Rethinking and Collaborative Learning Enhanced Cross-lingual Dependency Parsing

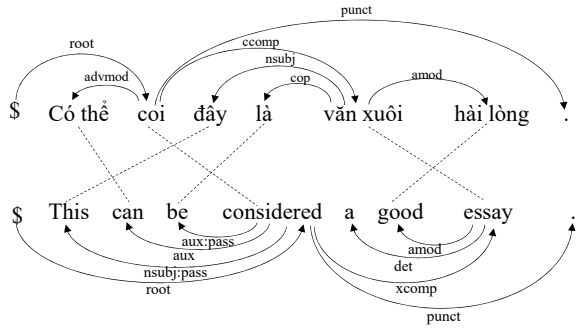
Anonymous ACL submission

Abstract

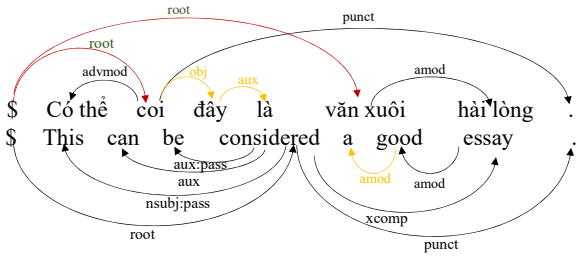
Large language models (LLMs) have shown strong syntax understanding capability in rich-source languages. However, their performances decline sharply when directly apply to low-resource languages. The key challenge is the data deviation and weak alignment across the source and target languages. To alleviate these issues, we propose a novel rethinking and collaborative learning approach for cross-lingual dependency parsing. On the one hand, we exploit a progressive thinking technique to guide LLMs to generate diverse and aligned synthetic data, thus making up for the data shift drawback. On the other hand, we introduce a collaborative learning strategy to further activate the alignment ability of both traditional cross-lingual models and LLMs by making full use of our synthetic data. Experiments on various benchmark datasets show that our proposed method outperform all strong baselines, leading to new state-of-the-art results on all language. Detailed comparison demonstrates that our synthetic data is extremely useful for enhancing the alignment between source and target languages. In-depth analysis reveals that both rethinking and collaborative learning can boost the cross-lingual parsing performance.

1 Introduction

Dependency parsing is a foundational natural language processing (NLP) task that aims to analyze the syntactic structure of an input sentence (Kondratyuk and Straka, 2019b). It first identifies the head word for each word in the input sentence, and then obtains the syntactic relationship between head and modifier words based on grammatical rules (Kulmizev et al., 2019). This process is essential for various NLP applications, such as machine translation (Ahmad et al., 2019), automatic summarization (Zhang et al., 2020), sentiment analysis (Droganova et al., 2021), and information retrieval (Osa et al., 2023).



a. Golden dependency trees



b. Generated dependency trees by LLMs

Figure 1: Examples of dependency trees where orange color and red color represent wrong relation labels and root nodes, respectively.

Recently, dependency parsing in rich-resource languages has made significant advancements. However, cross-lingual dependency parsing remains challenging due to data deviation and weak alignment between rich-resource source and low-resource target languages. The cross-lingual dependency parsing approaches are mainly categorized into two lines, i.e., data augmentation and feature transformation. The key idea of data augmentation is automatically generating target language dependency trees to alleviate the data shift problem of low-resource target languages (Feng et al., 2021; Shorten et al., 2021; Bayer et al., 2022; Wang et al., 2024; Sapkota et al., 2025). Recent studies highlight the promising potential of large language models (LLMs) in data augmentation (Wu et al., 2023; Yoo et al., 2021; Ko et al., 2023). Zhang et al. (2025) uses grammar and lexical information to help LLMs create subtrees, and then hybridize

them with existing source-domain subtrees to augment the diversity of training data. The goal of feature transformation is to learn beneficial feature representations from high-resource languages, enabling the model to adapt to low-resource target languages (Basu Roy Chowdhury et al., 2019; Xu et al., 2020). Liu et al. (2025a) design the dynamic syntactic feature filtering and injecting networks to enhance the language-invariant and language-specific feature presentations and achieve outstanding performances on cross-lingual dependency parsing.

Motivated by these works, we first analyze the relevance of source and target languages. As shown in Figure 1 (a), we can see that although the aligned positions between English words "good essay" and Vietnamese words "ăn xuôi (essay) hài lòng (good)" are changed, they still own the same relation label "amod". Then, we leverage LLMs to directly generate the dependency trees in both rich-resource English and low-resource Vietnamese. As described in Figure 1 (b), we find that the generated Vietnamese tree contains multiple erroneous relation labels and root nodes, indicating LLMs have a strong syntax understanding of English, while their ability obviously declines in Vietnamese. Therefore, it becomes the key challenge to alleviate data deviation and enhance aligned knowledge.

To address this issue, we propose a novel approach improving cross-lingual dependency parsing via LLM rethinking and collaborative learning. First, we exploit LLMs and traditional parsers to generate aligned multi-lingual dependency trees. Then, we design a rethinking chain to guide LLMs to self-optimize aligned multilingual dependency trees. Finally, we propose a multilingual cooperative learning algorithm to effectively utilize both our aligned dependency trees and existing multilingual dependency trees. Experiments on the benchmark datasets demonstrate that our proposed model significantly improves cross-lingual dependency parsing performance, leading to new state-of-the-art results on all languages. Comparison experiments validate the effectiveness of different marginal knowledge.

