

Semantic Composition with PSHRG for Derivation Tree Reconstruction from Graph-Based Meaning Representations

Chun Hei Lo Wai Lam Hong Cheng

Department of Systems Engineering and Engineering Management

The Chinese University of Hong Kong

{chlo, wlam, hcheng}@se.cuhk.edu.hk

Abstract

We introduce a data-driven approach to generating derivation trees from meaning representation graphs with probabilistic synchronous hyperedge replacement grammar (PSHRG). SHRG has been used to produce meaning representation graphs from texts and syntax trees, but little is known about its viability on the reverse. In particular, we experiment on Dependency Minimal Recursion Semantics (DMRS) and adapt PSHRG as a formalism that approximates the semantic composition of DMRS graphs and simultaneously recovers the derivations that license the DMRS graphs. Consistent results are obtained as evaluated on a collection of annotated corpora. This work reveals the ability of PSHRG in formalizing a syntax-semantics interface, modelling compositional graph-to-tree translations, and channelling explainability to surface realization.

1 Introduction

General graph-based meaning representations (MRs) that model sentence-level semantics aim to provide interpretable intermediate representations that are application- and domain-independent (Koller et al., 2019). Recently, graph grammars and algebras that formalize semantic constructions were introduced to MR processing (for example: (Koller, 2015; Drewes and Jonsson, 2017; Groschwitz et al., 2017; Chen et al., 2018; Groschwitz et al., 2018; Lindemann et al., 2019; Donatelli et al., 2019; Chen and Sun, 2020)). These formal grammars bridge between linguistic assumptions and data-driven parsing, and offer the benefit of cross-framework adaptability. For instance, the Apply-Modify algebra was adopted in parsing across 5 MR frameworks (Oepen et al., 2019) (Lindemann et al., 2019; Donatelli et al., 2019). Another formalism that was adopted in generating semantic graphs from syntax trees is synchronous hyperedge replacement grammar (SHRG) (Peng et al., 2015; Chen et al., 2018; Chen and Sun, 2020).

The use of SHRG in recovering syntax trees from MRs has however received scant research coverage. Empirical results of PSHRG’s application are limited to Jones et al. (2012)’s work in semantic-based machine translation.

An immediate application of MR-to-tree parsing is surface realization. Previous data-driven approaches to it include rule-based (Flanigan et al., 2016; Song et al., 2017; Horvat, 2017; Ye et al., 2018) and neural methods (Song et al., 2018; Damonte and Cohen, 2019; Hajdik et al., 2019). All these methods do not generate syntactic analyses. In contrast, the Answer Constraint Engine (ACE; Carroll et al., 1999; Carroll and Oepen, 2005; Veldal and Oepen, 2006), an HPSG grammar-based parser, generates both derivations and sentences from Minimal Recursion Semantics (MRS; Copestake et al., 2005). If we can induce an SHRG from data, MRs can be translated into derivation trees without relying on a hand-engineered grammar, and natural language texts can be obtained by realizing the terminals. Combining the strengths of rule-based systems and the data-driven paradigm, such an approach gives both linguistically-informed realization processes and explainable results, removes syntactic ambiguities that would otherwise exist in flattened surface strings, and provides potential usage for downstream tasks such as chunking.

Among the different MR frameworks, we investigate Dependency Minimal Recursion Semantics (DMRS; Copestake, 2009). DMRS are directed graphs derived losslessly from MRS, whereas an MRS structure with respect to a reading of an English sentence is composed along with a derivation tree using the English Resource Grammar (ERG; Flickinger, 2000, 2011), a broad-coverage hand-engineered HPSG grammar of English. DMRS encodes logical formulae with underspecified scopes (for an introduction, see: Copestake, 2009). Fig. 1 shows an example of an ERG analysis.

Copestake et al. (2001) gave a compositional se-

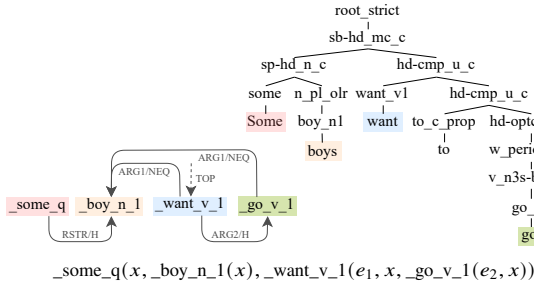


Figure 1: A DMRS (left), an HPSG derivation (right) that licenses the DMRS by an ERG analysis of the sentence *Some boys want to go*, and a scope-resolved representation of the DMRS with variables (bottom). Each DMRS node is aligned to the surface string as indicated by the colours. In the derivation, each preterminal is a lexical item, e.g., `_go_v_l`, and each nonterminal above is labelled by a lexical or syntactic rule, e.g., `hd-cmp_u_c` denotes the head-complement construction. In a DMRS, each node is either a predicate corresponding to a lexeme (e.g., `_boy_n_l`) or a nonlexical predicate (e.g., `loc_nonsp`) (see Fig. 4). For ease of exposition, node attributes (e.g., number and tense) are not shown. The primary edge labels (e.g., ARG1) denote argument relationship, the secondary edge labels (e.g., /NEQ and /H) encode constraints on scopal relationships, and TOP specifies the top scopal non-quantifier node. Based on the scopal information in the DMRS, the bottom scope is the only scope possible.

mantic algebra on the constructions of MRS, where the introduced constraints allow the semantics of complex sentences to be derived from simple composition rules. The compositionality exhibited in the MRS (for a discussion, see: Bender et al., 2015) is not obvious in some other MRs, e.g., Abstract Meaning Representation (AMR; Banarescu et al., 2013) (Bender et al., 2015), and we suggest that HRG be particularly suitable to simulate the DMRS algebra if equipped with adequate adaptations.

In this paper, we shed light on the applicability of a succinct grammar formalism in approximating the semantic compositions of a graph-based MR. Essentially, we capture the syntax-semantic interface of the ERG by inducing a PSHRG from an annotated treebank to provide generative models for both DMRS graphs and derivation trees. With the induced PSHRG, derivation trees that license the DMRS graphs can be reconstructed through graph parsing. We describe the procedures and the relevant adaptations involved, from PSHRG induction, parsing to generating derivation trees. Finally, we present the empirical results on derivation trees and surface strings reconstruction from DMRS graphs under different configurations.

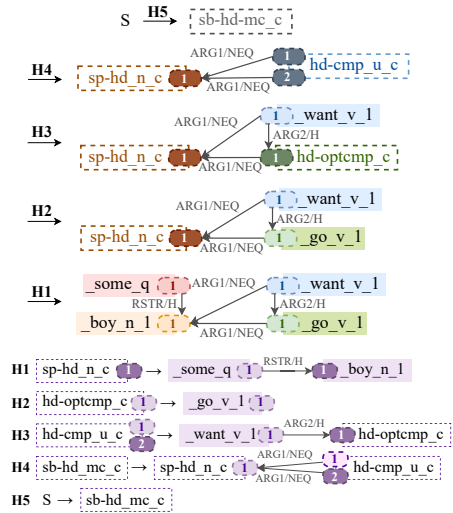


Figure 2: Derivation of an HRG for the hypergraph of DMRS in Fig. 1 using five productions **H5** to **H1** in order. A labelled rectangle represents a labelled hyperedge, where a hollow one represents a nonterminal. The numbered rounded rectangles attached to each hyperedge denote the ordered collection of nodes the hyperedge connects to, each of which corresponds to a DMRS node. A labelled arrow is a terminal hyperedge that connects to exactly two nodes and that corresponds to a DMRS edge. In an HRG production $A \rightarrow R$, each external node of R is mapped to a node of the same colour in A . **H5** rewrites the start symbol to a hyperedge that connects to one node. Note that all nodes of **H4** are non-external nodes of the hypergraph fragment.

