Dependency Parsing with the Structuralized Prompt Template

Anonymous ACL submission

Abstract

Dependency parsing is a crucial task in natural language processing that involves identifying syntactic dependencies to construct a structural tree of a sentence. Traditional models conduct dependency parsing by constructing embeddings and utilizing additional layers for prediction. We propose a novel method for performing dependency parsing using only a pre-trained encoder model with a text-to-text training approach. To facilitate this, we define the structured prompt template that effectively captures the structural information of the dependency tree. Our experimental results demonstrate that the proposed method achieves outstanding performance when comparing to traditional models, in spite of relying solely on an encoder model. Moreover, this method can be easily adapted to various encoder models that are suitable for different target languages or training environments, and it easily embody special features into the encoder models.

1 Introduction

Dependency parsing is a crucial task in natural language processing, involving the analysis of syntactic relationships between words in a sentence. Traditionally, dependency parsing has been performed in 2 steps: 1) creating word-level embeddings, 2) identifying the head word of each word and their dependency relation using the created embeddings. In the past, the first step of dependency parsing generally used simple pre-processed contextual vectors for initializing embeddings (Li et al., 2018; Strzyz et al., 2019; Vacareanu et al., 2020). These days, because pre-trained language models, such as BERT (Devlin et al., 2019), achieve a high ability to capture contextual characteristics, recent dependency parsing approaches tend to initialize word embeddings solely using the pre-trained language models and perform better than past methods (Amini et al., 2023). In the second step of dependency parsing, previous studies have shown that graph-based methods, such as biaffine (Dozat and Manning, 2017), yield good performance in identifying relations. Consequently, this approach was extended to learn the subtree information of the dependency tree (Yang and Tu, 2022). Since the structural characteristics of dependency trees increased training complexity and difficulties, some studies use the sequence tagging method for parsing (Li et al., 2018; Amini and Cotterell, 2022). These approaches add simple layers after embedding construction and label the words into structural information sequentially. In particular, the hexatagging method achieved state-of-the-art performance by generating structural information with a finite set of tags through decoding (Amini et al., 2023). In addition, Lin et al. (2022) have shown that encoder-decoder models perform well to generate relation unit text from the input text. This demonstrates that dependency parsing can be accomplished with pre-trained language models alone.

In this paper, we propose a novel method to perform dependency parsing solely on pre-trained encoder models that are constructed by prompt engineering using additional tokens as soft prompts. We think that prompt engineering can effectively convert the text-to-structure task in dependency parsing to the text-to-text task by pre-trained language models, just like the sequence tagging method. Hence, the output text sequence of the proposed method has to reflect the tree structure of dependency parsing well. For this, we develop several soft prompts so that our model can identify the structural information of the tree structure, and then do the Structuralized Prompt Template (SPT) for each processing unit of dependency parsing using the developed soft prompt. We believe that prompt learning with the structuralized prompt template enables effective and efficient dependency parsing only on the pre-trained language models. Eventually, by learning through the structuralized prompt template, the **S**tructuralized **P**rompt

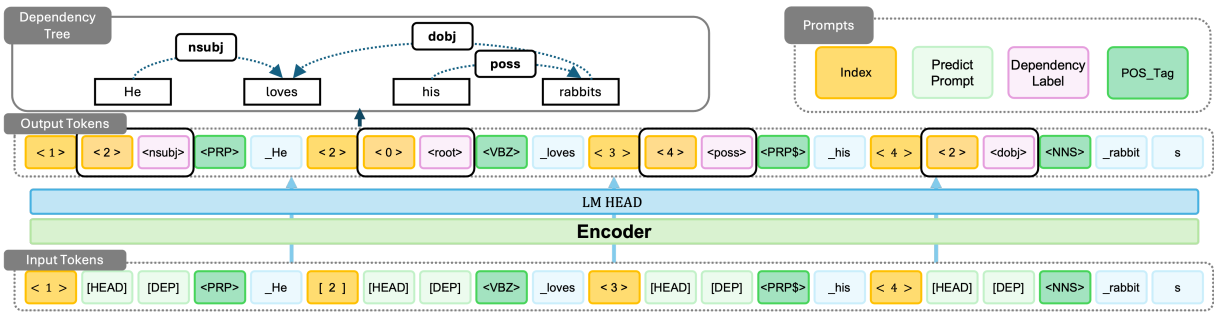

Figure 1: Overview of the Structuralized Prompt Template based Dependency Parsing (SPT-DP) method.

Template based **D**ependency **P**arsing (SPT-DP) method achieves the effective and efficient performance by reducing the gap between pre-training and fine-tuning because it is based on only the pre-trained language models for the text-to-text task.