2 Related Work

Cross-lingual dependency parsing aims to learn useful information from rich-resource source languages to boost the parsing accuracy of low-resource target languages. Typical cross-lingual dependency parsing methods can be categorized into two lines, i.e., data augmentation and feature transfer.

Data augmentation aims to alleviate the data scarcity problem in low-resource languages by automatically generating pseudo corpora that approximate realistic data distributions. Traditional approaches include hierarchical augmentation, back-translation, and paraphrasing (Yu et al., 2019; Sennrich et al., 2016; Dai et al., 2025; Cegin et al., 2023, 2024; Wang et al., 2025). For instance, Sennrich et al. (2016) generate synthetic data through back-translation, while Yu et al. (2019) apply hierarchical attention to crop and concatenate salient sentence segments. More recent methods leverage large language models'(LLMs) strong generative and understanding abilities to further improve data quality and diversity (Lewis, 2020; et al., 2024b; Li et al., 2024; et al., 2024a, 2025; Anonymous, 2025; Liu et al., 2025b). LLM-based augmentation techniques include zero-shot prompting (Ubani et al., 2023), classifier-based filtering of unfaithful samples (Sahu et al., 2022), and LLM-guided few-shot data synthesis for downstream tasks like NER (Ye et al., 2024).

Feature transfer aims to transfer learned knowledge from a source domain to improve model performance in a target domain. The main approach includes parameter transfer, feature distribution transfer and knowledge distillation(Long et al., 2013; Yin et al., 2019; Lu et al., 2020; Yu et al., 2024; Fu et al., 2024; Gou et al., 2021). Chen et al. (2022) propose a parameter-efficient transfer learning method combining low-rank adaptation (LoRA) and prompt-tuning, achieving strong performance with minimal updates in low-resource NLP tasks. Wang et al. (2023) introduce Feature Correlation Matching (FCM), aligning cross-domain feature distributions to improve domain adaptation. Hou et al. (2024) develop Online Knowledge Distillation (OKD), combining contrastive learning and memory replay for robust performance in dynamic learning scenarios.

Despite the strong syntactic understanding capabilities of LLMs in resource-rich languages, the complexity of long sentences and data scarcity in low-resource languages remain key challenges for LLM-based dependency parsing in data augmentation and transfer learning. To tackle these issues, we propose a novel leverages the multilingual alignment ability of LLMs to transfer more effective features from multiple source languages. This

approach not only boosts parsing accuracy in the target language but also enhances the overall parsing capabilities of large models.

3 Our Approach

To enhance the cross-lingual dependency parsing performance, this work proposes a novel large language model (LLMs) rethinking and multi-lingual co-training approach. The basic idea is to activate the multilingual alignment ability of LLMs, thus transferring more effective features from multiple source languages to boost the target language parsing accuracy. As shown in Figure 2, our approach contains two stages, i.e., *Two-step progressive thinking for multi-lingual aligned dependency trees generation* and *Multi-lingual cooperative training*.

In the first stage, we exploit both traditional parsers and LLMs to generate accurate and aligned multi-lingual dependency trees. In the second stage, we propose a multi-lingual cooperative training method to enhance the alignment and parsing capability of all cross-lingual models by effectively leveraging our constructed aligned and existing multi-lingual dependency trees.

3.1 Two-step Progressive Thinking for Multi-lingual Aligned Dependency Trees Generation

Since the syntax understanding capability of existing LLMs for low-resource languages is limited, directly LLM-based corpus annotation leads to various incorrect dependency relation labels and root nodes, especially for complex sentences. To alleviate this drawback, we propose a multilingual aligned trees generation approach using a two-step progressive thinking. Concretely, we first obtain publicly released multi-lingual aligned unlabeled sentences. Then, each sentence in the source or target language is annotated by both traditional parsing models and LLMs. Intensively, the traditional parsing model relies on language-specific knowledge, enabling it to accurately handle basic syntactic structures, while LLMs benefit from extensive multilingual knowledge to more effectively capture complex semantic dependencies. Ultimately, we design a progressive thinking strategy to guide LLMs to filter out the best dependency tree based on outputs from traditional parsers and LLMs.

Pseudo corpus generation. To obtain rich and accurate aligned trees, we adopt two classic methods for dependency tree annotation, i.e., the traditional parser and LLM.

For *traditional parser based corpus generation*, we adopt the BiAffine parser as our baseline model and enhance its representational capacity by integrating the multilingual pre-trained language model XLM-RoBERTa. As the top block of Figure 2, we first utilize target language training data to update the initial parameters of the BiAffine parser. Then, the pre-trained BiAffine parser is leveraged to predict unlabeled data of the target language, thus obtaining the original traditional parser based corpus. Specifically, each input sentence is transformed into a sequence of dense vectors $\mathbf{x}_i$, where each vector is constructed by concatenating a word-level and a character-level representation. The word representation is formed by summing the averaged outputs from the last four layers of XLM-RoBERTa $\mathbf{rep}^{\text{XLM-R}}$ and a randomly initialized word embedding $\mathbf{emb}^{\text{word}}$. The character-level representation $\mathbf{word}^{\text{char}}$ is derived from a BiLSTM network that encodes the character sequence of each word. The final input vector is defined as Equation 1.