2 Probabilistic Synchronous Hyperedge Replacement Grammar

Hyperedge replacement grammar (HRG) is a context-free rewriting formalism for generating graphs (Drewes et al., 1997). A synchronous HRG (SHRG) defines mappings between languages of graphs, which in our context are an HRG and a context-free grammar (CFG) for strings. We give the definitions in this section.

Hypergraph and Hypergraph Fragments. A hypergraph is specified by a tuple $H = \langle V, E, l \rangle$, where V is a finite set of nodes, $E \subseteq V^+$ is a finite set of hyperedges, each of which connects one or more distinct nodes, $l : E \rightarrow L$ assigns a label from the finite set L to each hyperedge. A hypergraph fragment is a tuple $R = \langle V, E, l, X \rangle$, where $\langle V, E, l \rangle$ is a hypergraph and $X \in V^+$ is an ordered list of distinct nodes called the external nodes.

HRG. An HRG is a tuple $G = \langle N, T, P, S \rangle$, where N and T are disjoint finite sets of nonterminal and terminal symbols respectively, P is a

finite set of productions of the form $A \rightarrow R$, where $A \in N$ and R is a hypergraph fragment where hyperedge labels are over $N \cup T$, and $S \in N$ is the start symbol. In a step of rewriting a hyperedge e by a production $A \rightarrow R$, e is replaced with a copy of R by identifying each of the nodes connected by e with a distinct external node of R , whose mapping is specified in the production. Fig. 2 illustrates the HRG rewriting process.

Synchronous SHRG. An SHRG is a tuple $G = \langle N, T, T', P, S \rangle$, where N is a finite set of nonterminal symbols in both the CFG and the HRG, T and T' are finite sets of terminal symbols in HRG and CFG respectively, $S \in N$ is the start symbol, and P is a finite set of productions of the form $A \rightarrow \langle R, R', \sim \rangle$, where R is a hypergraph fragment with hyperedge labels over $N \cup T$, R' is a symbol sequence over $N \cup T'$, and \sim is a bijection between the nonterminal hyperedges of the same labels in R and R' . When applying a production $A \rightarrow \langle R, R', \sim \rangle$, $A \rightarrow R$ rewrites the graph as described in the HRG and the synchronous operation on the CFG counterpart is a string rewrite by $A \rightarrow R'$ (see Fig. 3).

Probabilistic SHRG. A probabilistic SHRG is obtained by assigning a constant probability to each production in the SHRG, where probabilities of the productions that rewrite the same nonterminal add up to one. In this work, the probability of $A \rightarrow \langle R, R', \sim \rangle$ is simply modelled as the fraction of times it appears among all $A \rightarrow *$ in the training data. The probability of a derivation is the product of the probabilities of the context-free SHRG productions applied.

3 PSHRG Induction and Parsing

In this section, we describe how a PSHRG is induced from training data and how a derivation tree is reconstructed from a DMRS of the test data with the induced grammar. We also describe two methods for modelling the semantics of lexical items.

DMRS as Hypergraph. We first establish the connections between DMRS and HRG. A DMRS graph is modelled as a hypergraph H with terminal hyperedges. In H , each terminal hyperedge corresponds to a DMRS node or edge: the former can connect to an arbitrary number of nodes in H , and the latter connects to only two nodes in H .

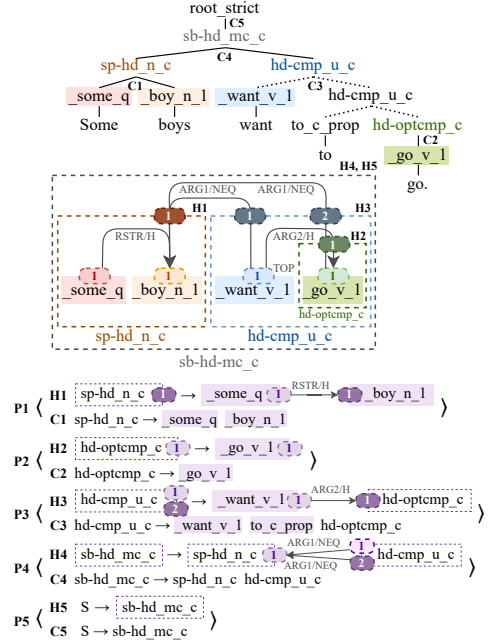


Figure 3: Illustration of the derivation, induction and parsing processes of an SHRG for the analysis in Fig. 1. At the top is the normalized derivation (see §4.1). Each SHRG production comprises one HRG and one CFG rule. During SHRG derivation, nonterminal hyperedges are rewritten by **H5** to **H1** and strings by **C5** to **C1** sequentially. During SHRG induction, synchronous rules **P1** to **P5** are extracted in order. During SHRG parsing, **H1** to **H5** are recognized in order, and **C1** to **C5** are applied synchronously. **P3** extracts (recovers) the subtree connected by dotted lines that contains the semantically empty word *to* during induction (parsing).

3.1 PSHRG Induction

We describe how to induce a PSHRG from pairs of aligned trees and graphs. The PSHRG induction procedure generally follows Chen et al. (2018)’s SHRG extraction algorithm, which operates based on the surface string alignment information between DMRS graphs and their derivation trees (for expositions, see: Chen et al., 2018). Fig. 3 illustrates the grammar induction process. For each nonterminal in the tree, i.e., a node labelled by an ERG syntactic construction (a label of the form $*_c$, e.g., $hd-cmp_u_c$), a production $A \rightarrow \langle R, R', \sim \rangle$ is extracted, where A , R and R' are the ERG syntactic construction, connected DMRS hypergraph fragment and daughter ERG rule(s) respectively. For a binary ERG construction, if any of its daughters is a semantically empty terminal (e.g., for the lower $hd-cmp_u_c$ in Fig. 3), no productions are extracted (discussed in §3.3.3). The DMRS hypergraph fragment specified by R is then rewritten to a

nonterminal hyperedge labelled by A . We perform the same rule extraction procedure for each training instance to induce an SHRG from the training data, thus a PSHRG when we consider the frequency information.

3.2 PSHRG Parsing

To recover the derivation tree of a DMRS, we can translate the semantic compositions to the corresponding syntactic operations by parsing the DMRS with the induced PSHRG (see Fig. 3). We aim at recognizing the best derivation according to a PSHRG model, which requires exact graph parsing. Chiang et al. (2013); Groschwitz et al. (2015); Ye and Sun (2020) studied HRG parsing and proposed various techniques on improving the efficiency, but no evaluation on accuracy is performed on the parsed results with respect to a gold-standard grammar. Although efficient algorithms are developed for HRG parsing, existing parsers do not provide convenient adaptations to the extensions introduced in this work. Rather than efficiency, our work focuses on the correctness of derivations as measured against the original derivations that license each DMRS. Therefore, we implement a parser that returns the best PSHRG derivation of DMRS via bottom-up passive chart parsing (for details of the parsing algorithms, see Appendix A).

3.3 PSHRG Adaptations

We introduce two adaptations to align PSHRG with the semantics introduced by the ERG lexical items.

3.3.1 Semantics of Lexical Items

Complex semantics. While most lexical items introduce just one DMRS predicate each, some introduce more complex semantics. As an example, Fig. 4 shows that *somebody* provides both the *_some_q* quantifier and the *person* predicate. Therefore, the R.H.S. hypergraph fragment of an SHRG production is not confined to a hypergraph fragment with one or two terminal hyperedges, but one with more than two terminal hyperedges that corresponds to a connected DMRS subgraph.

Empty semantics. There are semantically empty lexical items that do not contribute predicates to the DMRS, e.g., auxiliary verbs and particles. This poses another challenge for derivation reconstruction because the syntactic properties of these lexical items are highly language-dependent, yet they are not captured by general semantic representations as they are not semantically functional.

