JBNU at MRP 2019: Multi-level Biaffine Attention for Semantic Dependency Parsing

Seung-Hoon Na[†], **Jinwoo Min**[†], **Kwanghyeon Park**[†], **Jong-Hun Shin**[‡] and **Young-Kil Kim**[‡]
[†]Computer Science and Engineering, Jeonbuk National University, South Korea
[‡]Electronics and Telecommunication Research Institute (ETRI), South Korea
nash@jbnu.ac.kr, jinwoomin4488@gmail.com, hpk23@naver.com,
{jhshin82,kimyk}@etri.re.kr

Abstract

This paper describes Jeonbuk National University (JBNU)'s system for the 2019 shared task on Cross-Framework Meaning Representation Parsing (*MRP 2019*) at the Conference on Computational Natural Language Learning. Of the five frameworks, we address only the DELPH-IN MRS Bi-Lexical Dependencies (DP), Prague Semantic Dependencies (PSD), and Universal Conceptual Cognitive Annotation (UCCA) frameworks. We propose a unified parsing model using *biaffine attention* (Dozat and Manning, 2017), consisting of 1) a *BERT-BiLSTM* encoder and 2) a biaffine attention decoder. First, the BERT-BiLSTM for sentence encoder uses BERT to compose a sentence's wordpieces into word-level embeddings and subsequently applies BiLSTM to word-level representations. Second, the biaffine attention decoder determines the scores for an edge's existence and its labels based on biaffine attention functions between *role*-dependent representations. We also present *multi-level* biaffine attention models by combining all the role-dependent representations that appear at multiple intermediate layers.

1 Introduction

Recent studies on meaning representation parsing (MRP) have focused on different semantic graph frameworks such as bilexical semantic dependency graphs (Peng et al., 2017; Wang et al., 2018; Peng et al., 2018; Dozat and Manning, 2018), universal conceptual cognitive annotation (Hershcovich et al., 2017, 2018), and abstract meaning representation (Wang and Xue; Guo and Lu; Song et al., 2019; Zhang et al., 2019). To jointly address various semantic graphs, the aim of the Cross-Framework MRP task (MRP 2019) at the 2019 Conference on Computational Natural Language Learning (CoNLL) is to develop semantic graph parsing across the following five frameworks (Oepen et al., 2019): 1) **DM**: DELPH-IN MRS Bi-Lexical Dependencies (Ivanova et al., 2012), 2) **PSD**: Prague Semantic Dependencies (Hajič et al., 2012; Miyao et al., 2014), 3) **EDS**: Elementary Dependency Structures (Oepen and Lønning, 2006), 4) **UCCA**: Universal Conceptual Cognitive Annotation (Abend and Rappoport, 2013), and 5) **AMR**: Abstract Meaning Representation (Banarescu et al., 2013).

One of the main aims of MRP 2019 is to indduce a unified parsing model for different semantic frameworks such that parsing models can be trained using multi-task learning or transfer learning. To enable multi-task learning, we explicitly make *shared* common components in a neural network architecture across different frameworks. For MRP 2019, we propose a unified neural model for the DM/PSD/UCCA frameworks based on the *biaffine attention* used in (Dozat and Manning, 2017, 2018; Zhang et al., 2019) by deploying the sentence encoder part as a "shared" component across these three frameworks. Our system consists of two main components:

1. **BERT-BiLSTM sentence encoder** (*shared* across frameworks): Given a sentence, the BERT encoder (Devlin et al., 2019) encodes to its wordpieces and the encoded word piece-level represenations are composed into word-level embeddings based on BiLSTM. Another BiLSTM layer is then applied to the resulting word-level embeddings to create the final sentence representations. We refer to this neural layer for encoding sentences as the *BERT-BiLSTM sentence encoder*. For multi-task learning, the BERT-BiLSTM sentence encoder is shared across all target frameworks.

2. **Biaffine attention decoder** (*framework-specific*): Role-dependent representations

Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 CoNLL, pages 95–103
Hong Kong, November 3, 2019. ©2019 Association for Computational Linguistics
https://doi.org/10.18653/v1/K19-2009

for each word are first induced from the sentence-level embeddings of the BERT-BiLSTM encoder using simple feed-forward layers. Biaffine attention is then performed on the resulting role-dependent representations to predict the existence of an edge and its labels. However, the biaffine attention decoder is not shared but separately trained for each framework. Thus, we have three different biaffine decoders corresponding to DM, PSD, and UCCA.