$$\boldsymbol{x}_i = (\mathbf{rep}^{\text{XLM-R}} + \mathbf{emb}^{\text{word}}) \oplus \mathbf{word}^{\text{char}} \quad (1)$$

Subsequently, a three-layer BiLSTM is employed as the encoder to generate contextualized representations. These representations are then passed through two separate multilayer perceptrons (MLPs) to obtain low-dimensional syntactic vectors corresponding to heads ($\mathbf{h}_i$) and dependents ($\mathbf{d}_i$). Finally, a BiAffine transformation is applied to compute each arc score between head word w_i and dependent word w_j as in Equation 2.

$$\text{arc}_{i \leftarrow j} = \begin{bmatrix} \mathbf{d}_i \\ 1 \end{bmatrix}^{\text{T}} \mathbf{U}_1 \mathbf{h}_j \quad (2)$$

Meanwhile, the parser uses another MLP and BiAffine to obtain the label score $label_{i \leftarrow j}$. Finally, for each position i, if the gold-standard head of word w_i is word w_j and its corresponding gold relation label is l, the parsing loss $\mathcal{L}_{\text{tra}}$ is computed as follows,

$$\begin{aligned} \mathcal{L}_{\text{tra}} = &- \log \frac{e^{\text{arc}_{i \leftarrow j}}}{\sum_{0 \le k \le n, k \ne i} e^{\text{arc}_{i \leftarrow k}}} \\ &- \log \frac{e^{label_{i \leftarrow j}}}{\sum_{label' \in L} e^{label'_{i \leftarrow j}}} \end{aligned} \quad (3)$$

where L is the set of labels.

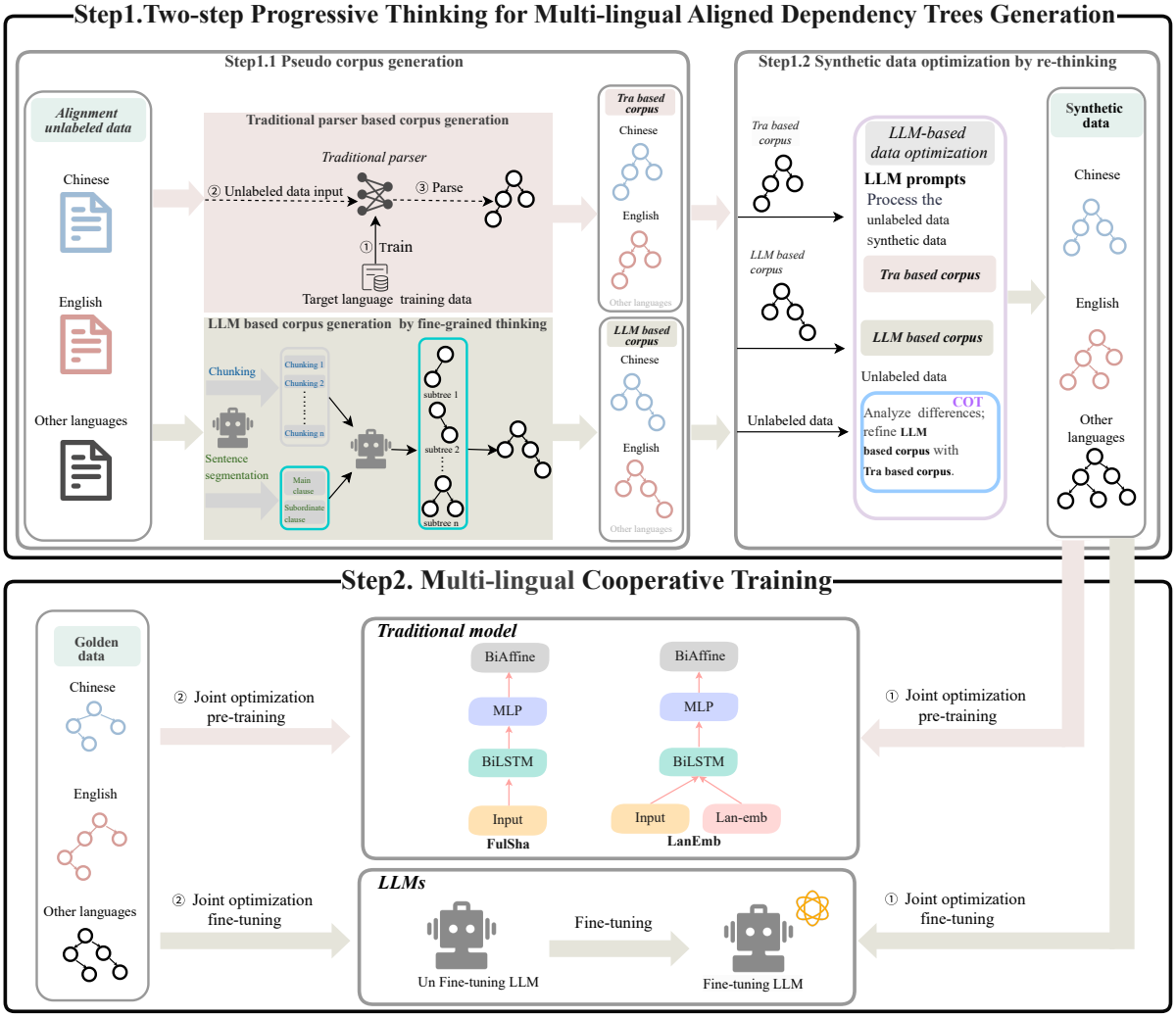


Figure 2: The overall architecture of our method on both traditional parsers and LLMs.