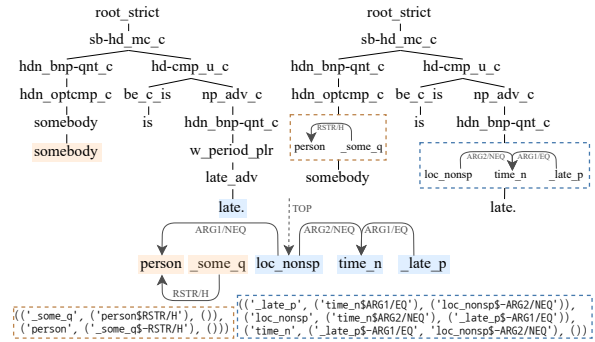


Figure 4: Canonization of two small subgraphs. The top left tree is the original derivation, where the terminals are aligned to DMRS nodes as indicated by the colour. The top right tree is the normalized derivation, where two preterminals are replaced by their DMRS subgraphs, each represented by its canonical form at the bottom.

3.3.2 Canonization of Small Subgraphs

To recognize complex semantics, we borrow the idea of the graph canonization method described by Horvat (2017) that isomorphic DMRS subgraphs can be identified by comparing if their canonical representations are the same. Graph canonization is achieved in two steps: first, each node in the DMRS subgraph is given a canonical node representation by encoding its 1- and 2-hop neighbours; then the final canonical form is obtained by concatenating the sorted node representations based on a canonical ordering. Most subgraphs introduced contain fewer than seven DMRS nodes, for which the canonization method is sound (Horvat, 2017). Fig. 4 exemplifies the idea.

During grammar induction, the canonical forms of all small subgraphs that correspond to ERG lexical items are extracted from the training data. Then, given a DMRS from the test data, we first identify its subgraphs that are isomorphic to any of the extracted ones before parsing. This is achieved by first enumerating all small subgraphs from the DMRS, then computing the canonical form for each of them, and finally comparing the canonical forms with those of the subgraphs extracted. The process can be sped up by computing the canonical form of a subgraph only if its collection of DMRS predicates is present in the set of those extracted from the training data.

3.3.3 Semantically Empty Lexical Items

We devise a semi-automatic method to extract (during grammar induction) and recover (during parsing) the syntax of common semantically empty words. To this set of words with empty seman-

tics, we define a collection of linguistic signals that can serve as their cues, and match each signal to the set of lexical items it can recover. For example, the tense and aspects of verbs and predicative adjectives are signals for auxiliary verbs.

During SHRG induction, the signals are first identified from the DMRS node attributes and are then generally passed up from the syntactic head daughter. When extracting a binary construction, if the unextracted subtree (described in §3.1) contains a semantically empty lexical item that matches a signal of any of the daughters, a hypergraph fragment is extracted together with the subtree associated with that signal. During parsing, the same bottom-up signal passing procedures apply. If the R.H.S. hypergraph fragment of a binary production is recognized and any signal passed up matches that of an extracted subtree, the HRG production is applied and the CFG subtree is added on top of the two daughters in the derivation tree (see Fig. 3).

4 Towards Practical Grammar Approximation and Modelling

To approximate complex grammars or implicit relations established between trees and graphs by PSHRG, we introduce extensions for improving on both the precision and generalizability of modelling. The application of most of the proposed techniques is not limited to the DMRS, but to general MR parsing by PSHRG.

4.1 Annotations for Refining Compositions

Three techniques are introduced to impose restrictions to semantic compositions, allow probability to be estimated on more fine-grained SHRG productions, and prevent overgeneration.

Typed HRG. Chen and Sun (2020) introduced typed HRG, where each node of a hypergraph (fragment) and hyperedge is assigned a label chosen from a finite set. In typed HRG, an R.H.S. hypergraph fragment is recognized only if the type of its every node matches that of the corresponding node in the input graph. In our case, we propose to type a node by the major sense tag of the corresponding DMRS predicate. For example, in Fig. 5, the nodes on the R.H.S. correspond to `_want_v_I` and `_go_v_I` respectively, so both are typed `v`.

Annotation and Normalization of Derivation. An HPSG derivation tree merely records the recipe of a derivation, where the non-atomic rule symbols

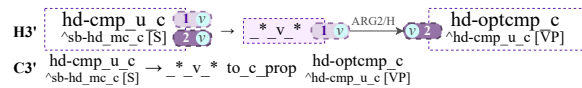


Figure 5: **P3** in Fig. 3 with annotated syntactic constructions, typed nodes and a delexicalized DMRS predicate.

only convey highly generalized linguistic principles. Each rule represents a unification of typed feature structures. Inspired by Zhang and Krieger (2011), we annotate each syntactic construction with that of their immediate parent (order-2 vertical markovization) and the syntactic category of the phrase (see Fig. 5). This adds extra contextual information to the constituents. We further normalize the derivations by substituting chains of unary lexical rules, affixation rules for punctuation marks, and the preterminals by the canonical form of a DMRS node or subgraph (see Fig. 4).

Framework-Specific Constraints. PSHRG Parsing on DMRS without regard for the MRS semantic algebra leads to inefficiency and overgeneration. In particular, the features `INDEX` and `LTOP` in MRS specify the semantic materials of a phrase that are accessible during composition. In the ERG, their values are determined by the type of composition.¹ Hence, when parsing a DMRS, every composition should ensure that subsequent compositions can only happen to the two variables of the newly composed item. This procedure resembles Carroll and Oepen (2005)’s proposal on index accessibility filtering. The checks can be easily incorporated into SHRG parsing (for the details, see Appendix B). Nevertheless, `INDEX` and `LTOP` are not the only features that permit compositions in the ERG. Therefore, the constraints introduced prevent overgeneration to a large extent but lead to undergeneration. In this work, we examine the two most prominent features, and how we should further integrate the MRS algebra to HRG is an open question.

4.2 Underspecification for Generalization

Two underspecification methods are developed to alleviate the rule sparsity and out-of-vocabulary (OOV) problems, which are the main challenges faced by general rule-based systems.

¹The precise algebras of the two features in the ERG are not discussed fully here. In brief, the `INDEX` and the `LTOP` usually come from the syntactic head, but come from a scopal modifier if the semantic composition is scopal. For expositions, see: Copestake et al. (2001); Copestake (2009).

Extended HRG Productions. When recognizing an HRG production $A \rightarrow R$ on a hypergraph H , we suggest that the external nodes of R be not distinguished from the non-external nodes. Consequently, the rewriting hyperedge A connects to a variable number of nodes, and such number depends on H . To motivate such decision, consider **H1** of Fig. 3. The fact that the hyperedge *_boy_n_l* connects to an external node is not significant to the characterization of the *sp-hd_n_c* (the specifier-head construction where the specifier is the semantic head). This effectively creates more SHRG productions that share the same probabilities if their R.H.S. hypergraphs (minus external nodes) are identical.

Delexicalization. The PSHRG models are estimated based on delexicalized productions: the lexeme stems of verbs, adjectives, adverbs, and nouns (whose DMRS predicates are in the form of ‘*_v_*’, ‘*_a_*’ and ‘*_n_*’ respectively) are underspecified (see Fig. 5). This is a significant distinction between our grammar and the ERG since the ERG is highly lexicalized.² Hence, delexicalization trades lexical preciseness for OOV coverage. Furthermore, since an approximation grammar is assumed to have no access to the lexical information of the underlying grammar, the results of our experiments would reflect the viability of PSHRG as a general approach to grammar approximation.

5 Experiments

The main objective of the experiments is to assess the performance of PSHRG models on simulating DMRS compositions and producing approximating derivation trees. Specifically, we reconstruct a derivation tree for each DMRS whose nonterminals are aligned to DMRS subgraphs and labelled by an ERG syntactic construction; the ERG 1214 contains more than 210 fine-grained syntactic constructions that reflect the distinguishing properties of different syntactic constructions.