In addition, our system handles the following specific issues for UCCA parsing and node property prediction:

1. *UCCA parsing using biaffine attention* To handle UCCA formats using a biaffine attention model, we convert a UCCA graph to a bilexical framework using the `semstr` tool, which is based on the head rules of UCCA in (Hershcovich et al., 2017). [1] After the biaffine attention is performed, the parsed bilexical graph is converted back to the UCCA format.

2. *BiLSTM neural models for node property prediction*: In addition to predicting the existence and labels of an edge, the system is required to predict node properties (for DM and PSD). To handle node properties, we further develop *property-specific* BiLSTM-based neural models.[2] These property-specific neural components are designed in a framework-specific manner and are not shared across frameworks.

Furthermore, we present *multi-level* biaffine attention models, motivated by the multi-level architecture of FusionNet in the machine reading comprehension task (Huang et al., 2018).

The preliminary unofficial experiments using our own development seting show that multi-task learning is helpful in improving UCCA's performance, but it does not lead to improvement in performances on the DM and PSD frameworks.

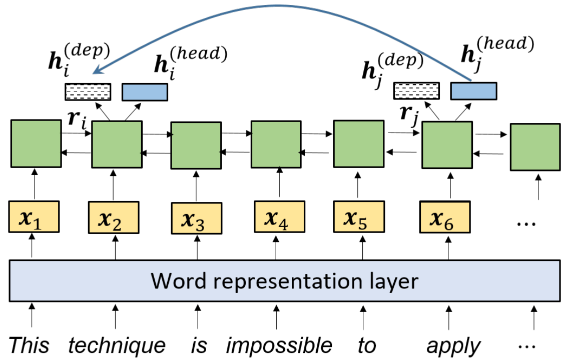

Figure 1: Biaffine attention for bilexical semantic dependency parsing based on word representation using BERT, Glove and POS embeddings.

The remainder of this paper is organized as follows: Section 2 presents our system architecture with details, Section 3 describes the detailed process for training biaffine attention models. Section 4 and 5 provide the preliminary experiment results and the official results at MRP 2019, respectively, and our concluding remarks and a description of future work are given in Section 6.

2 Model

Figure 1 shows the neural architecture based on biaffine attention for bilexical semantic dependency parsing. The neural architecture consists of two components: 1) the BERT-BiLSTM encoder and 2) the biaffine attention decoder. 1) In BERT-BiLSTM encoder, an input sentence is fed to a word representation layer using BERT, resulting in a sequence of word embedding vectors, which are then given to the BiLSTM layer to produce a sentence representation. 2) In biaffine attention, additional feed-forward layers are applied to obtain *role*-dependent representations for head and dependent roles, which are then forwarded to the biaffine attention.

2.1 Encoder: BERT-BiLSTM

2.1.1 Word representation layer using BERT

The word representation using BERT uses BiLSTM for composing to word-level embeddings from wordpiece-level embeddings, similar to (Zhang et al., 2019), which used the average pooling for composition. Specifically, suppose that an input sentence consists of n words, i.e., $x_1 \cdots x_n$. To obtain the word representation $\mathbf{x}_i$ for x_i, we use BERT from (Devlin et al., 2019), as shown in Figure 2. An input sentence is segmented into word-

[1] We first converted a UCCA MRP format to its xml format and then applied the converter (`semstr/convert.py`) in `semstr` to obtain its CoNLL format: https://github.com/danielhers/semstr

[2] The node properties required for DM and PSD are a POS tag and a *frame*. We prepared a BiLSTM neural model for predicting the frame information of a node only, whereas we used the companion data of MRP 2019 to predict POS tags.

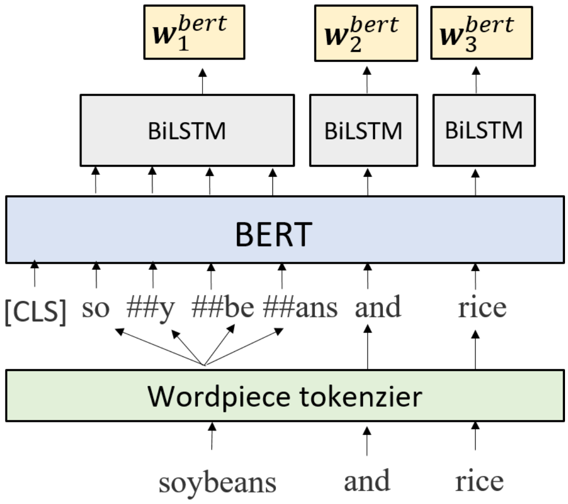

Figure 2: BERT Word embedding using Bi-LSTM.

