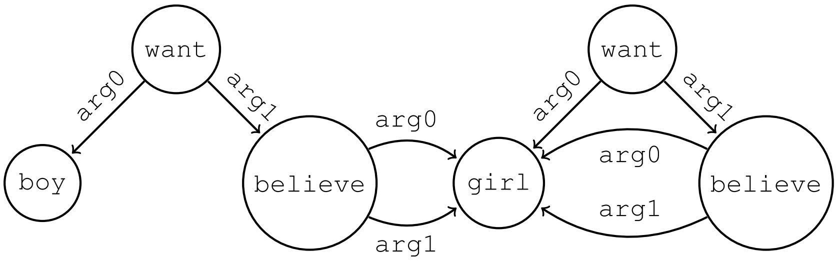

Figure 2: An AMR representing the meaning "The boy wants the girl to believe herself and the girl wants to believe herself" that is not minimal with respect to the number of nodes, but can be generated by the PTD parsable boy-girl grammar.

for AMR generation. Furthermore, the difficulties emerging when removing the limitation of this study posed by the small AMR domain should be investigated.

As one of the reviewers pointed out, the construction of a kind of pumping lemma would also be of interest; the lemma would ensure that if a grammar can generate a minimal AMR for a certain meaning, it can generate all larger AMRs that represent the same meaning.

An important question regarding the PTD parsable grammar that should also be answered is how to construct a grammar that we know will pass the PTD analysis. Having such knowledge would substantially facilitate the usage of PTD parsable grammar (and its possible extensions).

Acknowledgements

The author would like to thank Johanna Björklund, Frank Drewes and the reviewers for their valuable input.

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710. Association for Computational Linguistics.

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*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

Henrik Björklund, Frank Drewes, and Petter Ericson. 2015. Between a rock and a hard place — parsing for hyperedge replacement DAG grammars. Technical Report UMINF 15.13, Umeå University.

Henrik Björklund, Frank Drewes, and Petter Ericson. 2016. Between a rock and a hard place — parsing for hyperege replacement dag grammars. In *10th International Conference on Language and Automata Theory and Applications (LATA 2016)*. Springer Publishing Company.

Fabienne Braune, Daniel Bauer, and Kevin Knight. 2014. Mapping between English strings and reentrant semantic graphs. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4493–4498.

David Chiang, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, Bevan Jones, and Kevin Knight. 2013. Parsing graphs with hyperedge replacement grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 924–932, Sofia, Bulgaria, August. Association for Computational Linguistics.

Frank Drewes and Berthold Hoffmann. 2015. Contextual hyperedge replacement. *Acta Informatica*, 52(6):497–524.

Frank Drewes, Hans-Jörg Kreowski, and Annegret Habel. 1997. Hyperedge replacement graph grammars. In Grzegorz Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation*, pages 95–162. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Frank Drewes, Berthold Hoffmann, and Mark Minas. 2015. Predictive top-down parsing for hyperedge replacement grammars. In *Proceedings of the 8th International Conference on Graph Transformation*, pages 19–34.

Jeffrey Flanigan, Sam Thomson, Jaime G. Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1426–1436.

Anna Jonsson. 2016a. Context-free graph grammars for recognition of abstract meaning representations. In Suna Bensch and Thomas Hellström, editors, *Proceedings of Umeå's 20th student conference in computing science: USCCS*, pages 69–81. Umeå universitet, January.

Anna Jonsson. 2016b. Generation of abstract meaning representations by hyperedge replacement grammars – a case study. Master's thesis, June. Available at http://www8.cs.umu.se/education/examina/Rapporter/AnnaJonsson-final.pdf.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 704–710, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaochang Peng, Linfeng Song, and Daniel Gildea. 2015. A synchronous hyperedge replacement grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Parsing English into abstract meaning representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1143–1154, Lisbon, Portugal, September. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 857–862.

Chuan Wang, Nianwen Xue, Sameer Pradhan, and Sameer Pradhan. 2015b. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.

Keenon Werling, Gabor Angeli, and Christopher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. *CoRR*.