As a secondary evaluation, we analyze our performance on the task of surface realization. The purpose of this is twofold: first, assessing the quality of the surface strings produced from the recon-

²Each lexical entry in the ERG is assigned to exactly one lexical type, which determines most of its syntactic and semantic properties. For example, *inform*, *advise* and *remind* share a lexical type because they all select a noun phrase and a sentential complement. The detailed lexical types interact with the highly generalized linguistic principles to produce precise linguistic interpretations.

structed derivation trees gives additional perspectives on the evaluation of our models; and secondly, there are existing works on surface realization from DMRS, so our models can be benchmarked against.

Finally, we evaluate the significance of the two proposed adaptations, namely recovering words with empty semantics and incorporating framework-specific constraints to PSHRG parsing. An instance of sample input and sample output are provided in Appendix C.

5.1 Data

The main data set we experiment on is the Ninth Growth of the Redwoods Treebank (Oepen et al., 2002).³ It contains English sentences from a range of domains including Wall Street Journal (WSJ) and the Brown corpus, each paired with the analyses of the 1214 version of the ERG. Each MRS is converted into a DMRS using Pydelphin (Copestake et al., 2016).⁴ We discard the instances with ambiguous analysis, disconnected DMRS, and unparsable MRS by Pydelphin.

To assess the scalability of our models, we further sampled sentences from the Gigaword v.5 corpus (Parker et al., 2011) for model training, where extra training instances are obtained by parsing sentences with the ACE and choosing the best ERG analysis for each sentence as ranked by the ACE.

After preprocessing, the total number of instances in the training and test sets are 70,774 and 10,042 respectively under the standard Redwoods data split. 70,774 extra training instances are created from the Gigaword corpus.

5.2 Experimental Configurations

We removed the mostly uninformative syntactically covert quantifiers (e.g., *udef_q*, *proper_q*) in all DMRS graphs. The numbers of DMRS nodes reported below are counted after the removal. Subgraph canonization (§3.3.2) was performed only on the DMRS subgraphs of fewer than seven nodes. The maximum length of the unary chains in the generated derivation trees was set to be three. The parser was implemented in PyPy3.6 and ran under one Intel Xeon E5-2697 CPU on x86_64 Linux. Our implementation is available online.⁵

³<http://svn.delph-in.net/erg/tags/1214/tsdb/gold>

⁴<https://github.com/delph-in/pydelphin>

⁵<https://github.com/aaronlolo326/pshrgOnDMRS>

Derivation Annotation	Model	ParsEval-Graph			Coverage
		P	R	F ₁	
M1C	SHRG-PCFG	76.79	74.90	75.84	80.58%
	PSHRG	81.86	79.85	80.84	80.58%
	PSTHRG	83.81	81.23	82.50	74.31%
M2C	SHRG-PCFG	83.13	80.44	81.77	68.69%
	PSHRG	84.52	81.60	83.03	68.69%
	PSTHRG	87.04	83.66	85.32	61.50%

Table 1: Results on the accuracy of derivation reconstruction under standard Redwoods data split.

5.3 Results

5.3.1 On Derivation Tree Reconstruction

If a DMRS is parsed correctly, the synchronously reconstructed derivation should not deviate too much from the original derivation. Nevertheless, equivalence is a sufficient, but not necessary condition for a parse to be correct since syntactic differences do not necessarily contribute to semantic differences.

We devise a modified version of the ParsEval (Black et al., 1991) measure, ParsEval-Graph, to assess the quality of the generated trees, as the constituents of the trees are not aligned to a surface string but on the input semantics. In ParsEval-Graph, the alignment of a constituent refers to the DMRS nodes covered instead of the characters covered in the surface string. Following ParsEval, ParsEval-Graph only accounts for binary rules, and preterminals are disregarded. All ParsEval-Graph scores are evaluated on the parsable instances of the respective models on unannotated derivation trees.

As introduced in §4.1, we experiment with the typed variant of PSHRG, PSTHRG, and two configurations of tree annotation, namely order-2 vertical markovization with syntactic category annotation (M2C), and only syntactic category annotation (M1C). Since no existing works on data-driven DMRS parsing or surface realization produce syntactic derivations, we develop a baseline model SHRG-PCFG for benchmarking. SHRG-PCFG parses a DMRS with the SHRG induced and the probability of each parse is given by a PCFG model where the probability of each CFG production $A \rightarrow R'$ is modelled as the fraction of times $A \rightarrow R'$ appears among all $A \rightarrow *$ in the training data. The ACE is not evaluated because it always generates derivations faithful to the ERG.

As reported in Table 1, all PSHRG models attain F₁ scores of over 80 consistently across settings

Model	Training Set	ParsEval-Graph			Coverage
		P	R	F ₁	
PSHRG (M1C)	Redwoods	81.86	79.85	80.84	80.58%
	Redwoods + Gigaword	81.60	79.66	80.62	85.11%
PSTHRG (M1C)	Redwoods	83.81	81.23	82.50	74.31%
	Redwoods + Gigaword	83.39	80.85	82.10	80.62%

Table 2: Results on derivation reconstruction on the Redwoods test set with different training sets.

and outperform the baseline under the same annotation configurations. More extensive annotation and typing the grammar respectively improve the F₁ score by about 2, at the expense of coverage reduction caused by rule sparsity when the Redwoods training set does not provide adequate data for the over-specific annotation. It is insightful to note that the baseline under M2C performs at a level between the PSHRG model and PSTHRG model on M1C, which conveys that the contextual information of a nonterminal node could already provide ample information on the semantic compositions.

To study models’ performance with respect to the size of the training data, we add the Gigaword instances on top of the Redwoods training set. This doubles the amount of training data. As reported in Table 2, the coverages of the two models increase by 4.50% and 6.31% respectively with more data. Nevertheless, the accuracy of derivations does not improve further, as frequency-based context-free probability models have low learning capacities.

Despite all DMRS being generated by the ERG, the ACE does not parse every DMRS—parsing fails when a DMRS predicate is OOV. In contrast, our models parse more instances than the ACE when given more training data, since they generalize to OOV with delexicalization. Although delexicalization removes much lexical information, we suggest that SHRG-based parsing and the incorporation of MRS-specific constraints can restrict compositions outside of the ERG to a large extent.

5.3.2 On Surface Realization

To produce a surface string from a DMRS, we realize the most frequently recorded surface form for each preterminal from the reconstructed derivation tree. We compare our work with the Neural MRS (Hajdik et al., 2019) and the ACE. The Neural MRS generates in an end-to-end manner without intermediate syntactic derivations. For more comparable results, we evaluate the models under M1C annotation configuration since they have similar parse

Model	BLEU			Generate Syntactic Structures
	Redwoods	WSJ	Brown	
Neural MRS	66.11	65.78	45.00	No
ACE (ERG)	62.21	-	-	Yes
PSHRG (M1C)	59.33	63.65	58.29	Yes
PSTHRG (M1C)	60.67	64.77	59.47	Yes

Table 3: Results on surface realization. Redwoods, WSJ and Brown refer to the Redwoods standard splits, WSJ–WSJ and WSJ–Brown splits respectively.

coverages to the ACE.

We evaluate the generation quality with BLEU (Papineni et al., 2002) using SacreBLEU (Post, 2018)⁶. Following (Hajdik et al., 2019)’s evaluation on in- and out-of-domain performances, we experiment on the different training–test data splits, namely the WSJ–WSJ and WSJ–Brown splits. WSJ contains 34,751 training instances and 1,442 test instances, and Brown contains 2,181 test instances.⁷ All BLEU scores are evaluated on the parsable instances of the respective models.

Table 3 shows that our models perform consistently across data splits. Under the Redwoods standard data split, our models are worse than the neural model. With similar parse coverages to the ACE, our performance is also close to the ACE. Under the WSJ–Brown split, our PSHRG models outperform the Neural MRS.⁸ The PSHRG (M1C) parses 74.90% of the test set under the WSJ–WSJ split. When the model is typed and when switching from in- to out-of-domain, the models parse about 7% less data respectively. The coverages of M1C models reported here are lower than those in Table 1 since the amount of training data is halved. Therefore, we consider the relative decreases in coverage from in- to out-of-domain of our respective models to be more insightful on models’ transferability between domains than the absolute coverage.