pieces and they are fed to the BERT encoder. The resulting output from BERT, which consists of the word pieces in the i-th word are aggregated using BiLSTM, producing $\mathbf{w}_i^{bert}$, named *BERT word-level embedding*.[3]

The BERT word-level embedding is further combined with the pretrained GloVe word embedding of (Pennington et al., 2014) and part-of-speech (POS) tag embedding to produce the final word representation, as follows:

$$\mathbf{x}_i = \left[\mathbf{w}_i^{bert}; \mathbf{e}_i^{glove}; \mathbf{e}_i^{POS}\right]$$

where $\mathbf{e}_i^{glove}$ and $\mathbf{e}_i^{POS}$ denote the pretrained GloVe word embedding and the POS tag embedding for the i-th word, respectively.

2.1.2 BiLSTM sentence encoding layer

Once word representations are obtained, we further apply BiLSTM to $\mathbf{x}_1 \cdots \mathbf{x}_n$ to obtain the following initial hidden representation of the i-th word:

$$\mathbf{r}_i = BiLSTM_i\left(\mathbf{x}_1 \cdots \mathbf{x}_n\right)$$

where $BiLSTM_i$ refers to the i-th hidden representation obtained by applying BiLSTM to a given sequence.

2.2 Decoder: Biaffine attention

To formulate a decoder using biaffine attention, let $BiAff(x, y)$ be a biaffine function using the notations of (Dozat and Manning, 2018) and (Socher et al., 2013) as follows:

$$BiAff_m(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{U}^{[1:m]}\mathbf{y} + \mathbf{V}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} + \mathbf{b}$$

where $\mathbf{U}^{[1:k]} \in \mathbb{R}^{d\times d\times m}$ is a tensor, $\mathbf{x}^T\mathbf{U}^{[1:m]}\mathbf{y}$ produces vector $\mathbf{r} \in \mathbb{R}^k$, $\mathbf{V} \in \mathbb{R}^{m\times d}$ is a matrix and $\mathbf{b} \in \mathbb{R}^m$ is a vector for the bias term.

Our biaffine attention decoder is similar to that of (Dozat and Manning, 2018) and is formulated as follows:

$$\begin{aligned}
FFN\left(\mathbf{x}\right) &= f\left(\mathbf{A}\mathbf{x} + \mathbf{b}\right)\\
\mathbf{h}_i^{(head)} &= FFN^{(head)}\left(\mathbf{r}_i\right)\\
\mathbf{h}_i^{(dep)} &= FFN^{(dep)}\left(\mathbf{r}_i\right)\\
\mathbf{h}_i^{(l\text{-}head)} &= FFN^{(l\text{-}head)}\left(\mathbf{r}_i\right)\\
\mathbf{h}_i^{(l\text{-}dep)} &= FFN^{(l\text{-}dep)}\left(\mathbf{r}_i\right)\\
s_{i,j}^{(edge)} &= BiAff_1^{(edge)}\left(\mathbf{h}_i^{(dep)}, \mathbf{h}_j^{(head)}\right)\\
\mathbf{s}_{i,j}^{(label)} &= BiAff_k^{(label)}\left(\mathbf{h}_i^{(l\text{-}dep)}, \mathbf{h}_j^{(l\text{-}head)}\right)\\
s_i^{(top)} &= FFN^{(top)}\left(\mathbf{r}_i\right) \qquad (1)
\end{aligned}$$

where k is the number of node labels, and f is the activation function used in the feed-forward layer FFN.[4]

In contrast to the setting of (Dozat and Manning, 2018), the top score $s_i^{(top)}$ is newly introduced in our model, where we exploit a simple feed-forward layer for predicting top nodes instead of using an attention method.

Using the score functions of Eq. (1), the prediction results for arcs, labels, and top nodes are formulated as follows:

$$\begin{aligned}
y_{i,j}^{(edge)} &= \mathcal{I}\left(s_{i,j}^{edge} \geq 0\right)\\
y_{i,j}^{(label)} &= argmax\left\{\mathbf{s}_{i,j}^{(label)}\right\}\\
y_i^{(top)} &= \mathcal{I}\left(s_i^{(top)} \geq 0\right) \qquad (2)
\end{aligned}$$

where $\mathcal{I}(expr)$ is an indicator function which gives 1 if $expr$ is true and 0 otherwise.

2.3 Multi-level Biaffine attention

We also investigated a *multi-level* biaffine attention, whose information flow is described in Figure 3. Motivated by (Huang et al., 2018), we assume that multi-layer encoders gradually transform from a low-level word representation into a

[3] This aggregation is similar to the BiLSTM-based composition in (Ballesteros et al., 2015; Na et al., 2018) which uses characters as subtokens, whereas our aggregation uses word pieces as subtokens.