For *LLMs based corpus generation by fine-grained thinking*, we design a special two-stage chain-of-thought (CoT) prompt for guide LLMs to reason fine-grained grammatical knowledge, thus yielding an original LLMs-based corpus. In the first stage, LLMs need to segment sentences into meaningful linguistic units, such as chunks and clause boundaries. In the second stage, LLMs systematically process each linguistic unit, which first derives local subtrees based on chunks and clause boundaries, and then compositionally assembling these subtrees into a complete yet grammatically valid dependency tree.

Synthetic data optimization by rethinking. To enhance the synthetic corpus quality, we introduce another CoT prompt to boost LLM in-depth rethinking and reconstruct more reliable dependency trees. In the practical application, LLMs are more effective for handling complex or unofficial sentences, while traditional parsers have outstanding performance in sentences with standardized grammar. Therefore, we fist obtain two heterogeneous dependency trees for each target language sentence by traditional models and LLMs. Then, LLMs in-depth rethink to incorporate the advantages of traditional models and LLMs based on heterogeneous trees. Finally, a more accurate tree is reconstructed through the rethinking results.

3.2 Multilingual Collaborative Learning

To balance data deviation and align grammatical knowledge in cross-lingual dependency parsing models, we propose a multilingual collaborative learning algorithm as Algorithm 1. Specifically, at each training or fine-tuning step, we first alternately sample mini-batches from Chinese, English, and the target low-resource language. Then, model parameters are updated by minimizing the parsing loss $\mathcal{L}^{par}$ in a joint fashion. Finally, the training or fine-tuning process continues until convergence or

Algorithm 1 multilingual collaborative learning

Input: Chinese data Ch, English data En, low-resource target language data T
Parameters: total iterations k
Output: updated model

```
Initialize iter ← 0
while iter < k and not converged do
    Select mini-batch x alternately from Ch, En, or T
    if x ∈ Ch then
        Compute parsing loss L_tra or L_LLM
    else if x ∈ En then
        Compute parsing loss L_tra or L_LLM
    else
        Compute parsing loss L_tra or L_LLM
    end if
    Update model parameters by minimizing L_tra or L_LLM
    iter ← iter + 1
end while
```

an early stopping criterion is met.

In this work, we adopt two typical traditional cross-lingual dependency parsing models and two LLMs as our strong baseline models to verify the effectiveness of our collaborative learning.

For traditional cross-lingual dependency parsing models, we first sample LLM-optimized multilingual aligned dependency trees iteratively for model pre-training, and then the parameters of pre-trained models are updated by leveraging multilingual golden training data. Two typical traditional cross-lingual dependency parsing models include *Full shared model (FulSha)* and *Language embedding model (LanEmb)*. Concretely, the *FulSha* model is first utilized by Peng et al. (2017) for cross-domain dependency parsing, which treats all training data equally and shares all model parameters. Here, we also share all parameters of the BiAffine parser, no matter which language the data comes from. The *LanEmb* model is demonstrated to be extremely useful for cross-domain dependency parsing by Li et al. (2019b), which incorporates domain embeddings as additional input to indicate the domain type of each word. Here, we exploit language embeddings as auxiliary input to enhance the cross-lingual dependency parsing performance.

For LLM-based cross-lingual dependency parsing models, we adopt the widely-used fine-tuning strategy (low-rank adaptation, LoRA) to optimize parameters of LLMs efficiently with minimum resource consumption (Hu et al., 2022). First, each sentence from source or target languages are mapped into dense input vectors $\mathbf{x}$. Second, these input vectors are fed into multiple-layer Transformer blocks to obtain contextualized representations $\mathbf{y}$. The LoRA strategy introduces trainable low-rank matrices into each Transformer block and keeps the original pre-trained weights $\mathbf{W}$ frozen, which is defined as

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}, \tag{4}$$

where weight matrixes $\mathbf{W} \in \mathbb{R}^{d\times k}$, $\mathbf{B} \in \mathbb{R}^{d\times r}$, and $\mathbf{A} \in \mathbb{R}^{r\times k}$ with rank $r \ll \min(d, k)$. During the fine-tuning process, we compute the cross-entropy loss over the predicted heads and dependency labels, which is calculated as follows,

$$\mathcal{L}_{\text{LLM}} = -\sum_{i=1}^{H} h_i \log(\hat{h}_i) - \sum_{j=1}^{L} l_j \log(\hat{l}_j) \tag{5}$$

w here h_i and l_j are the gold-standard distributions of heads and labels, and $\hat{h}_i$ and $\hat{l}_j$ are the predicted probabilities of heads and labels generated by LLMs. We select two widely-used open-source LLMs to demonstrate the effect of collaborative learning, i.e., *Qwen2.5-7B-Instruct* and *Qwen2.5-14B-Instruct*. Specifically, the *Qwen2.5-7B-Instruct* model contains about 7 billion parameters and also owns remarkable Southeast Asian languages understanding capability, such as Chinese, Vietnamese, and Tamil. In contrast, the *Qwen2.5-14B-Instruct* model has more parameters and stronger linguistic comprehensive ability, specially for low-resource languages. In practical experiments, we further mix all synthetic and golden training data to align linguistic knowledge between source and target languages in these LLMs by simple LoRA fine-tuning.