Apart from the automatic evaluation, we also value qualitative details and seek linguistically interesting phenomena that result from a grammar-based approach. To this end, we observe that our models identify different realization possibilities from the original sentence to the same semantics

⁶<https://github.com/mjpost/sacreBLEU>

⁷The numbers of training and test instances of Neural MRS are slightly greater than ours after respective preprocessing.

⁸To provide context on assessing our model under a full-coverage scenario, we try to provide a naive fallback for the unparseable instances: For each of these instances, the partially realized non-overlapping constituents are concatenated in order of decreasing number of the DMRS nodes covered to form the final sentence. With this quick fix, our PSHRG (M1C) model attains a BLEU score of 51.08 on the WSJ–Brown split.

[Ref]	Also of interest are the mountains around the Rosendal community in Kvinnherad kommune.
[Sys]	The mountains around the Rosendal community in Kvinnherad kommune are also of interest.
[Ref]	Here is a list of the source and target languages SYSTRAN works with.
[Sys]	Here is a list of the source and target languages with which SYSTRAN works.
[Ref]	Earlier this year, bankers and other investors were willing to provide financing because they assumed there would be major gains in both profitability and sales, Mr. Rosenthal added.
[Sys]	Earlier this year bankers and other investors were willing to provide financing because they assumed that it would be major gains in profitability and sales, added Mr. Rosenthal.

Table 4: Examples of syntactic variations demonstrated in the generated texts. In the first example, the canonical subject–verb order is recovered when the original text is an inverted sentence. In the second example, inversion is preserved and preposition fronting is adopted instead of stranding as in the reference sentence (see Fig. 8 of Appendix C). The third example reflects multiple linguistic variations, including the insertion of complementizer *that*, omission of semantically empty *both* and quotative subject–verb inversion .

that conform to the ERG. Some syntactic variations are reported and explained in Table 4.

Compared to neural approaches, PSHRG is a shallow statistical model with a high inductive bias. It encodes a syntax–semantics interface effectively through tree- and graph- rewriting. Even though our models are not engineered towards the task of surface realization, and with limited morphological analyses and no language modelling, our approach is still competitive as evaluated quantitatively and qualitatively. We suggest that PSHRG-based approaches and neural models be decent alternatives to each other for general surface realization from MR: neural models provide full-coverage and high-quality generation when substantial training data is available, whereas PSHRG-based solutions extrapolate from limited data and in out-of-domain scenarios, and produce interpretable derivations.

5.3.3 Ablation Studies

We conduct a few more experiments to test against the significance of two proposed adaptations, namely the solution to semantically empty words (§3.3.3) and framework-specific constraints (§4.1). We implement two models, PSTHRG- \emptyset (M1C) and PSTHRG- λ (M1C), which parse DMRS without regard for semantically empty lexical items and MRS constraints respectively.

As reported in Table 5, The treatment of empty semantics not only adds 15.51% more parsable instances but also corrects some parses in the 58.80% that require analyses of empty semantics, thus pro-

Model	ParsEval-Graph			BLEU	Coverage
	P	R	F ₁		
PSTHRG (MIC)	83.81	81.23	82.50	60.67	74.31%
PSTHRG- \emptyset (MIC)	84.68	81.09	82.85	54.07	58.80%

Table 5: Results of ablation of inserting words with empty semantics under the Redwoods standard split.

ducing more accurate surface strings.

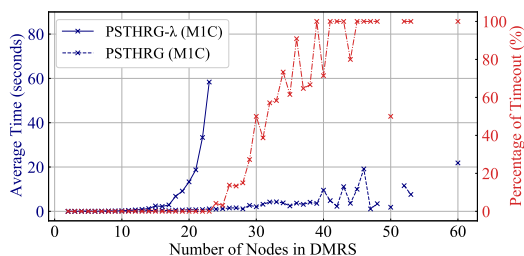


Figure 6: Results of ablation of framework-specific constraints on parsing time under the WSJ-WSJ split.

Model	ParsEval-Graph			BLEU
	P	R	F ₁	
PSTHRG (MIC)	86.16	82.66	84.37	64.77
PSTHRG- λ (MIC)	85.06	81.59	83.29	64.02

Table 6: Results of ablation of framework-specific constraints on derivation reconstruction and surface realization qualities under the WSJ-WSJ split.

Fig. 6 shows the importance of MRS constraints on parsing efficiency. We set a time limit of 300 seconds on parsing, and the parsing of 16.71% of the WSJ test set exceeds the time limit. When the DMRS contains 24 or more nodes, timeouts occur on at least one DMRS graph of each size. Table 6 shows that when the constraints are enforced, all parses are completed in much shorter times and the derivations reconstructed are more accurate.

6 Discussion

The proposed frequency-based PSHRG models are simple yet competitive data-driven baselines for recovering derivations of DMRS. They can be further combined with sophisticated machine learning methods for a more accurate parse ranking. More extensive features can also be included to enhance grammar approximation. For instance, the feature-paths of ERG signs are shown to be helpful for PCFG approximation (Zhang and Krieger, 2011). Language-specific knowledge about words with empty semantics is also critical for syntactic purposes. In terms of efficiency, Ye and Sun (2020)

showed that exact parsing can be very practical on Elementary Dependency Structures (EDS; Oepen and Lønning, 2006), a close equivalent to DMRS that excludes scopal information. Different from Ye and Sun (2020)’s implementation, we retain more than 210 ERG syntactic constructions for precision and adopt delexicalization for generalization, both of which increase the search space of parsing and trade efficiency. We suggest that the described PSHRG-based approach can be a potential alternative to the unification-based ACE generator for surface realization from DMRS, whilst improving parsing coverage, accuracy and efficiency without sacrificing one another would be a critical problem for future research. In this paper, we report our findings with respect to the engineering decisions we investigated based on the data at hand.

In principle, the application of the described PSHRG-based approach is not limited to DMRS, but also to generic graph-to-tree translations that exhibit compositionality, if suitable data of aligned trees and graphs is available. If explicit associations do not exist between the trees and graphs, the induced grammar formalizes the underlying relations and patterns; otherwise, the induced grammar provides an approximation to such relations, which can be desirable for computation purposes.

7 Conclusion

Based on the experimental results, we can assess the contributions of this work from three perspectives: (1) PSHRG with framework-specific adaptations as a formalism that approximates the semantic composition process of DMRS, (2) PSHRG graph parsing with framework-independent extensions as a general approach to modelling compositional graph-to-tree translation, and (3) derivation reconstruction with a PSHRG induced from data as a solution to surface realization from MRs that provides explainability, syntactic disambiguation, and syntactic variations. We hope that this work provides relevant and substantial empirical insights to stimulate more research on approaching MR processing with linguistically-motivated methods.

Acknowledgements

The work was supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China [Project No.: CUHK 14205618], and CUHK Direct Grant No. 4055159.