[4] In our submission, we used the identity function for f.

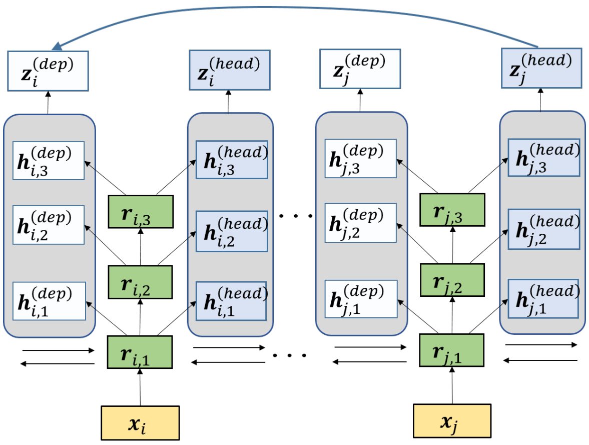

Figure 3: The neural architecture of multi-level biaffine attention. The hidden representations at three levels $\mathbf{h}_{i,k}^{(dep)}$ and $\mathbf{h}_{j,k}^{(head)}$ are composed to the final hidden representation $\mathbf{z}_i^{(dep)}$ and $\mathbf{z}_j^{(head)}$, respectively.

more abstract high-level representation. In the task of semantic graph parsing, predicting an arc and a label may be resolved not just by single-level representation but by the combination of various levels of representations; For example, predicting an arc between two *deep* semantic subgraphs (with high depths) may require more abstract representations for those graphs than the case of predicting an arc between two *shallow* semantic subgraphs (with low depths).

The multi-level biaffine attention is based on the *fusion* of all the *role*-dependent representations across levels.[5] This type of multi-level attention is different from deep biaffine attention of (Dozat and Manning, 2018), which uses only single role-dependent hidden representation at the final level.

To formulate the multi-level biaffine attention, we first apply deep BiLSTM encoder of L-levels to a list of word embeddings $\mathbf{x}_1, \cdots, \mathbf{x}_n$ as follows.

$$\begin{aligned}\mathbf{r}_{i,0} &= \mathbf{x}_i \\ \mathbf{r}_{i,l} &= BiLSTM_i\left(\mathbf{r}_{1,l-1} \cdots \mathbf{r}_{n,l-1}\right)\end{aligned}$$

where $\mathbf{r}_{i,l}$ is the hidden representation of the BiLSTM at the l-th layer.

The role-dependent representation for each l-th layer is formulated as follows:

$$\begin{aligned}\mathbf{h}_{i,l}^{(head)} &= FFN^{(head)}\left(\mathbf{r}_{i,l}\right) \\ \mathbf{h}_{i,l}^{(dep)} &= FFN^{(dep)}\left(\mathbf{r}_{i,l}\right)\end{aligned}$$

To aggregate all the role-dependent representations, we use the *fusion* function, denoted as $\mathbf{o} = fusion(\mathbf{x}, \mathbf{y})$, as defined in (Hu et al., 2018):

$$\begin{aligned}\tilde{\mathbf{x}} &= gleu\left(\mathbf{W}_r\left[\mathbf{x}; \mathbf{y}; \mathbf{x} \odot \mathbf{y}; \mathbf{x} - \mathbf{y}\right]\right) \\ \mathbf{g} &= \sigma\left(\mathbf{W}_g\left[\mathbf{x}; \mathbf{y}; \mathbf{x} \odot \mathbf{y}; \mathbf{x} - \mathbf{y}\right]\right) \\ \mathbf{o} &= \mathbf{g} \odot \tilde{\mathbf{x}} + (\mathbf{1} - \mathbf{g}) \odot \mathbf{x}\end{aligned}$$

where $\odot$ is element-wise multiplication. For notational simplicity, we further define $sfu(\mathbf{x}, \mathbf{y}, \mathbf{z})$, the fusion function that takes three arguments, as follows:

$$sfu\left(\mathbf{x}, \mathbf{y}, \mathbf{z}\right) = fusion\left(fusion\left(\mathbf{x}, \mathbf{y}\right), \mathbf{z}\right)$$