As a result, the performance of the proposed method surpasses the ones of most existing methods: 96.95 (UAS) and 95.89 (LAS) on English Penn Treebank (PTB;Marcus et al. (1993)). On the 2.2 version of Universal Dependencies (UD2.2;Nivre et al. (2018)), it obtains the state-of-art performance in 2 languages out of 12 languages when using the cross-lingual Roberta (Liu et al., 2019) model. Furthermore, our method achieves a performance comparable to that of the SOTA model with a complicated and heavy architecture in the Korean Sejong dataset.

In summary, we enumerate our main contributions as follows:

1. We define the structuralized prompt template that well reflects structural information of the dependency tree with soft prompts.

2. We fine-tuned our model with the structuralized prompt template as prompt engineering, which can convert the text-to-structure task (dependency parsing) to the text-to-text task by the pre-trained language model.

3. The proposed method achieves high performance only using the pre-trained language model and the structuralized prompt template, which has several strong points of easy training, requiring small memory, and fast inference time.

2 Structuralized Prompt Template

As aforementioned, the structuralized prompt template is invented so that the pre-trained model can directly perform dependency parsing through prompt engineering using additional tokens as soft prompts. Since the dependency parsing inputs a sentence and outputs its dependency tree, we need to convert the raw input text to a formatted text sequence that well contains the structural information of the dependency tree. We define the structuralized prompt template to generate the formatted text sequence with useful information for the dependency parsing. The template utilizes newly defined special tokens, which are added in the look-up table and used as soft prompts.

2.1 Dependency Parsing with Text Representation

Basically, dependency parsing is a task to find a word with a dependency relation and determine its corresponding dependency label for each word. To express the structured dependency relationships through the prompt template for each word, several conditions should be satisfied; 1) Each template must be distinguished by its pattern through whole training, 2) it must be able to indicate its position, 3) it must be able to refer to the other word template with dependency relationship through the output, and 4) it must be able to express the dependency relation label through the output.

In the first condition, we ensure that the language model can distinctly recognize each template by following a consistent pattern rather than a specific token through all the training process. To satisfy the second condition, we use the <index> prompts that serve two roles: representing the template and indicating the template's position. Because the <index> prompts represent the relative position of templates in a formatted text sequence, each template can refer to other template with dependency relationship regardless of any input sequence using the <index> prompts. For the third condition, we add the [HEAD] prompt in the second position of the template, which has a dependency relation with

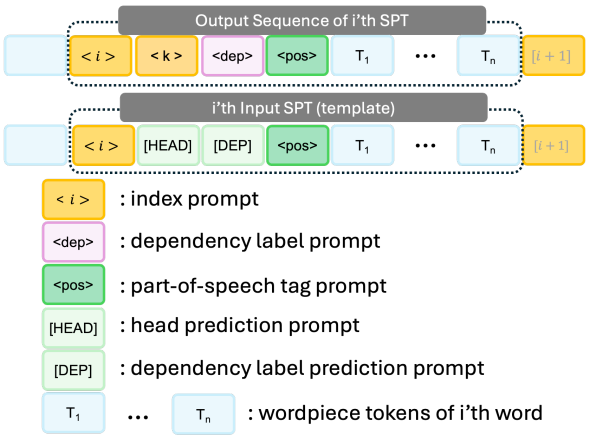

Figure 2: i'th input SPT and its output sequence.

another template: its output is the <index> prompt of the template with dependency relationship. The fourth condition is resolved by adding the dependency label prompt. The look-up table should be extended for prompt engineering by adding three kinds of prompt sets: the index numbers of templates (<index>), the dependency labels (<dep>), and the Part-of-Speech tags of words (<pos>). In addition, two more special tokens, [HEAD] and [DEP] are added as soft prompts and they take a role just like the masked token in the Masked Language Model (MLM) task. That is, we should infer which index is one of the prompt with correct dependency relation on [HEAD] and which dependency label is correct dependency label on [DEP]; in actual, correct index and label exist in output sequence at the training phase and this process is similar to the MLM task, as shown in Figure 2. The [HEAD] and [DEP] prompts are also added in the look-up table for prompt engineering.

2.2 Prediction Task using Soft Prompts: [HEAD] and [DEP]

As aforementioned, the [HEAD] and [DEP] prompts are used to infer two main prediction tasks for the head word and the dependency label. They are arranged in second and third positions of the structuralized prompt templates and they make a pattern in the input formatted text sequence. In our approach, the model is fine-tuned to predict head word and dependency labels by the [HEAD] and [DEP] prompts.

3 Prompt-based Training

Our training method is similar to the sequential labeling. However, the formatted text sequence is composed of repeated SPTs, and the <index> and dependency labels of the output sequence are replaced by the [HEAD] and [DEP] prompts for training; they are prompts for prediction tasks like the MLM task. Moreover, the other difference from sequence labeling is that our model trains in every word and prompts not only the [HEAD] and [DEP] prompts. In this training strategy, our model well learns the patterns of the repeated SPTs in the formatted text sequence.