References

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. [Abstract Meaning Representation for sembanking](#). In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.
- 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. Association for Computational Linguistics.
- E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. [A procedure for quantitatively comparing the syntactic coverage of English grammars](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.
- John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznanski. 1999. [An efficient chart generator for \(semi-\) lexicalist grammars](#). In *Proceedings of the 7th european workshop on natural language generation*, pages 86–95.
- John Carroll and Stephan Oepen. 2005. [High efficiency realization for a wide-coverage unification grammar](#). In *Second International Joint Conference on Natural Language Processing: Full Papers*.
- Yufei Chen and Weiwei Sun. 2020. [Parsing into variable-in-situ logico-semantic graphs](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6772–6782, Online. Association for Computational Linguistics.
- Yufei Chen, Weiwei Sun, and Xiaojun Wan. 2018. [Accurate SHRG-based semantic parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418, Melbourne, Australia. Association for Computational Linguistics.
- 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. Association for Computational Linguistics.
- 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. Association for Computational Linguistics.
- Ann Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszyńska. 2016. [Resources for building applications with dependency Minimal Recursion Semantics](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 1240–1247, Portorož, Slovenia. European Language Resources Association (ELRA).
- Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. [Minimal recursion semantics: An introduction](#). *Research on language and computation*, 3(2):281–332.
- Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. [An algebra for semantic construction in constraint-based grammars](#). In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 140–147, Toulouse, France. Association for Computational Linguistics.
- Marco Damonte and Shay B. Cohen. 2019. [Structural neural encoders for AMR-to-text generation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3649–3658, Minneapolis, Minnesota. Association for Computational Linguistics.
- Lucia Donatelli, Meaghan Fowlie, Jonas Groschwitz, Alexander Koller, Matthias Lindemann, Mario Mina, and Pia Weißenhorn. 2019. [Saarland at MRP 2019: Compositional parsing across all graphbanks](#). In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 66–75, Hong Kong. Association for Computational Linguistics.
- Frank Drewes and Anna Jonsson. 2017. [Contextual hyperedge replacement grammars for Abstract Meaning Representations](#). In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 102–111, Umeå, Sweden. Association for Computational Linguistics.
- Frank Drewes, H-J Kreowski, and Annegret Habel. 1997. [Hyperedge replacement graph grammars](#). In *Handbook Of Graph Grammars And Computing By Graph Transformation: Volume 1: Foundations*, pages 95–162. World Scientific.
- Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. [Generation from Abstract Meaning Representation using tree transducers](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 731–739, San Diego, California. Association for Computational Linguistics.
- Dan Flickinger. 2000. [On building a more efficient grammar by exploiting types](#). *Natural Language Engineering*, 6(1):15–28.

- Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. *Language from a cognitive perspective: Grammar, usage, and processing*, 201:31–50.
- Jonas Groschwitz, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2017. [A constrained graph algebra for semantic parsing with AMRs](#). In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*.
- Jonas Groschwitz, Alexander Koller, and Christoph Teichmann. 2015. [Graph parsing with s-graph grammars](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1481–1490, Beijing, China. Association for Computational Linguistics.
- Jonas Groschwitz, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. [AMR dependency parsing with a typed semantic algebra](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1831–1841, Melbourne, Australia. Association for Computational Linguistics.
- Valerie Hajdik, Jan Buys, Michael Wayne Goodman, and Emily M. Bender. 2019. [Neural text generation from rich semantic representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2259–2266, Minneapolis, Minnesota. Association for Computational Linguistics.
- Matic Horvat. 2017. [Hierarchical statistical semantic translation and realization](#). Technical Report UCAM-CL-TR-913, University of Cambridge, Computer Laboratory.
- Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. 2012. [Semantics-based machine translation with hyperedge replacement grammars](#). In *Proceedings of COLING 2012*, pages 1359–1376, Mumbai, India. The COLING 2012 Organizing Committee.
- Alexander Koller. 2015. [Semantic construction with graph grammars](#). In *Proceedings of the 11th International Conference on Computational Semantics*, pages 228–238, London, UK. Association for Computational Linguistics.
- Alexander Koller, Stephan Oepen, and Weiwei Sun. 2019. [Graph-based meaning representations: Design and processing](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 6–11, Florence, Italy. Association for Computational Linguistics.
- Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. [Compositional semantic parsing across graphbanks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.
- Stephan Oepen, Omri Abend, Jan Hajic, Daniel Herscovich, Marco Kuhlmann, Tim O’Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 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. Association for Computational Linguistics.
- Stephan Oepen and Jan Tore Lønning. 2006. [Discriminant-based MRS banking](#). In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, Genoa, Italy. European Language Resources Association (ELRA).
- Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. [The LinGO redwoods treebank: Motivation and preliminary applications](#). In *COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword. *Linguistic Data Consortium*.
- 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*, pages 32–41, Beijing, China. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Linfeng Song, Xiaochang Peng, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2017. [AMR-to-text generation with synchronous node replacement grammar](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 7–13, Vancouver, Canada. Association for Computational Linguistics.
- Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. [A graph-to-sequence model for AMR-to-text generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, Melbourne, Australia. Association for Computational Linguistics.

Erik Velldal and Stephan Oepen. 2006. [Statistical ranking in tactical generation](#). In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 517–525, Sydney, Australia. Association for Computational Linguistics.

Yajie Ye and Weiwei Sun. 2020. [Exact yet efficient graph parsing, bi-directional locality and the constructivist hypothesis](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4100–4110, Online. Association for Computational Linguistics.

Yajie Ye, Weiwei Sun, and Xiaojun Wan. 2018. [Language generation via DAG transduction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1928–1937, Melbourne, Australia. Association for Computational Linguistics.

Yi Zhang and Hans-Ulrich Krieger. 2011. [Large-scale corpus-driven PCFG approximation of an HPSG](#). In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 198–208, Dublin, Ireland. Association for Computational Linguistics.

A A Passive Chart Parser

Although it is not the intent of this work to suggest a new HRG parsing algorithm, we provide more information on the parser implemented so that it provides more context for comprehending the absolute parsing time in Fig. 6.

A.1 Rewriting DMRS Hypergraph

As described in §3, a hypergraph can be instantiated corresponding to a DMRS. During SHRG parsing, the hypergraph undergoes graph-rewriting, where its subgraphs are rewritten by nonterminal hyperedges, and produces other hypergraphs.

For every hypergraph H , we categorize every hyperedge into one of the two types, s-hyperedge or c-hyperedge. An s-hyperedge is a terminal hyperedge that is identified and labelled by a DMRS edge. It connects to exactly two nodes of H . A c-hyperedge corresponds to a node or a connected DMRS subgraph. A terminal c-hyperedge is labelled by a DMRS predicate. A nonterminal c-hyperedge is labelled by the canonical form of a DMRS subgraph (see §3.3.2) or an ERG syntactic construction.

A.2 Representation of Chart Items

In our chart parser, a passive item is essentially a c-hyperedge packed with additional information. It is represented by $M = \langle Y, A, \Sigma \rangle$, where Y is the set

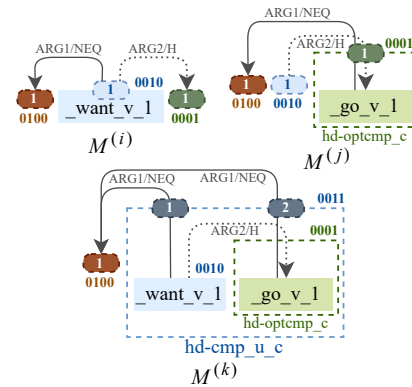


Figure 7: Example of an application of a binary rule on two passive items $M^{(i)}$ and $M^{(j)}$ to compose the item $M^{(k)}$. With respect to $M^{(i)}$ and $M^{(j)}$, the interior s-hyperedges are represented by dotted arrows and the exterior s-hyperedges by solid arrows. The bit vectors on the top right corner of each c-hyperedge indicate sets of DMRS nodes the passive item covers.

of DMRS nodes covered by the item, A is the c-hyperedge label and Σ is the set of s-hyperedges incident to the item. In a passive item $M = \langle Y, A, \Sigma \rangle$, each of the s-hyperedges $\sigma \in \Sigma$ is represented by $\sigma = \langle s, t, k, d, x \rangle$, where s, t, k represent the source and target DMRS node, and the key of the edge. These three variables together specify a unique edge in a DMRS multigraph; d is a boolean variable indicating if the DMRS edge $\langle s, t, k \rangle$ is an outgoing edge from the c-hyperedge and $x \in \mathbb{Z}^+$ specifies the order of the node of the c-hyperedge this s-hyperedge connects to. Fig. 7 illustrates three example passive items.

Algorithm 1 shows the high-level procedures of a minimal PSHRG parsing algorithm for DMRS. It only includes the application of unary and binary rules without constructional content⁹ and semantically empty words. The synchronous CFG part is also omitted for simplicity. It parses a DMRS D given a PSHRG G with production probabilities π and returns the best parse. Z is the set of the small subgraphs extracted from training data.