Applying the sfu function results in the compositional role-dependent representations $\mathbf{z}_i^{(head)}$ and $\mathbf{z}_i^{(dep)}$ at the i-th position. The multi-level biaffine attention is then defined on $\mathbf{z}_i^{(head)}$ and $\mathbf{z}_i^{(dep)}$ as follows:

$$\begin{aligned}\mathbf{z}_i^{(head)} &= sfu^{(head)}\left(\mathbf{h}_{i,1}^{(head)}, \mathbf{h}_{i,2}^{(head)}, \mathbf{h}_{i,3}^{(head)}\right) \\ \mathbf{z}_i^{(dep)} &= sfu^{(dep)}\left(\mathbf{h}_{i,1}^{(dep)}, \mathbf{h}_{i,2}^{(dep)}, \mathbf{h}_{i,3}^{(dep)}\right) \\ s_{i,j}^{(edge')} &= BiAff_1^{(edge')}\left(\mathbf{z}_i^{(dep)}, \mathbf{z}_j^{(head)}\right) \qquad (3)\end{aligned}$$

Similar to the arc scores in Eq. (3), we straightforwardly define multi-level terms related to label scores such as $\mathbf{h}_{i,k}^{(l\text{-}dep)}$, $\mathbf{h}_{i,k}^{(l\text{-}head)}$, $\mathbf{z}_i^{(l\text{-}head)}$, and $\mathbf{z}_i^{(l\text{-}dep)}$.

2.4 Property prediction based on BiLSTM

To predict *frame* information, which is one of the node properties in DM and PSD, we use a simple BiLSTM architecture with a single output layer that generates a node property for each word.[6]

Different from the biaffine attention model, the property predictor does not use BERT but a simple word representation that consists of the pretrained GloVe and the POS tag embedding as follows:

$$\mathbf{x}_i^{(prop)} = \left[\mathbf{e}_i^{glove}; \mathbf{e}_i^{POS}\right]$$

For encoding a sentence, another BiLSTM is then applied to the sequence of word representations, as follows:

$$\mathbf{r}_i^{(prop)} = BiLSTM_i^{(prop)}\left(\mathbf{x}_1 \cdots \mathbf{x}_n\right)$$

[5] A pair of syntactic roles in role-dependent representations are considered – head-dependent roles (or predicate-argument roles).

[6] Here, words (or tokens) correspond to nodes in a semantic graph.

The output layer uses the following simple affine transformation:

$$\mathbf{s}_i^{(prop)} = FFN^{(prop)}\left(\mathbf{r}_i^{(prop)}\right) \tag{4}$$

The loss function uses the cross entropy, which is formulated given a single training sentence as follows:

$$L^{(prop)} = \sum_i \log softmax_{g(i)}\left(\mathbf{s}_i^{(prop)}\right) \tag{5}$$

where $g(i)$ is the gold property value of the i-th word and $softmax_k$ is the function of k-th element of softmax values.[7]

3 Training

3.1 Preprocessing

We use word tokens and their POS tags in the companion dataset provided by MRP 2019. To perform UCCA parsing using biaffine attention, conversion between UCCA and bilexical formats is required. For the conversion, we use the `semstr` tool, which is based on the head rules defined in (Hershcovich et al., 2017).

3.2 Multi-task learning on a single framework

In each semantic graph framework, the biaffine attention models consist of three subtasks – edge detection, edge labeling, and top node prediction. We jointly train the neural components of all the subtasks for each framework in the multi-task learning setting using the following combined loss function:

$$L = \lambda_1 L^{(edge)} + \lambda_2 L^{(label)} + \lambda_3 L^{(top)} \tag{6}$$

where $L^{(edge)}$, $L^{(label)}$, and $L^{(top)}$ are the loss functions for edge detection, edge labeling, and top node prediction, respectively, and λ_i is the weight for each loss function.

However, the property predictor of Section 2.4 is not jointly trained on a single framework because its neural components can be shared in any component in the biaffine attention models.

GloVe	
source	840B
dim	300
BERT layer	
source	BERT-Base-cased
dim	784
Word embedding layer: BiLSTM	
hidden_size	384
num_layers	1
Sentence encoder: BiLSTM	
hidden_size	600
num_layers	3
(Multi-level) Biaffine decoder	
hidden_size	600
Property predictor	
BiLSTM_hidden_size	600
BiLSTM_num_layers	3
output_vocab_size(DM)	474
output_vocab_size(PSD)	5474
Adam optimizer	
learning_rate	0.001
weight_decay_rate	3e-9
Adam β_1	0.0
Adam β_2	0.95
BERT Adam optimizer	
learning_rate	2e-5
weight_decay_rate	0.01
Adam β_1	0.9
Adam β_2	0.999
Loss for multi-task learning of Eq. (6)	
λ_1	0.025
λ_2	0.975
λ_3	1.0
batch_size	16

Table 1: Hyper-parameter settings

3.3 Multi-task learning across frameworks

To enable multi-task learning across frameworks, we share the BERT-BiLSTM encoder as a common neural component across three frameworks and use framework-specific models for the biaffine attention decoder. Our approach to multi-task learning is similar to that of SHARED1 of (Peng et al., 2017).