The loss function for training is calculated by the following equations. In Equation 1, X(input) is the tokenized text that is a concatenated sequence of SPTs in which <index> and dependency label are replaced by [HEAD] and [DEP] prompts. In Equation 2, Y(label) is the tokenized text that is the formatted output sequence based on the repeated SPT pattern. Each X and Y contain a special token of the model. Since 100 index numbers (0 99), [HEAD], [DEP], POS tags, and dependency labels are added to the look-up table, the lengths of X and Y are always the same. This is a crucial condition in training models based on the pre-trained encoder. Since we train on all tokens in sequence, the training loss is described in Equation 3.

$$X_{input} = [x_1, x_2, ..., x_N] \quad (1)$$

$$Y_{label} = [y_1, y_2, ..., y_N] \quad (2)$$

$$\mathcal{L} = -\sum_{i=1}^{N} logP(y_i|X) \quad (3)$$

4 Experiments

4.1 Datasets and Pre-trained Language Models

For the PTB dataset, we preprocess the data with the v3.3.0 of Stanford Parser (de Marneffe and Manning, 2008) to convert it into CoNLL format and we organize this data by just following previous work (Mrini et al., 2020). For 2.2 version of UD datasets from 12 languages, we follow previous work (Amini et al., 2023) for data splitting and organizing. In UD2.2, the POS tag information is not used for the experiments by omitting POS prompts in the template. In the Sejong dataset, Korean words are composed of multiple morphemes, the POS tags of the first and last morphemes are used for this experiment.

XLNet-large for the Penn Treebank (PTB; Marcus et al. (1993)) and multilingual BERT, XLM-

	bg	ca	cs	de	en	es	fr	it	nl	no	ro	ru	Avg.
Dozat and Manning (2017)◇	90.30	**94.49**	92.65	**85.98**	91.13	93.78	**91.77**	94.72	91.04	94.21	87.24	94.53	91.82
Wang and Tu (2020)◇	91.30	93.60	92.09	82.00	90.75	92.62	89.32	93.66	91.21	91.74	86.40	92.61	90.61
Yang and Tu (2022)◇	91.10	94.46	92.57	85.87	91.32	**93.84**	91.69	**94.78**	91.65	94.28	87.48	94.45	91.96
Lin et al. (2022)*	**93.92**	93.75	92.97	84.84	**91.49**	92.37	90.73	94.59	**92.03**	**95.30**	88.76	**95.25**	**92.17**
Amini et al. (2023)◇	92.87	93.79	92.82	85.18	90.85	93.17	91.50	94.72	91.89	93.95	87.54	94.03	91.86
SPT-DP (multilingual BERT)	91.20	90.81	92.22	79.68	87.36	90.33	88.31	92.00	89.37	90.64	86.12	93.17	89.27
SPT-DP (XLNet-large)	-	-	-	-	90.58	-	-	-	-	-	-	-	-
SPT-DP (XLM-RoBERTa-large)	93.11	92.54	**94.14**	82.11	88.50	91.69	88.02	93.16	91.15	93.13	**88.87**	95.12	90.90

Table 1: 12 languages' LAS scores on the test sets in UD 2.2. ◇ use multilingual BERT for embedding and * uses T5-base model for sequence generation parsing.

	PTB		Sejong	
Model	UAS	LAS	UAS	LAS
Zhou and Zhao (2019)*	97.0	95.43	-	-
Mrini et al. (2020)*	97.42	96.26	-	-
Dozat and Manning (2017)	95.74	94.08	-	-
Wang and Tu (2020)	96.91	95.34	-	-
Yang and Tu (2022)	97.24	95.73	-	-
Lin et al. (2022)	96.64	95.82	-	-
Amini et al. (2023)	**97.4**	**96.4**	-	-
Park et al. (2019)	-	-	94.06	92.00
Lim and Kim (2021)	-	-	**94.76**	**92.79**
SPT-DP	96.95	95.88	94.52	92.36
SPT-DP (w/o <index>)	94.28	92.63	-	-
SPT-DP (w/o <pos>)	96.76	95.66	94.47	92.35

Table 2: Results on PTB and the Sejong Korean dataset. * use additional constituency parsing information so they are not comparable to other methods.

RoBERTa-large, XLNet-large for Universal Dependencies 2.2 (UD2.2; Nivre et al. (2018)), and the Korean version of Roberta (Liu et al., 2019) for the Korean Sejong dataset are used for our experiments.

4.2 Comparison Models

(Dozat and Manning, 2017) presented the biaffine model as a graph-based method. (Wang and Tu, 2020) introduced message passing for the second-order graph-based method. (Yang and Tu, 2022) invented a new method for projective parsing based on headed span. (Lin et al., 2022) proposed a parsing method with sequence generation. (Amini et al., 2023) utilized defined structural tags and sequential tag decoding for parsing.(Lim and Kim, 2021; Park et al., 2019) constructed a dependency parser using the Korean morpheme version of BERT.