A.3 Bottom-Up Parsing

Line 3 of Algorithm 1. We first initialize the adjacency information of each DMRS node. α maps a set of DMRS nodes to the set of their adjacent nodes. A DMRS node set of size n is represented by an n -hot bit vector. N_D returns the neighbour-

⁹Some ERG constructions introduce semantic content (DMRS predicates) that interact with the daughter’s semantic materials to the resulting MRS structure.

Algorithm 1: A minimal PSHRG parsing algorithm for DMRS.

```

1 Function PARSE( $G = \langle N, T, T', P, S, \pi \rangle, D, Z$ )
2    $C_{\text{new}} \leftarrow \{\}$ ;
3   for  $v \in V(D)$  do  $\alpha(\{v\}) \leftarrow N_D(v)$ ;
4   for  $Y \subseteq V(D) \mid |Y| < 7$  do
5     if  $|Y| = 1 \wedge \varphi_D(Y) \in T$  then
6        $A \leftarrow \varphi_D(Y)$ ;
7        $\Sigma \leftarrow \{\}$ ;
8       for  $\langle s, t, k \rangle \in E(D) \mid (s \in Y \oplus t \in Y)$ 
9         do
10         $\sigma \leftarrow \langle s, t, k, \llbracket s \in Y \rrbracket, 1 \rangle$ ;
11         $\Sigma \leftarrow \Sigma \cup \sigma$ ;
12      else if  $|Y| > 1 \wedge D[Y] \in Z$  then
13         $A, \Sigma \leftarrow \text{CANONIZE}(D[Y])$ ;
14         $\alpha(Y) \leftarrow \text{AND}(N_D(Y), \text{NOT}(Y))$ ;
15         $M \leftarrow \langle Y, A, \Sigma \rangle$ ;
16         $C_{\text{new}} \leftarrow C_{\text{new}} \cup M$ ;
17         $p_{\log}(M) \leftarrow 0$ ;
18       $C_S \leftarrow \{\}$ ;
19       $R_G \leftarrow \{R \mid A \rightarrow \langle R, R', \sim \rangle \in P\}$ ;
20       $R'_G \leftarrow \{R' \mid A \rightarrow \langle R, R', \sim \rangle \in P\}$ ;
21      while  $C_{\text{new}} \neq \emptyset$  do
22         $C_{\text{unary}}, p_{\log} \leftarrow$ 
23         $\text{APPLYUNARY}(P, \pi, C_{\text{new}}, p_{\log})$ ;
24         $C_{\text{new}} \leftarrow C_{\text{new}} \cup C_{\text{unary}}$ ;
25         $C_S \leftarrow C_S \cup \{\langle Y, A, \Sigma \rangle \in C_{\text{all}} \mid$ 
26         $Y = V(D) \wedge A = S\}$ ;
27         $C_{\text{all}} \leftarrow C_{\text{all}} \cup C_{\text{new}}$ ;
28         $C_{\text{binary}}, p_{\log}, \alpha \leftarrow \text{APPLYBINARY}($ 
29         $P, \pi, D, C_{\text{new}}, C_{\text{all}}, p_{\log}, \alpha, R_G, R'_G)$ ;
30         $C_{\text{new}} \leftarrow C_{\text{binary}}$ ;
31      return  $\arg \max_{M \in C_S} p_{\log}(M)$ ;

```

hood of a set of nodes in D .

Line 4 to 16 of Algorithm 1. In this section, we instantiate passive items for the terminal c-hyperedges which correspond to a DMRS node, and for the nonterminal c-hyperedges which correspond to a connected DMRS subgraph of fewer than seven nodes. $V(D)$ and $E(D)$ denote the sets of DMRS nodes and edges in D respectively. $\varphi_D(Y)$ returns the DMRS predicate of the unit set Y in D . For a recognized terminal, we compute the incident s-hyperedges Σ of it. CANONIZE returns A , the canonical form, and Σ , the incident s-hyperedges of $D[Y]$, where $D[Y]$ is the node-induced subgraph formed by the node set Y . The node order of the s-hyperedges in Σ follows the order of the corresponding node representation in the canonical form. The procedures of obtaining Σ here is similar to those described at line 7 to 10 and are omitted. Line 13 updates the adjacency information of Y (for the detailed procedure, refer to line 16 of Algorithm 2). Then, a passive item $M = \langle Y, A, \Sigma \rangle$ is created and added to the chart.

$p_{\log}(M)$ maps the passive item M to the highest possible log probability of M at that instant.

Line 20 to 27 of Algorithm 1. Given the extracted SHRG productions P , new items are created by applying unary rules to the passive items just created (APPLYUNARY), then binary rules between all the new passive items and their respective adjacent passive items (APPLYBINARY). These two steps are iterated repeatedly until no more passive items are created. Line 22 records the successful parses, i.e. the items that cover all DMRS nodes and whose c-hyperedge label equals the start symbol S of G . Finally, the successful parse with the highest log probability is returned.

A.4 Grammar Intersection

The grammar intersection of an HRG is similar to that of a CFG for strings in the CYK algorithm, where neighbouring constituents are combined to form new passive items. Algorithm 2 describes the generation of new passive items via binary rules (Line 24 of Algorithm 1).

For a set of passive items, we further define the following: the common incident s-hyperedges are called interior s-hyperedges; and the remaining s-hyperedges are called exterior s-hyperedges, which connect to other c-hyperedges in the subsequent rewriting steps (for an illustration, see Fig. 7).

Line 3 of Algorithm 2. Grammar intersection is performed only on the adjacent passive items (there exists at least one s-hyperedge that connects them) that cover disjoint sets of DMRS nodes. For efficiency, all passive items are indexed by Y , so that validating adjacency and disjointness of DMRS subgraphs can be manipulated by quick bitwise operations to return the pairs of items possible for grammar intersection. All items covering Y are further indexed by A so that we only consider the pairs of items whose labels exist in the set of the CFG daughter sequences of productions, R'_G .

Line 4 to 7 of Algorithm 2. For each pair of passive items in C_{\cap} , we try to generate possible new passive items. Line 5 first computes the union of set of DMRS nodes covered by the two items. Then, the interior s-hyperedges of the new passive item are computed at line 6. As the notion of external nodes is dropped when we match an R.H.S. hypergraph fragment R (see §4.2), the recognition only requires the identification of the set of c-hyperedges in R plus the interior s-hyperedges.

