HIT-SCIR at MRP 2019: A Unified Pipeline for Meaning Representation Parsing via Efficient Training and Effective Encoding

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, Ting Liu
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{car,lxdou,yxu,yxwang,yjliu,tliu}@ir.hit.edu.cn

A Training Details

A.1 Fine-tuning BERT with Parser

We find it beneficial to warm up learning rate at beginning of training progress and cool down after. With slanted triangular learning rate scheduler, the learning rate increases linearly from $lr/ratio$ to lr during the first $num\_step \times cut\_frac$ steps and decreases linearly back to $lr/ratio$ during the left steps.

Gradual unfreezing is also used during training so in the first few($1 \sim 5$) epochs BERT parameters are frozen. While being gradual unfrozen, the learning rate experiences a full warm-up and cool-down cycle per epoch. And then a full cycle is performed during the rest training progress once all parameters are unfrozen.

HYPERPARAMETER	VALUE
Hidden dimension	200
Action dimension	50
Optimizer	Adam
β_1, β_2	0.9, 0.99
Dropout	0.5
Layer dropout	0.2
Recurrent dropout	0.2
Input dropout	0.2
Batch size	16
Epochs	50
Base learning rate	1×10^{-3}
BERT learning rate	5×10^{-5}
Gradient clipping	5.0
Gradient norm	5.0
Learning rate scheduler	slanted triangular
Gradual Unfreezing	True
Cut Frac	0.1
Ratio	32

Table 1: A summary of model hyperparameters.

A.2 Hyperparameters

AMR model adopts the BERT-large cased (whole word masking) pre-trained weights while other models adopt the BERT-base cased pre-trained weights. In preliminary experiment on split dataset, we did not get the obvious improvement using BERT-large in DM, PSD, AMR and UCCA, thus we use BERT-base simply.

A.3 Model Selection

When training the final system for submission, we use all data and through observing the loss curve to find the first sub-optimal point of loss curve.

A.4 Decoding Constrain

We need to generate concept node in UCCA, EDS and AMR. During the early phase in training, our model may generate too many concept nodes. In such cases, we add the following decoding constraint: a) **Transition Step** We limit the transition step to ten times the length of the sentence. b) **Concept Node** We limit the number of generated concept node to ten times the length of the sentence.

During decoding, our system avoids infinite loops caused by incomplete training or incorrect transition actions.

B Transition Systems

Table 2, Table 3, Table 4, and Table 5 shows the transition set for DM/PSD, UCCA, EDS, and AMR respectively.

C Ensemble

Due to time constraints, only a single model can be submitted. Since the organizers has already released a subset of test set (so called 'lpps'), we decide to explore how much ensemble can benefit. Results are shown in Table 6.

Before Transition					Transition	After Transition						Condition
Stack	**List**	**Buffer**	**Nodes**	**Edges**		**Stack**	**List**	**Buffer**	**Nodes**	**Edges**	**Terminal?**	
S	L	$x \mid B$	V	E	SHIFT	$S \mid L \mid x$	$\emptyset$	B	V	E	$-$	
$S \mid x$	L	B	V	E	REDUCE	S	L	B	V	E	$-$	
$S \mid x$	L	$y \mid B$	V	E	RIGHT-EDGE$_X$	$S \mid x$	L	$y \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	
$S \mid y$	L	$x \mid B$	V	E	LEFT-EDGE$_X$	$S \mid y$	L	$x \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	
$S \mid x$	L	B	V	E	PASS	S	$x \mid L$	B	V	E	$-$	
S	$\emptyset$	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	$\emptyset$	V	E	$+$	

Table 2: The transition set of SDP (DM,PSD) parser. We write the **stack** with its top to the right, the **buffer** with its head to the left and the **list** with its head to the left.

Before Transition				Transition	After Transition					Condition
Stack	**Buffer**	**Nodes**	**Edges**		**Stack**	**Buffer**	**Nodes**	**Edges**	**Terminal?**	
S	$x \mid B$	V	E	SHIFT	$S \mid x$	B	V	E	$-$	
$S \mid x$	B	V	E	REDUCE	S	B	V	E	$-$	
$S \mid x$	B	V	E	NODE$_X$	$S \mid x$	$y \mid B$	$V \cup \{y\}$	$E \cup \{(y,x)_X\}$	$-$	$x \neq \text{root}$
$S \mid y, x$	B	V	E	LEFT-EDGE$_X$	$S \mid y, x$	B	V	$E \cup \{(x,y)_X\}$	$-$	$\left\{ \begin{array}{l} x \notin w_{1:n}, \\ y \neq \text{root}, \\ y \not\leadsto_G x \end{array} \right.$
$S \mid x, y$	B	V	E	RIGHT-EDGE$_X$	$S \mid x, y$	B	V	$E \cup \{(x,y)_X\}$	$-$	
$S \mid y, x$	B	V	E	LEFT-REMOTE$_X$	$S \mid y, x$	B	V	$E \cup \{(x,y)_X^*\}$	$-$	
$S \mid x, y$	B	V	E	RIGHT-REMOTE$_X$	$S \mid x, y$	B	V	$E \cup \{(x,y)_X^*\}$	$-$	
$S \mid x, y$	B	V	E	SWAP	$S \mid y$	$x \mid B$	V	E	$-$	$\mathrm{i}(x) < \mathrm{i}(y)$
[root]	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	V	E	$+$	

Table 3: The transition set of UCCA parser. We write the **stack** with its top to the right and the **buffer** with its head to the left. $(\cdot,\cdot)_X$ denotes a primary X-labeled edge, and $(\cdot,\cdot)_X^*$ a remote X-labeled edge. $\mathrm{i}(x)$ is a running index for the created nodes. In addition to the specified conditions, the prospective child in an EDGE transition must not already have a primary parent. From (Hershcovich et al., 2017).