Dataset	Train	Dev	Test	All
UD public datasets				
Chinese (GSDSimp)	3,997	500	500	4,997
English (EWT)	12,544	2,001	2,077	16,622
Vietnamese (VTB)	1,400	1,123	800	3,323
Tamil (TTB)	400	80	120	600
Telugu (MTG)	1051	131	146	1,328
Maltese (MUDT)	1,123	433	518	2,074
FLORES-200 Parallel Corpus Datasets				
Chinese	2,000	-	-	2,000
English	2,000	-	-	2,000
Vietnamese	2,000	-	-	2,000
Tamil	2,000	-	-	2,000
Telugu	2,000	-	-	2,000
Maltese	2,000	-	-	2,000
ALT Parallel Corpus Datasets				
Vietnamese	6,000	-	-	6,000

Table 1: Dataset statistics in sentence number.

Model	Synthetic Data		Vietnamese		Tamil		Telugu		Maltese		AVG	
	Source	Count	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
Results of previous works												
UDify(2019a)	-	-	66.00	74.11	68.29	78.34	83.91	92.23	75.56	83.07	73.44	81.94
ESR(2023)	-	-	60.80	70.21	66.40	74.12	80.10	81.60	74.20	82.34	70.38	77.07
Dynamic(2025a)	-	-	66.75	80.03	69.18	79.09	-	-	76.19	83.28	70.71	80.80
Results of traditional model												
FulSha	-	-	62.97	77.86	63.65	75.51	80.99	90.70	70.51	79.65	69.53	80.93
LanEmb	-	-	66.08	80.25	68.27	78.43	83.63	92.37	76.89	83.77	73.72	83.71
Our FulSha	GPT	2000	65.27	79.43	66.12	76.42	83.36	92.93	74.98	82.15	72.93	83.50
Our LanEmb	GPT	2000	67.14	81.38	68.23	78.69	81.28	91.82	75.77	83.07	73.11	83.23
Our FulSha	Qwen	2000	65.46	79.74	66.67	76.83	82.39	92.10	74.58	87.92	72.77	84.15
Our LanEmb	Qwen	2000	**67.02**	**80.99**	**67.67**	**78.48**	**83.08**	**92.93**	**77.18**	**83.85**	**73.74**	**84.56**
Fine-tuning Results												
Qwen2.5-7B-Instrcut												
LoRA	-	-	38.27	51.52	33.82	47.23	63.10	79.63	50.03	59.35	46.56	59.43
Our LoRA	Qwen	2000	61.86	74.19	53.58	64.37	76.42	86.96	67.38	74.19	64.81	74.93
Qwen2.5-14B-Instruct												
LoRA	-	-	41.98	55.79	36.62	49.97	65.25	83.03	51.63	60.98	48.87	62.44
Our LoRA	Qwen	2000	**63.74**	**77.99**	**59.54**	**71.77**	**78.50**	**89.74**	**72.67**	**78.97**	**68.61**	**79.62**

Table 2: Main results on the test dataset, where "Qwen" represents "Qwen2.5-7B-instruct", and "GPT" means "GPT-4o-mini".

4 Experiments

4.1 Experimental Setups

Datasets 1) *For training traditional parsers.* We collect gold-standard dependency trees for Chinese (zh), English (en), Vietnamese (vi), Tamil (ta), Telugu (te), and Maltese (mt) to train traditional parsing models, which are obtained from the Universal Dependencies (UD) v2.13 corpus [1]. 2) *For LLM-based syntactic data generation.* We utilize high-quality parallel unlabeled sentences from the ALT [2] and FLORES-200 [3] datasets to construct synthetic dependency trees for the six languages above. Detailed statistics are provided in Table 1. It is worth noting that ALT data is used only for the comparative analysis in Table 4, whereas all other experiments are conducted using data derived from FLORES-200 exclusively.

Evaluation We utilize Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS) as evaluation metrics (Li et al., 2019a). All models are trained for up to 500 iterations, and their performance is evaluated on the UD development dataset after each iteration. Training is stopped if no improvement is observed for 50 consecutive iterations. As shown in Table 2, we add our synthetic data to the initial UD training dataset to train traditional parsers and fine-tune LLMs, then test their performance on the UD test set.

4.2 Main results of experiments

Table 2 presents the main results of all baseline models and our approaches on both traditional models and LLMs across four low-resource languages.

For traditional models, we first find that their parsing accuracy has a significant improvement by adding our synthetic data, illustrating that our synthetic data can provide accurate syntax knowledge. Then, the synthetic data stemming from the Qwen series outperforms the one that comes from GPT. For large language models, our LoRA method outperforms the normal LoRA across all languages, demonstrating that our synthetic data can implicitly contain rich yet useful language-specific syntax knowledge to enhance the dependency parsing performance of LLMs.