Algorithm 2: Generation of passive items with binary rules.

```

1 Function APPLYBINARY( $P, \pi, D, C_{\text{new}}, C_{\text{all}}, p_{\log}, \alpha, R_G, R'_G$ )
2    $C_{\text{binary}} \leftarrow \{\}$ ;
3    $C_{\cap} \leftarrow \{\langle \langle Y^{(i)}, A^{(i)}, \Sigma^{(i)} \rangle, \langle Y^{(j)}, A^{(j)}, \Sigma^{(j)} \rangle \rangle \in C_{\text{new}} \times C_{\text{all}} \mid$ 
   ( $\text{AND}(Y^{(i)}, Y^{(j)}) = 0 \wedge \text{AND}(\alpha(Y^{(i)}, Y^{(j)}) \neq 0 \wedge (\langle A^{(i)}, A^{(j)} \rangle \in R'_G \vee \langle A^{(j)}, A^{(i)} \rangle \in R'_G))\}$ ;
4   for  $\langle \langle Y^{(i)}, A^{(i)}, \Sigma^{(i)} \rangle, \langle Y^{(j)}, A^{(j)}, \Sigma^{(j)} \rangle \rangle \in C_{\cap}$  do
5      $Y^{(k)} \leftarrow \text{OR}(Y^{(i)}, Y^{(j)})$ ;
6      $I^{(k)} \leftarrow \text{GETINTERIORS-HYPEREDGES}(\Sigma^{(i)}, \Sigma^{(j)})$ ;
7      $H^{(k)} \leftarrow \text{GETHYPERGRAPH}(A^{(i)}, A^{(j)}, I^{(k)})$ ;
8     if  $H^{(k)} \in R_G$  then
9        $\Sigma^{(k)} \leftarrow \text{GETEXTERIORS-HYPEREDGES}(\Sigma^{(i)}, \Sigma^{(j)})$ ;
10      for  $A \rightarrow \langle R, R', \sim \rangle \in P \mid (R = H^{(k)} \wedge \text{ISVALIDCOMPOSITION}(D, I^{(k)}, \Sigma^{(k)}, A))$  do
11         $M^{(k)} \leftarrow \langle Y^{(k)}, A, \Sigma^{(k)} \rangle$ ;
12         $\hat{p} \leftarrow p_{\log}(\langle Y^{(i)}, A^{(i)}, \Sigma^{(i)} \rangle) + p_{\log}(\langle Y^{(j)}, A^{(j)}, \Sigma^{(j)} \rangle) + \log(\pi(A \rightarrow \langle H^{(k)}, R', \sim \rangle))$ ;
13        if  $M^{(k)} \in C_{\text{binary}} \cup C_{\text{all}}$  then
14           $p_{\log}(M^{(k)}) \leftarrow \max(\hat{p}, p_{\log}(M^{(k)}))$ ;
15        else
16           $p_{\log}(M^{(k)}) \leftarrow \hat{p}$ ;
17           $\alpha(Y^{(k)}) \leftarrow \text{AND}(\text{OR}(\alpha(Y^{(i)}), \alpha(Y^{(j)})), \text{NOT}(Y^{(k)}))$ ;
18           $C_{\text{binary}} \leftarrow C_{\text{binary}} \cup M^{(k)}$ ;
19   return  $C_{\text{binary}}, p_{\log}, \alpha$ ;

```

Thus, the hypergraph $H^{(k)}$ is obtained at line 7 simply by gluing the two c-hyperedges with the interior s-hyperedges.

Line 8 to 18 of Algorithm 2. If the hypergraph obtained at line 8 appeared on the R.H.S of any production in P (line 8), we proceed to compute the exterior s-hyperedges of the new passive item (line 9). Line 10 searches for the productions of G whose (1) R.H.S. hypergraph is $H^{(k)}$ and (2) rewriting syntactic construction A licenses a valid composition (see §4.1; for the detailed procedures, see Appendix B). Line 11 to 13 factor local ambiguities; if we wish to extend the algorithm to obtain k-best derivations, we just keep the k-best hypotheses instead of the 1-best. Line 16 updates the adjacency information of $Y^{(k)}$ through bitwise operations. Finally, the newly generated passive items, the updated log probabilities, and the adjacency map are returned.

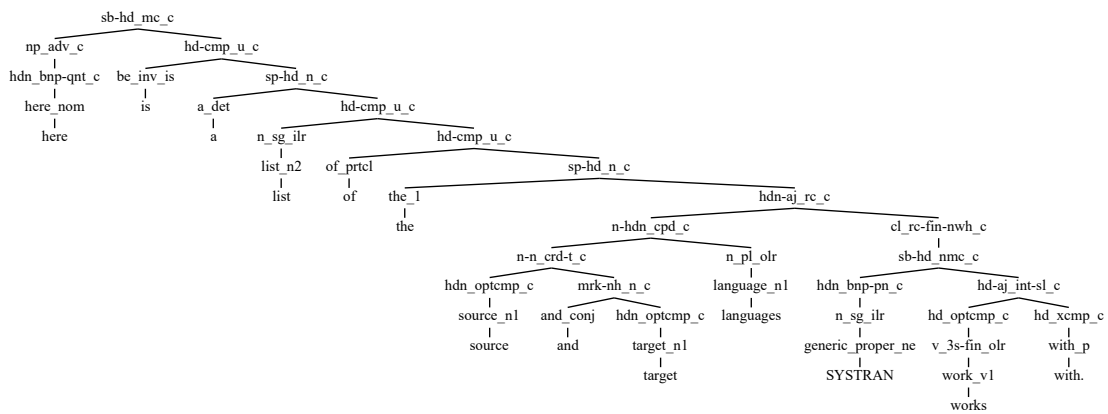
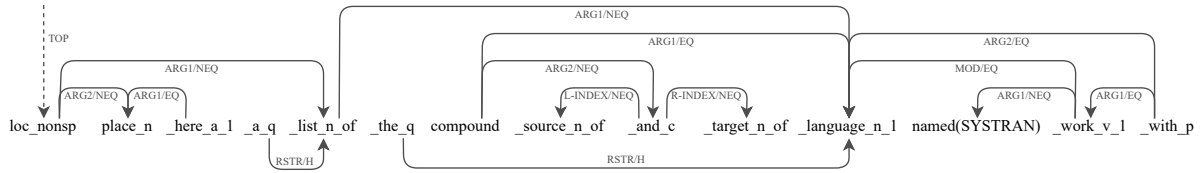
B MRS-Specific Constraints in HRG parsing

We describe how the MRS-specific constraints introduced in §4.1 can be incorporated into HRG parsing. In general, the accessibility check amounts to validating the following conditions for each c-hyperedge e in the R.H.S. hypergraph fragment when composing a new c-hyperedge a : (1) if e is not the INDEX of a , it is not the INDEX of the DMRS, and no s-hyperedges of type /EQ or /NEQ

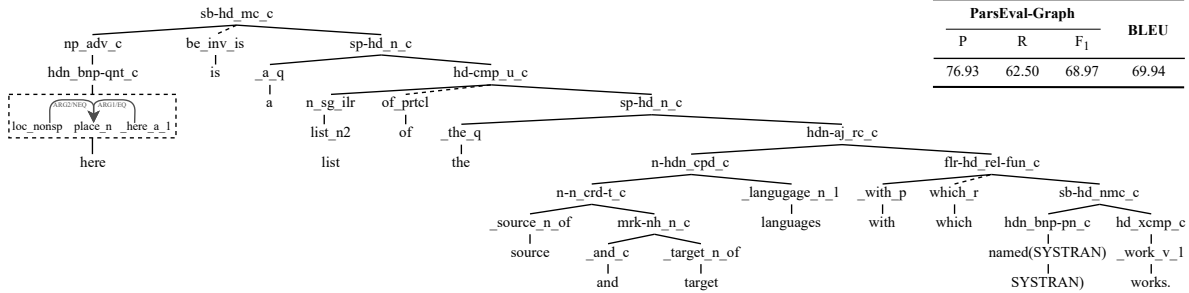
outside a are connected to e , and (2) if the scope of e , l , is not the LTOP of a , l is not the TOP of the DMRS, and no s-hyperedges of type /H, /HEQ or /EQ outside a are connected to e . By scanning the interior s-hyperedges once, we can decide on the new LTOP, and by scanning the exterior s-hyperedges once, we can verify the accessibility of both variables. Our implementation excludes the check of /NEQ edges due to undergeneration as described in §4.1.

C Sample Input and Output

Fig. 8 shows an instance of sample input and output compared against the original derivation. The model used to produce the results is PSTHRG (MIC) trained on the Redwoods training data.



[Ref] Here is a list of the source and target languages SYSTRAN works with.



ParsEval-Graph			
P	R	F ₁	BLEU
76.93	62.50	68.97	69.94

[Sys] Here is a list of the source and target languages with which SYSTRAN works.

Figure 8: From top to bottom are a DMRS (with syntactically covert quantifiers removed), the original ERG derivation (with punctuation rules removed) with the original sentence, and the reconstructed derivation with the realized sentence. Note that the nodes of an input DMRS are not ordered. The preterminal of *here* corresponds to a DMRS subgraph of three nodes. With preposition fronting, the semantically empty particle *of*, the auxiliary verb *is* and the relative pronoun *which* are all recovered and inserted in correct positions.