In multi-task learning, we alternate training examples for each framework using the framework-specific loss function of Eq. (6) such that, over each epoch, all the training examples across the three frameworks are fairly fed without bias to a specific framework.

3.4 Hyperparameters

We used Adam optimizer (Kingma and Ba, 2015) to train our biaffine attention models. Table 1 summarizes the hyper-parameters used for training these models

[7] We allow a NULL value to be a gold property value. Given this setting, the values of $g(i)$ are mostly NULL in the frame property of PSD.

Framework	Train	Dev
DM	32091	3565
PSD	32091	3565
UCCA	5915	656

Table 2: Statistics of dataset used in the preliminary experiment

4 Unofficial Results: Preliminary Experiment

In this section, we present the preliminary experimental results, which compare variants of our models. To perform the preliminary experiment, we randomly split the MRP 2019 dataset into training and development sets. Table 2 shows the statistics of training and development sets for the three frameworks.

The evaluation measures are unlabeled dependency F1 scores (**UF**), labeled dependency F1 scores (**LF**), and top node prediction accuracy (**Top**). We report the evaluation metrics for the development sets.

4.1 Experimental results

We evaluated the following four biaffine attention methods:

1. **Biaffine**: This model is the baseline biaffine attention model based on the BiLSTM sentence encoder without using BERT.

2. **BERT+Biaffine**: This model uses the BERT-BiLSTM encoder of Section 2.1 and the biaffine attention model of Section 2.2.

3. **BERT+Multi-level Biaffine**: This model uses BERT-BiLSTM encoder of Section 2.1 and uses the multi-level attention method of Section 2.3.

4. **BERT+Biaffine+MTL**: This model is the same as BERT+Biaffine but uses the multi-task learning across frameworks described in Section 3.3.

Table 3 shows the UF, LF, and Top on the three semantic graph frameworks, comparing the four variants of biaffine attention models. BERT+Biaffine performs better than Biaffine, in particular, obtaining the increases of about 5% for UF and LF on the UCCA framework. However, BERT+Multi-level Biaffine does not achieve any further improvements with respect to Biaffine, often yielding weak performances similar to that of the BERT-Biaffine model on the PSD and UCCA frameworks.

BERT+Biaffine+MTL only achieves small improvements on UCCA framework whereas no improvements on DM and PSD frameworks can be observed. A statistically insignificant improvement for multi-task learning in BERT+Biaffine+MTL was similarly reported in the results of SHARED1 in (Peng et al., 2017). These results imply that instead of naively using the shared encoder only, other advanced multi-task learning approaches such as placing task-specific encoding, as detailed in (Peng et al., 2017), need to be considered.

5 Official Results

Given the preliminary results, we chose the basic biaffine model “BERT+Biaffine” of Table 3 for the final submission to MRP 2019. The official results using BERT+Biaffine are summarized in Tables 4 and 5, which compare the results of ERG (Oepen and Flickinger, 2019) and TUPA (Hershcovich and Arviv, 2019) which were provided by the task organizer. Table 4 shows the performances of the *MRP metrics* on the three frameworks, whereas Table 5 presents the performances of *task-specific metrics* using the SDM metrics (Oepen et al., 2014) and UCCA metric (Hershcovich et al., 2019). The SDM metrics use the unlabeled dependency precision/recall/F1 (UP/UR/UF), the labeled dependency precision/recall/F1 (LP/LR/LF), and the unlabeled/labeled exact matches (UM/LM). The UCCA metrics use the unlabeled and labeled arc precision/recall/F1 for primary, remote and all types of arcs.[8]

Overall, our system shows better performances over the baseline TUPA's system, except for the results of UCCA metrics. Comparing to ERG which is the top-performing system in MRP metric on DM, our biaffine system shows slightly improved performance over ERG in terms of UF of the SDP metric. Comparing to the published MRP metrics of the best system (i.e. MRP all metric), the performances of our system are about 1.5 *percentage point* (p.p.) lower on DM framework, about 3.4

[8] Our system ranked fifth for framework-specific LF on DM and PSD, ranked eighth on UCCA, first for framework-specific UF using the 100-sentence LPPS sub-set, and second for LF on the PSD framework.