In addition, we compare our approach with several previous works, i.e., *UDify* (Kondratyuk and Straka, 2019a), *ESR* (Effland and Collins, 2023), and *Dynamic* (Liu et al., 2025a). *UDify* utilize a shared multilingual encoder trained on universal treebanks to enable cross-lingual dependency parsing. *ESR* introduces regularization based on expected syntactic statistics to guide the parser in low-resource settings. *Dynamic* dynamically filters and injects syntactic features from source languages to improve cross-lingual transfer. Our models outperform most baseline models, verifying the robustness and generalizability of our multi-lingual

[1] https://universaldependencies.org/

[2] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/

[3] https://github.com/facebookresearch/flores/

Priori knowledge source	Vietnamese	
	LAS	**UAS**
None	16.51	32.69
Chunks data	23.35	42.51
Clause data	24.42	44.06
Traditional parser results	38.33	59.66
Chunking and Clause data	40.26	60.12
Our method	**58.51**	**75.31**

Table 3: Effectiveness of prior knowledge on Vietnamese parsing.

Synthetic data			Vietnamese	
Lang	Source	Count	LAS	UAS
zh	FLORES-200	2000		
en	FLORES-200	2000	57.66	74.38
vi	FLORES-200	2000		
vi	ALT	6000	60.06	76.67

Table 4: Scores are summarized for every group of synthetic data.

collaborative training approach. These results highlight the potential of combining synthetic data generation with cross-lingual transfer techniques to improve parsing performance for low-resource languages.

4.3 Ablation Study

Table 3 presents the ablation study results on Vietnamese development data. First, the addition of syntactic boundaries information (chunk and clause) can effectively improve parsing performance. Then, combining chunking and clause data further boosts the parsing accuracy, demonstrating the complementary nature of these two types of structural guidance. In addition, using the outputs of a traditional parser as soft supervision significantly enhances the syntax structure and semantic understanding of LLMs. Finally, our proposed method integrates all the above sources of prior knowledge, achieving the highest performance. This confirms the effectiveness of combining multiple types of linguistic priors in guiding LLMs' dependency parsing, especially in low-resource settings.

4.4 Multi-lingual Cooperative Training Study

Table 4 investigates whether using aligned high-resource languages can effectively assist low-resource languages in dependency parsing under the same amount of training data. First, we select 2000 aligned sentences from the FLORES-200 dataset in Chinese (zh), English (en), and Vietnamese(vi), respectively, and additionally sample 6000 unlabeled Vietnamese sentences from the ALT corpus. Second, we generate corresponding dependency parsing trees using our proposed method for both configurations. Then, these parsing trees are used to train traditional parsers, and the performance is evaluated on the Vietnamese test set from the UD dataset. Finally, although using only 6000 multilingual sentences does not outperform the 6000 monolingual Vietnamese setup, the performance gap remains small. This demonstrates that the existence of common syntactic knowledge across languages can complement the lack of syntactic information in low-resource languages, thus improving their parsing performance.

Traditional model Reference	LAS	UAS
None	40.26	60.12
English Parser	46.34	69.98
Vietnamese Parser	**58.51**	**75.31**

Table 5: Impact of different pre-trained parsers on Vietnamese dependency parsing. Better pseudo-data improves LLMs generation quality.

4.5 Comparative Study

Table 5 verifies the impact of using different reference syntax trees generated by various traditional parsers in our method. First, we employ the high-resource language (English) training data to train a traditional parser, then utilize the trained parser to obtain the target low-resource language (Vietnamese) parsing syntax tree. Next, we use these trees as reference syntax trees to enhance LLMs' parsing capability. As the results show, although the English parser does not outperform the Vietnamese-specific parser, it still provides an outstanding improvement over using no reference syntax tree at all. This suggests that when standard training data is unavailable for a low-resource target language, leveraging parsers trained on high-resource languages can be a viable strategy, which proves that a cross-lingual parser can leverage the common syntax knowledge from the source languages to parse the target languages accurately.

4.6 Error Analysis

To assess the quality of the synthetic data, we conducted a manual evaluation comparing dependency trees generated directly by large language models (LLMs) with those produced by our pro-

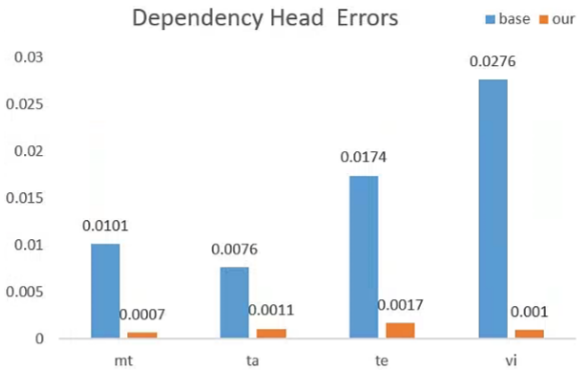


Figure 3: Proportion of head node errors in the final result.

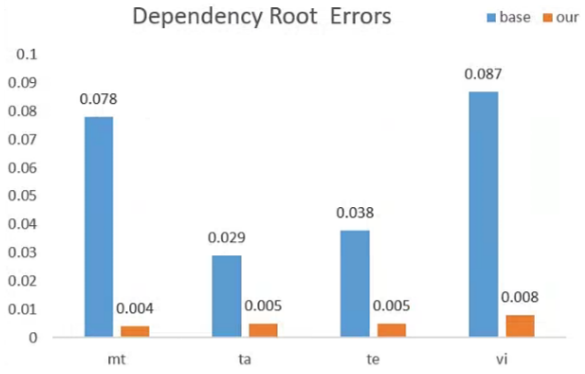


Figure 4: Proportion of root errors in the final result.

