Supplementary Material: Semantic Graph Parsing with Recurrent Neural Network DAG Grammars

Federico Fancellu[⊙] and **Sorcha Gilroy**[☽] and **Adam Lopez**[⊕] and **Mirella Lapata**[⊕]
[⊙]Samsung AI Centre (SAIC), Toronto, Canada
[☽]Peak AI, Manchester, United Kingdom
[⊕]University of Edinburgh, Edinburgh, United Kingdom

A Cross-lingual Experiments: Feature Ablation

Results for the ablation study on cross-lingual parsing are shown in Figure 1.

B Restricted DAG Grammar

In Section 2 of the main paper, we have defined the graph-aware rewriting system as a *graph grammar*, specifically a *restricted DAG grammar* (RDG, Björklund et al., 2016), a formalism designed to model linearized DAGs. In the interest of space and clarity, we have presented these linearized DAGs as strings; here, for completeness, we describe their graph counterparts.

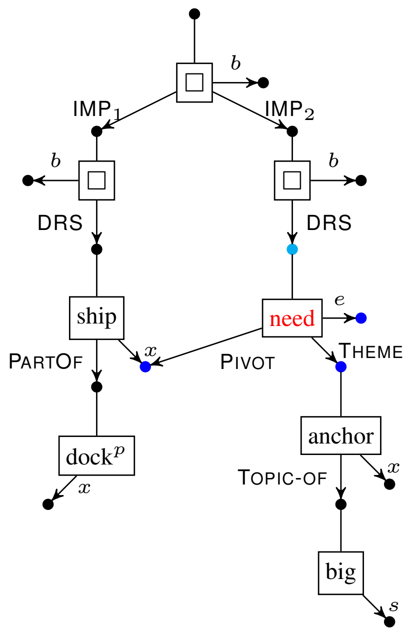

Figure 1: Hypergraph for the sentence 'Every ship in the dock needs a big anchor'. A component hyperedge is presented colorcoded with the label in red, the source node in cyan and target nodes in blue.

RDG operates on *directed, acyclic hypergraph*.

Definition 1. *A **directed acyclic hypergraph** G over a label set Σ and a tentacle label set Γ is a tuple $(V, E, src, tar, lab, tent\text{-}lab)$, where V is a finite set of nodes, E is a finite set of hyperedges, src: $E \to V$ assigns a **source** node to each hyperedge, tar: $E \to V^*$ assigns a sequence of **target** nodes to each edge, lab: $E \to \Sigma$ assigns a label to each edge, and tent-lab: $E \times V \to \Gamma$ assigns a label to each tentacle. G is marked if it includes an extra set $X \in V$, containing the external nodes.*

We exemplify Definition 1 by representing the graph in Figure 2 in the main paper as an hypergraph, as shown in Figure 1. Each hyperedge can be identified around each label (inside square nodes), this being either a box label (e.g. □) or a lexical predicate (e.g. 'need'). Source nodes are placed right above the label and targets nodes are the endpoints of edges going out of the label; these edges are also labelled with tentacle labels, these usually being semantic roles (e.g. THEME, PIVOT) or operators (e.g. IMP_1). An extra edge is used to encode the semantic type of the hyperedge. These typed edges are also used to mark reentrancies as in the case of the edge PIVOT connecting 'need' and the variable x of 'ship'.

Given a hypergraph G, we extract a restricted DAG grammar (RDG), which we describe as follows:

Definition 2. *A DAG grammar is a system $H = (\Sigma, N, S, P)$ where Σ and N are the sets of terminals and non-terminals respectively where $\Sigma \cap N = \emptyset$, S is the start non-terminal and P is a set of productions. For each production $A \to F, A \in N$ and F is a marked DAG containing an extra sequence of nodes X – the marking of G, with the nodes in X are referred to as **external nodes** (or ext(F)). For each F, | ext(F) | = rank(A), where ext(F) only contains the leaves of F.*

A restricted DAG grammar is required to satisfy

	it				de				nl			
	P	R	F1	ill	P	R	F1	ill	P	R	F1	ill
all	65.91	67.99	66.93	**0%**	56.60	61.49	58.94	0.3%	56.53	60.37	58.38	0.4%
-pos	65.26	67.98	66.59	0.54%	57.18	61.57	59.29	**0.2%**	57.38	62.17	59.68	0.4%
-semtag	52.22	55.05	53.60	0.1%	46.79	51.90	49.20	0.7%	45.29	48.80	46.98	**0.4%**
-word	**70.29**	**72.16**	**71.21**	0.3%	**61.17**	**65.46**	**63.24**	0.3%	**63.11**	**66.44**	**64.73**	0.4%
-label	64.65	67.22	65.90	0.1%	55.65	61.06	58.22	0.4%	56.54	61.49	58.91	1%

Table 1: Feature ablation experiments when training in English and testing in Italian (*it*), german (*de*) and Dutch (*nl*).