method	DM			PSD			UCCA		
	Top	UF	LF	Top	UF	LF	Top	UF	LF
Biaffine	93.67	92.08	90.86	95.97	90.50	78.21	72.60	69.67	65.17
BERT+Biaffine	95.06	93.85	93.00	96.89	92.30	80.24	77.09	74.85	70.15
BERT+Multi-level Biaffine	95.09	93.86	93.02	96.76	91.95	79.76	78.12	74.42	69.81
BERT+Biaffine+MTL	N/A	93.66	92.73	N/A	92.13	79.63	N/A	75.40	70.59

Table 3: Unofficial results of Top, UF, and LF metrics on the three frameworks (DM, PSD, and UCCA), comparing variants of biaffine attention models.

p.p. lower on PSD framework, and about 31 p.p. lower on UCCA framework.

6 Summary and Conclusion

In this paper, we presented the Jeonbuk National University's system based on unified biaffine attention models for DM, PSD, and UCCA frameworks for the MRP 2019 task. We investigated the extensions of the original biaffine models using multi-level biaffine attention and multi-task learning. The preliminary experiment results show that the use of multi-level models and multi-task learning had no effect on MRP performances under our current settings. The statistically insignificant results of multi-task learning imply that there may be some necessary conditions beyond the default setting to meet before multi-task learning with parameter sharing is effective. In this direction, we plan to explore why multi-task learning is not effective in our current experiment, try to postulate reasonable hypothesis that will help clarifying the effect of multi-task learning, and further examine other advanced multi-task learning including the approaches of (Peng et al., 2017). In addition, we would like to examine alternative fusion functions for multi-level affine attention.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement)

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL '13, pages 228–238. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15)*, pages 349–359.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations*, ICLR '17.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL'18, pages 484–490.

Zhijiang Guo and Wei Lu. Better transition-based AMR parsing with a refined search space. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English dependency treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, LREC-2012, pages 3153–3160.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL'17, pages 1127–1138.

method		tops			labels			properties			anchors			edges			all		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ERG	all	0.92	0.92	0.92	0.99	0.99	0.99	0.96	0.96	0.96	0.99	0.99	0.99	0.91	0.91	0.91	0.96	0.96	0.9608
	lpps	0.95	0.95	0.95	0.99	0.99	0.99	0.98	0.98	0.98	0.99	1.00	0.99	0.93	0.93	0.93	0.97	0.97	0.9731
TUPA	all	0.53	0.51	0.52	0.40	0.75	0.52	0.22	0.66	0.33	0.85	0.83	0.84	0.24	0.54	0.33	0.31	0.69	0.4270
	lpps	0.74	0.67	0.71	0.35	0.73	0.48	0.19	0.64	0.29	0.85	0.84	0.85	0.21	0.56	0.31	0.28	0.68	0.3946
BERT+Biaffine	all	0.92	0.92	0.92	0.91	0.90	0.90	0.91	0.95	0.94	0.95	0.99	0.98	0.99	0.92	0.91	0.94	0.94	0.9401
	lpps	0.96	0.96	0.96	0.88	0.88	0.88	0.91	0.92	0.91	0.98	0.98	0.98	0.93	0.92	0.92	0.92	0.92	0.9240

(a) The official results of MRP metrics on the DM framework

method		tops			labels			properties			anchors			edges			all		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TUPA	all	0.58	0.46	0.51	0.56	0.77	0.65	0.34	0.57	0.42	0.82	0.80	0.80	0.27	0.39	0.32	0.45	0.63	0.5265
	lpps	0.62	0.53	0.57	0.58	0.77	0.66	0.31	0.60	0.41	0.82	0.81	0.81	0.30	0.42	0.35	0.47	0.65	0.5453
BERT+Biaffine	all	0.96	0.96	0.96	0.86	0.85	0.86	0.88	0.88	0.88	0.99	0.98	0.99	0.79	0.78	0.78	0.88	0.88	0.88
	lpps	0.96	0.96	0.96	0.77	0.77	0.77	0.78	0.95	0.86	0.98	0.98	0.98	0.79	0.79	0.79	0.84	0.88	0.8568

(b) The official results of MRP metrics on the PSD framework

method		tops			anchors			edges			attributes			all		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TUPA	all	0.87	0.83	0.8492	0.90	0.52	0.6574	0.08	0.29	0.1299	0.10	0.08	0.0907	0.17	0.38	0.2365
	lpps	0.90	0.88	0.8889	0.93	0.67	0.7776	0.19	0.42	0.2645	0.28	0.14	0.1832	0.34	0.52	0.4104
BERT+Biaffine	all	0.91	0.91	0.9142	0.77	0.80	0.7833	0.33	0.28	0.3026	0.19	0.11	0.1405	0.53	0.49	0.5069
	lpps	0.91	0.91	0.9100	0.90	0.92	0.9126	0.47	0.42	0.4411	0.13	0.07	0.0882	0.66	0.62	0.6365