Before Transition					Transition	After Transition						Condition
Stack	**List**	**Buffer**	**Nodes**	**Edges**		**Stack**	**List**	**Buffer**	**Nodes**	**Edges**	**Terminal?**	
S	L	$x \mid B$	V	E	SHIFT	$S \mid L \mid x$	$\emptyset$	B	V	E	$-$	concept(x)
$S \mid x$	L	B	V	E	REDUCE	S	L	B	V	E	$-$	
$S \mid x$	L	$y \mid B$	V	E	RIGHT-EDGE$_X$	$S \mid x$	L	$y \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	concept(x) $\wedge$ concept(y)
$S \mid y$	L	$x \mid B$	V	E	LEFT-EDGE$_X$	$S \mid y$	L	$x \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	concept(x) $\wedge$ concept(y)
$S \mid x$	L	B	V	E	PASS	S	$x \mid L$	B	V	E	$-$	
S	L	$x \mid B$	V	E	DROP	$S \mid L$	$\emptyset$	B	V	E	$-$	token(x)
S	L	$x \mid B$	V	E	TOP	S	L	$x \mid B$	$V \cup$ Top(x)	E	$-$	concept(x)
S	L	$x \mid B$	V	E	NODE-START$_X$	$S \mid y$	L	$x \mid B$	$V \cup \{y_{start=x,label=X}\}$	E	$-$	token(x)
$S \mid y$	L	$x \mid B$	V	E	NODE-END	$S \mid y$	L	$x \mid B$	$V \cup \{y_{end=x}\}$	E	$-$	token(x)
[root]	$\emptyset$	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	$\emptyset$	V	E	$+$	

Table 4: The transition set of EDS parser. The elements in **stack** and **list** are all concept node. Indicator function token(x) means x is a token of the sentence , while concept(x) means it's a concept node. Top(x) indicates x is the top node. $y_{start=w_i,label=X,end=w_j}$ indicates the alignments of concept node y is starting at token w_i, ending at token w_j and label is X.

Before Transition					Transition	After Transition						Condition
Stack	**List**	**Buffer**	**Nodes**	**Edges**		**Stack**	**List**	**Buffer**	**Nodes**	**Edges**	**Terminal?**	
S	L	$x \mid B$	V	E	SHIFT	$S \mid L \mid x$	$\emptyset$	B	V	E	$-$	concept(x)
$S \mid x$	L	B	V	E	REDUCE	S	L	B	V	E	$-$	
$S \mid x$	L	$y \mid B$	V	E	RIGHT-EDGE$_X$	$S \mid x$	L	$y \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	concept(y)
$S \mid y$	L	$x \mid B$	V	E	LEFT-EDGE$_X$	$S \mid y$	L	$x \mid B$	V	$E \cup \{(x,y)_X\}$	$-$	concept(x)
$S \mid x$	L	B	V	E	PASS	S	$x \mid L$	B	V	E	$-$	
S	L	$x \mid B$	V	E	DROP	S	L	B	V	E	$-$	token(x)
S	L	$x \mid y \mid B$	V	E	MERGE	S	L	$x\_y \mid B$	V	E	$-$	token(x) $\wedge$ token(y)
S	L	$x \mid B$	V	E	ENTITY$_X$	S	L	$X \mid B$	$V \cup$ Attribute$_X$	$E \cup$ Relation$_X$	$-$	token(x)
S	L	$x \mid B$	V	E	CONFIRM$_X$	S	L	$X \mid B$	V	E	$-$	token(x)
S	L	$x \mid B$	V	E	NEW$_X$	S	L	$X \mid x \mid B$	V	E	$-$	concept(x)
[root]	$\emptyset$	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	$\emptyset$	V	E	$+$	

Table 5: The transition set of AMR parser. Indicator function token(x) means x is a token of the sentence, while concept(x) means it's a concept node. Attribute$_X$ and Relation$_X$ indicates properties nodes of entity X and edges from X to its properties, respectively.

	DM	PSD	EDS	UCCA	AMR
official all	94.64	89.66	90.75	81.67	72.94
official lpps	93.98	87.41	**89.83**	82.61	69.03
5-ensemble lpps	**94.00**	**87.79**	89.57	**83.41**	**71.35**
model-1	94.62	87.59	<u>89.83</u>	83.60	<u>70.30</u>
model-2	94.73	87.70	87.87	82.34	69.42
model-3	<u>94.80</u>	87.43	87.72	83.12	68.99
model-4	94.66	<u>87.98</u>	88.02	<u>83.80</u>	68.18
model-5	93.83	87.80	86.84	82.80	69.92

Table 6: Results of ensemble model. Top part contains final results while bottom part contains results of single models with best single result being underlined. Official all means result of our submitted model on the full test set. Official lpps and 5-ensemble lpps means result of our submitted model and ensemble model on the subset of test set, respectively.

References

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *ACL*.