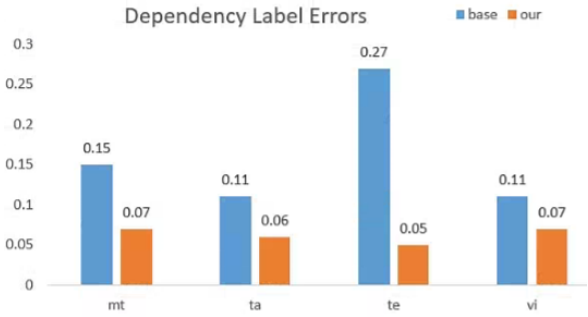


Figure 5: Proportion of label errors in the final result.

posed method. Concretely, the evaluation focuses on three main types of errors, i.e., dependency head errors, dependency label errors, and dependency root errors. First, Figure 3 reports statistics on invalid head word distribution in the dependency trees, including cases where the head word index exceeds the sentence length or introduces cyclic structures (violation of the single concatenation rule for syntactic trees). These structural errors compromise the integrity of the syntactic tree. Figure 4 analyzes root-related errors, including cases where a tree contains multiple roots or no root at all, which are critical violations in dependency parsing that signal incomplete or invalid syntactic structures. Figure 5 focuses on label-related issues, identifying non-standard or unknown dependency labels such as mark, sortof, case, multi-nsubj, and others. These labels are not part of the standard Universal Dependencies tagset, indicating semantic confusion or label misuse during generation.

Overall, the results show that our method significantly reduces all types of errors compared to the direct LLM-based generation method, with error rates dropping by more than half on average. Notably, root and head errors see the most dramatic reduction. This highlights the effectiveness of incorporating prior syntactic knowledge and structure-aware constraints into the LLMs syntax parsing process. Our approach enables the LLMs to better identify fine-grained sentence components and construct more accurate dependency trees.

5 Conclusion

We propose a novel rethinking and collaborative learning enhanced cross-lingual dependency parsing approach to alleviate the data deviation and weak alignment problem. Benchmark experiments demonstrate that our approach consistently improves cross-lingual dependency parsing performance on both traditional models and LLMs, leading to state-of-the-art results. An in-depth comparison shows that traditional parser-based pseudo samples are more effective on standard sentences, while LLM-based ones are better on unofficial ones since LLMs own stronger generation and reasoning capabilities. Furthermore, manual evaluations confirm that LLM re-thinking is extremely useful for combining the strengths of traditional parser-based and LLM-based pseudo data, thus yielding more accurate and higher-quality synthetic data. Detailed analysis indicates that our collaborative learning is extremely useful to align the linguistic community across source and target languages.

Limitations

First, our experiments cover only a limited number of large language models. In addition, the external knowledge for injecting large language models is not comprehensive enough. We will supplement the current shortcomings with in-depth research in our further work.

References

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Kai-Wei Chang, and Nanyun Peng. 2019. Cross-lingual

dependency parsing with unlabeled auxiliary languages. In *Proceedings of CoNLL*, pages 372–382, Hong Kong, China. Association for Computational Linguistics.

Anonymous. 2025. *Causal Reasoning in Natural Language Processing with Large Language Models*. Ph.D. thesis, Unknown University.

Somnath Basu Roy Chowdhury, Annervaz M, and Ambedkar Dukkipati. 2019. Instance-based inductive deep transfer learning by cross-dataset querying with locality sensitive hashing. In *Proceedings of DeepLo*, pages 183–191.

Markus Bayer, Marc-Antoine Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39.

Jakub Cegin, Branislav Pecher, Jakub Simko, Ivan Srba, Maria Bielikova, and Peter Brusilovsky. 2024. Effects of diversity incentives on sample diversity and downstream model performance in llm-based text augmentation. *arXiv preprint arXiv:2401.06643*.

Jakub Cegin, Jakub Simko, and Peter Brusilovsky. 2023. Chatgpt to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. *arXiv preprint arXiv:2305.12947*.

Tianlong Chen, Yu Cheng, Zhangyang Wang, and Yang Liu. 2022. Parameter-efficient transfer learning for nlp. *ICML*, pages 1120–1135.

Hang Dai, Zhiyuan Liu, Weixin Liao, Xuan Huang, Yang Cao, Zhiwei Wu, Lin Zhao, Sheng Xu, Fan Zeng, Wei Liu, et al. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*. In press.

Kira Droganova, Daniel Zeman, and Tanja Samardžić. 2021. Udpipe future: Towards more accurate and lightweight dependency parsing. In *Proceedings of UDW*, pages 46–54, Barcelona, Spain. Association for Computational Linguistics.

Thomas Effland and Michael Collins. 2023. Improving low-resource cross-lingual parsing with expected statistic regularization. *TACL*, pages 122–138.

Anonymous et al. 2025. Context-alignment: Activating and enhancing llms capabilities in time series. In *ICLR*.

Peking University et al. 2024a. Chain-of-discussion: A multi-model framework for complex evidence-based question answering. In *arXiv*.

Tsinghua University et al. 2024b. Longalign: A recipe for long context alignment of large language models. In *arXiv*.

S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. 2021. A survey of data augmentation approaches for nlp. In *Findings of ACL-IJCNLP*, pages 968–988.