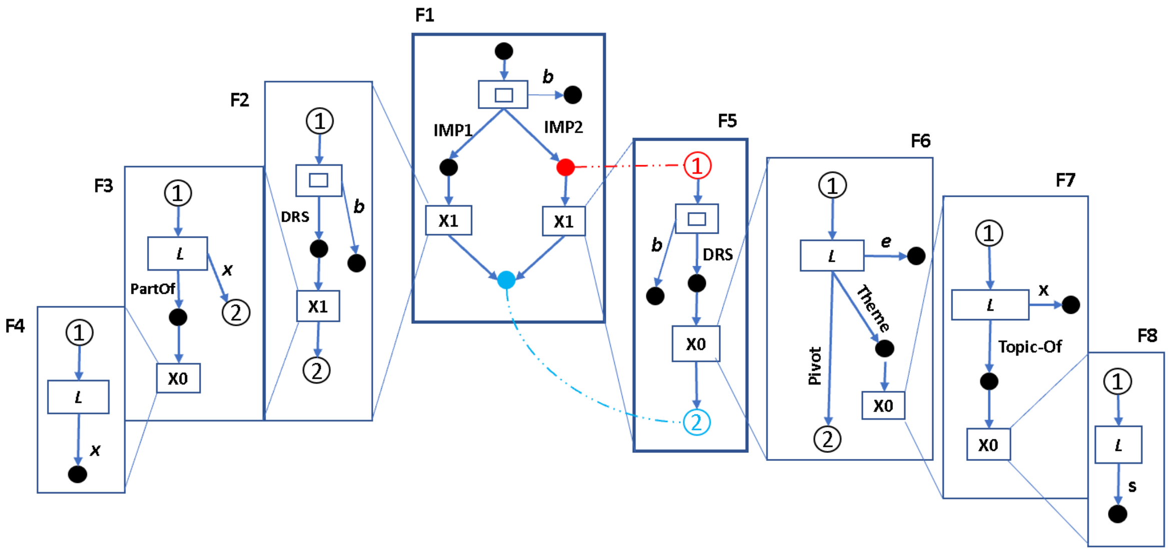

Figure 2: Grammar fragments extracted from the hypergraph in Figure 1.

the following conditions:

1. If a node v has in-degree larger than one, then v is a leaf.
2. For every non-terminal e in F, all nodes in tar(e) are leaves.
3. If a leaf of F has in-degree exactly one, then it is an external node or its unique incoming edge is a terminal.
4. The leaves of F are ordered and ext(F) follows this order.

Grammar productions are in **normal form** in that F has a single edge e which is a terminal or F has height 2, where one edge e (with src(e)=root(F)) is a terminal and all others are non-terminals.

Figure 2 shows the 8 grammar fragments extracted from the hypergraph in Figure 1 where each fragment corresponds closely to the RHS of the productions shown in §. 2 of the main paper. To better illustrate this comparison, we use F_1 and F_5 as examples which correspond to the following productions respectively:

$$T_0 \rightarrow (b/\square \text{ :IMP}_1\ T_1(\$1) \text{ :IMP}_2\ T_1(\$1))$$
$$T_1(\$1) \rightarrow (b/\square \text{ :DRS } T_1(\$1))$$

Each fragment is made of terminal and non-terminal hyperedges, the latter comprising of non-terminal symbol (labelled with an 'X'), each specifying the number of connected external nodes – the *rank*. There is a direct mapping between non-terminal hyperedges and non-terminal functions in the string rewriting system, where variable reference is expressed by shared target nodes. This is the case of F_1 where the variable reference (the node in cyan) is the target node of both non-terminals 'X1'.

Rewriting consists in substituting fragments in place of a non-terminal hyperedge, as shown in the case of F_1 and F_5. This mirrors the second production above where the variable reference \$1 is represented in F_5 via an *external node*, which during the derivation will then be bound to the variable x in F_3. External nodes determine how the RHS will substitute in the LHS. The source node of any RHS by convention plugs into the source of a non-terminal hyperedge, as shown in red in Figure 2. Targets nodes of the RHS plugs in the target nodes of the LHS in clockwise order, following the numbering on the external nodes, as shown in the same figure in cyan.

References

Björklund, H., Drewes, F., and Ericson, P. (2016). Between a rock and a hard place–uniform parsing for hyperedge replacement dag grammars. In *International Conference on Language and Automata Theory and Applications*, pages 521–532. Springer.