4.3 Experimental Results

Table 2 shows the performance of each model on the PTB dataset. The proposed method achieves the comparable performances to the SOTA models, which have extra complicated modules or extra constituency parsing information; these performances position in second place although our model uses only pre-trained languag models. In addition, we do the ablation test about additional special tokens (prompts) to construct SPT: <index> and <pos>. As you can see in Table 2, the performance of experiment without <index> shows more performance decrease than one without <pos>. This demonstrates that the configuration of the template highly affects performance. As shown in Table 1, when the experiments are conducted using multilingual BERT in UD2.2, it shows lower performance than other models. After we exploit a slightly larger model, XLM-RoBERTa-large, our method significantly improves performance for the most part and the SOTA peformances are achieved in 2 languages. In addition, better performance was achieved when learning UD2.2-en data through XLNet-large, which was only pre-trained in English. This indicates that the type and size of the pre-trained model significantly impact parsing performance because we only use a pre-trained model for parsing. Our method achieves comparable performances to those of the SOTA model with a complicated and heavy architecture in the Korean Sejong dataset for Korean.

5 Conclusions

In this paper, we introduce the **SPT-DP**, structuralized prompt template based dependency parsing method. We perform text-to-text dependency parsing by prompt engineering using additional tokens using only pre-trained encoder models without any layer. Despite solely utilizing the pre-trained encoder model, the proposed model achieves comparable performances to existing models. Therefore, our method has several strong points in that it can be easily applied to various encoder models that are appropriate to the target language or training environment and easily embody special features into the encoder models.

Limitation

In our method, there is a limitation with sequence length. Although sentences with too many words are occurred in rare cases, additional prompts also increase linearly with the number of words, which can make it difficult to use for encoder models with a short maximum length. In addition, additional research is needed to perform semantic dependency parsing with a dynamic number of relationships.

Ethics Statement

We perform dependency parsing using a pre-trained model. The datasets may contain ethical issues or biased sentences, but the model does not influence them through dependency parsing.

References

Afra Amini and Ryan Cotterell. 2022. On parsing as tagging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8884–8900, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Afra Amini, Tianyu Liu, and Ryan Cotterell. 2023. Hexatagging: Projective dependency parsing as tagging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1453–1464, Toronto, Canada. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Joonho Lim and Hyunki Kim. 2021. Korean dependency parsing using token-level contextual representation in pre-trained language model. *Journal of KIISE Vol.48 No.1*, pages 27–34.

Boda Lin, Zijun Yao, Jiaxin Shi, Shulin Cao, Binghao Tang, Si Li, Yong Luo, Juanzi Li, and Lei Hou. 2022. Dependency parsing via sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7339–7353, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Khalil Mrini, Franck Dernoncourt, Quan Hung Tran, Trung Bui, Walter Chang, and Ndapa Nakashole. 2020. Rethinking self-attention: Towards interpretability in neural parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 731–742, Online. Association for Computational Linguistics.

Joakim Nivre et al. 2018. Universal dependencies 2.2. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Cheoneum Park, Changki Lee, Joonho Lim, and Hyunki Kim. 2019. Korean dependency parsing with bert. In *Proc. of the KIISE Korea Computer Congress*, pages 533–536.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. 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 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert Vacareanu, George Caique Gouveia Barbosa, Marco A. Valenzuela-Escárcega, and Mihai Surdeanu. 2020. Parsing as tagging. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5225–5231, Marseille, France. European Language Resources Association.

Xinyu Wang and Kewei Tu. 2020. Second-order neural dependency parsing with message passing and end-to-end training. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 93–99, Suzhou, China. Association for Computational Linguistics.

Songlin Yang and Kewei Tu. 2022. Headed-span-based projective dependency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2188–2200, Dublin, Ireland. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

A Implementation Details

For experiments for PTB, xlnet-large-cased[1] are used. For experiments for UD2.2, bert-multilingual-cased[2], xlm-roberta-large[3] and xlnet-large-cased are used. For the Korean Sejong dataset, we use roberta-large[4], which is a pre-trained model for the Korean language. We use NVIDIA RTX A6000 for experiments. The models are fine-tuned with 8 batch size, 1e-5 learning rate, and 10 training epochs. We train models with the linear scheduler and AdamW as a optimizer.

B Licenses

The PTB dataset is licensed under LDC User Agreement. The UD2.2 dataset is licensed under the Universal Dependencies License Agreement.

[1] https://huggingface.co/xlnet-large-cased

[2] https://huggingface.co/bert-base-multilingual-cased

[3] https://huggingface.co/FacebookAI/xlm-roberta-large

[4] https://huggingface.co/klue/roberta-large