Saarland at MRP 2019: Compositional parsing across all graphbanks Supplementary Materials

Lucia Donatelli, Meaghan Fowlie*, Jonas Groschwitz, Alexander Koller, Matthias Lindemann, Mario Mina, Pia Weißenhorn
Department of Language Science and Technology, Saarland University
* Department of Linguistics, Utrecht University
{donatelli|jonasg|koller|mlinde|mariom|piaw}@coli.uni-saarland.de
m.fowlie@uu.nl

Activation function	tanh
Optimizer	Adam
Learning rate	0.001
Dim of lemma embeddings	64
Dim of POS embeddings	32
Dim of NE embeddings	16
Minimum lemma frequency	7
Hidden layers in all MLPs	1
Hidden units in LSTM (per direction)	256
Hidden units in edge existence MLP	*300*
Hidden units in edge label MLP	*300*
Hidden units in supertagger MLP	1024
Hidden units in lexical label tagger MLP	1024
Layer dropout in LSTMs	0.3
Recurrent dropout in LSTMs	0.4
Input dropout	0.3
Dropout in edge existence MLP	0.0
Dropout in edge label MLP	0.0
Dropout in supertagger MLP	0.4
Dropout in lexical label tagger MLP	0.4
Elmo Dropout	*0.1*

Table 1: Hyperparameters used in all experiments, changes to the ACL-19 parser in italics.

A Supplemental Material

A.1 Hyperparameters

Table 1 shows the hyperparameters used in our experiments. We train our models for 45 epochs with batch size of 64. We noticed that the UCCA model needed more time to converge due the small amount of available data and trained it for 60 epochs. We employed early stopping with patience of 15 epochs. To avoid spending a lot of time on the evaluation after each epoch, we set the parsing timeout (see section 5.1) to 15 seconds. The stopping criterion was the MRP F-score, except on EDS, where we used Smatch (Cai and Knight, 2013) because computing the MRP F-score took an unreasonable amount of time.

Label	**To Origin**	**Source**
A	✓	a
C	✓	op
F	✗	aux
H	✓	a
U	✗	pnct
P,S,N,L,R,LR,LA	✓	M
E,D,R,G	✗	M

Table 2: Heuristics for UCCA.

L, LR, LA, H, P, S, A, N, C, D, T, E, R, F, G, Q, U

Table 3: Head percolation precedence rules for UCCA

A.2 UCCA heuristics

Table 2 shows our edge attachment and source assignment heuristics for UCCA. The heuristics are broken down by the edge's label in the 'Label' column ('*' is a wildcard matching any string). A checkmark (✓) in the 'To Origin' column means that all edges with this label are attached to their origin node, a cross (✗) means the edge is attached to its target node.

A.3 UCCA head percolation

In order to align the tokens of the string with all of the nodes in the graph, we use head percolation to project the aligned tokens upwards; we iterate over the edge labels as shown in table 3. If the aligned leaf node can percolate upwards via an edge with a label in the order shown in table 3, we align the upper node with the leaf node. We perform the head percolation until all the nodes in the graph are aligned to a leaf node.

A.4 Postprocessing AMR: restoring properties

During postprocessing we have to convert some edges back to properties. Since properties are non-recursive, we only have to take a closer look at those edges-in-question where the target node (pos-

sible property value) has rank 1. Moreover a target node with a source annotation cannot be a property value. For each remaining edge-in-question we apply a cascade of very simple regular expression rules based on the edge label and the labels of the two incident nodes. If none of these rules apply we don't assume a property. The high-precision rules were hand-crafted by looking at the train and dev set. See table 4 for the rules. One of these rules, for instance, filters out all the edges-in-question with a number or URL as target node label because these are typical values of properties but never appear as true node labels. Another rule is responsible for 'op' edges starting from a 'name' node since named entities are usually represented as properties of a name node (like `name :op1 "Pierre" :op2 "Vinken"`).

References

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Rules	matching	notes
`"[a-z]+-[0-9][0-9]", "amr-unknown", "truth-value"`	target node label	assume these never appear as a property value (first regex is for wordsense)
`"-?[0-9]+([.:/][0-9]+)?(:[0-9][0-9])?", "http.*", "www[.].*", "-"`	target node label	integers, floats, time points, and URLs assumed to be property values always
`"li", "polite", "mode", "year[0-9]*", "month", "day", "quarter", "decade", "century", "era", "polarity", "value", "timezone"`	edge label	in at least 80% of the cases when the target of the relation is a leaf, these labels are properties
ends with `"-quantity"` or `"-entity"`	target node label	not a property value: temporal-quantity, monetary-quantity, date-entity, ordinal-entity, percentage-entity ...
`"name"`	source node label	named entities: always properties
`"op[0-9]*"`	edge label	

Table 4: Rules for identifying properties in AMR postprocessing. The comma in the regular expression is to be interpreted as disjunction.