(c) The official results of MRP metrics on the UCCA framework

Table 4: The official results of MRP metrics on the three frameworks (DM, PSD, and UCCA), comparing ERG (Oepen and Flickinger, 2019), TUPA (Hershcovich and Arviv, 2019), and our system (BERT+Biaffine).

method		labeled				unlabeled			
		LP	LR	LF	LM	UP	UR	UF	UM
ERG	all	0.91	0.91	0.9121	0.5144	0.92	0.92	0.9204	0.5374
	lpps	0.93	0.93	0.9295	0.6900	0.93	0.94	0.9348	0.7200
TUPA	all	0.51	0.62	0.5623	0.0723	0.63	0.66	0.6430	0.0848
	lpps	0.50	0.63	0.5571	0.1400	0.62	0.67	0.6468	0.1700
BERT+Biaffine	all	0.92	0.90	0.9119	0.3998	0.93	0.92	0.9233	0.4329
	lpps	0.93	0.92	0.9265	0.5700	0.95	0.94	0.9413	0.6100

(a) The official results of SDP metrics on the DM framework

method		labeled				unlabeled			
		LP	LR	LF	LM	UP	UR	UF	UM
TUPA	all	0.47	0.53	0.5012	0.0863	0.65	0.67	0.6599	0.2200
	lpps	0.52	0.59	0.5533	0.1500	0.67	0.71	0.6876	0.2700
BERT+Biaffine	all	0.80	0.80	0.7998	0.1920	0.92	0.91	0.9164	0.4519
	lpps	0.82	0.81	0.8147	0.2800	0.93	0.93	0.9272	0.5500

(b) The official results of SDP metrics on the PSD framework

method		labeled									unlabeled								
		primary			remote			all			primary			remote			all		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
TUPA	all	0.30	0.19	0.23	0.08	0.06	0.07	0.28	0.19	0.22	0.37	0.23	0.28	0.09	0.06	0.07	0.35	0.22	0.27
	lpps	0.33	0.26	0.29	0.21	0.10	0.14	0.32	0.25	0.28	0.38	0.31	0.34	0.23	0.10	0.14	0.38	0.30	0.33
BERT+Biaffine	all	0.19	0.17	0.18	0.13	0.08	0.10	0.19	0.17	0.18	0.23	0.20	0.21	0.13	0.08	0.10	0.22	0.20	0.21
	lpps	0.35	0.32	0.34	0.04	0.02	0.03	0.34	0.31	0.33	0.41	0.39	0.40	0.04	0.02	0.03	0.40	0.37	0.38

(c) The official results of UCCA metrics on the UCCA framework

Table 5: The official results of task-specific metrics on the three frameworks, comparing ERG (Oepen and Flickinger, 2019), TUPA (Hershcovich and Arviv, 2019), and our system (BERT+Biaffine).

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 373–385.

Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elior Sulem, Ari Rappoport, and Omri Abend. 2019. SemEval-2019 task 1: Cross-lingual semantic parsing with UCCA. In *Proceedings of the 13th International Workshop on Semantic Evaluation*.

Daniel Hershcovich and Ofir Arviv. 2019. TUPA at MRP 2019: A multi-task baseline system. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 27 – 38, Hong Kong, China.

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI '18, pages 4099–4106.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *International Conference on Learning Representations*.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, LAW VI '12, pages 2–11.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, ICLR '13.

Yusuke Miyao, Stephan Oepen, and Daniel Zeman. 2014. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, SemEval 2014.

Seung-Hoon Na, Jianri Li, Jong hoon Shin, and Kangil Kim. 2018. Transition-based Korean dependency parsing using hybrid word representations of syllables and morphemes with LSTMs. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(2).

Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, and Milan Straka. 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 – 26, Hong Kong, China.

Stephan Oepen and Dan Flickinger. 2019. The ERG at MRP 2019: Radically compositional semantic dependencies. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 39 – 43, Hong Kong, China.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

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.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 2037–2048.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL '18, pages 1492–1502.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, pages 926–934.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Chuan Wang and Nianwen Xue. Getting the most out of AMR parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, EMNLP '17.

Yuxuan Wang, Wanxiang Che, Jiang Guo, and Ting Liu. 2018. A neural transition-based approach for semantic dependency graph parsing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '18, pages 5561–5568.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 80–94.