Shiyi Fu, Shengyu Tao, Hongtao Fan, Kun He, Xutao Liu, Yulin Tao, Junxiong Zuo, Xuan Zhang, Yu Wang, and Yaojie Sun. 2024. Data-driven capacity estimation for lithium-ion batteries with feature matching based transfer learning method. *Applied Energy*, 353:121991.

Jiuxiang Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *IJCV*, 129(6):1789–1819.

Yuenan Hou, Yue Ma, Ziwei Liu, and Chen Change Loy. 2024. Distill on the go: Online knowledge distillation in self-supervised learning. *NeurIPS*, 37.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Hyun-Koo Ko, Hyeon Jeon, Gwangyeon Park, Dong-Hyun Kim, Nam Wook Kim, Jinwook Kim, and Joonhwan Seo. 2023. Natural language dataset generation framework for visualizations powered by large language models. *arXiv preprint arXiv:2309.10245*.

Dan Kondratyuk and Milan Straka. 2019a. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of EMNLP-IJCNLP*, pages 2779–2795.

Daniel Kondratyuk and Milan Straka. 2019b. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of EMNLP-IJCNLP*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing – a tale of two parsers revisited. In *Proceedings of EMNLP-IJCNLP*, pages 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Patrick et al. Lewis. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*.

Ying Li, Zhenghua Li, Min Zhang, Rui Wang, Sheng Li, and Luo Si. 2019a. Self-attentive biaffine dependency parsing. In *Proceedings of IJCAI*, pages 5067–5073.

Ying Li, Jianjian Liu, Zhengtao Yu, Shengxiang Gao, Yuxin Huang, and Cunli Mao. 2024. Representation alignment and adversarial networks for cross-lingual dependency parsing. In *Findings of EMNLP*, pages 7687–7697. Association for Computational Linguistics.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019b. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of ACL*, pages 2386–2395.

Jianjian Liu, Zhengtao Yu, Ying Li, Yuxin Huang, and Shengxiang Gao. 2025a. Dynamic syntactic feature filtering and injecting networks for cross-lingual dependency parsing. In *Proceedings of the AAAI*, volume 39, pages 24614–24622.

Jianjian Liu, Zhengtao Yu, Ying Li, Yuxin Huang, and Shengxiang Gao. 2025b. Dynamic syntactic feature filtering and injecting networks for cross-lingual dependency parsing. In *Proceedings of AAAI*, pages 24614–24622. Proceedings of AAAI.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. 2013. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE*, pages 2200–2207.

Yan Lu, Yue Wu, Bin Liu, Tianzhu Zhang, Baopu Li, Qi Chu, and Nenghai Yu. 2020. Cross-modality person re-identification with shared-specific feature transfer. In *Proceedings of the IEEE/CVF*, pages 13379–13389.

Yusuke Osa, Daisuke Kawahara, and Sadao Kurohashi. 2023. Multilingual parsing from raw text with a lightweight joint part-of-speech tagger and dependency parser. In *Proceedings of EACL*, pages 1247–1261, Dubrovnik, Croatia. Association for Computational Linguistics.

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of ACL*, pages 2037–2048.

Gaurav Sahu, Pau Rodriguez, Issam H. Laradji, Pooya Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *arXiv preprint arXiv:2204.01959*.

Ramesh Sapkota, Syed Raza, Mohamed Shoman, Anish Paudel, and Manoj Karkee. 2025. Image, text, and speech data augmentation using multimodal llms for deep learning: A survey. *arXiv preprint arXiv:2501.18648*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of ACL*.

Connor Shorten, Taghi M. Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of Big Data*, 8(1):101.

Samuel Ubani, Serhan O. Polat, and Rodney Nielsen. 2023. Zeroshot dataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Kai Wang, Jun Zhu, Ming Ren, Zhen Liu, Shuo Li, Zhi Zhang, Chao Zhang, Xia Wu, Qiang Zhan, Qiang Liu, et al. 2024. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.

Ximei Wang, Jingjing Liu, Dapeng Li, and Mingsheng Xue. 2023. Adversarial domain adaptation with feature correlation matching. *IEEE*, 45(3):2987–3001.

Zhen Wang, Jiahao Zhang, Xinyu Zhang, Kai Liu, Peiyi Wang, and Yang Zhou. 2025. Diversity oriented data augmentation with large language models. *arXiv preprint arXiv:2502.11671*.

Shih-Lun Wu, Xuankai Chang, Gordon Wichern, Jeeweon Jung, François Germain, Jonathan Le Roux, and Shinji Watanabe. 2023. Improving audio captioning models with fine-grained audio features, text embedding supervision, and llm mix-up augmentation. *arXiv preprint arXiv:2309.17352*. Version: 2023c.

Chengdong Xu, Xiaorui Zhao, Xiaoming Jin, and Xiansheng Wei. 2020. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the CVPR*, pages 11724–11733, Virtual.

Jiahua Ye, Ning Xu, Yixuan Wang, Jiacheng Zhou, Qian Zhang, Tao Gui, and Xuanjing Huang. 2024. LLM-DA: Data augmentation via large language models for few-shot named entity recognition. *arXiv preprint arXiv:2402.14568*.

Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. 2019. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF*, pages 5704–5713.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. In *Findings of EMNLP*, pages 2225–2239.

Sheng Yu, Jie Yang, Dong Liu, Ru Li, Yang Zhang, and Shikun Zhao. 2019. Hierarchical data augmentation and the application in text classification. *IEEE Access*, 7:185476–185485.

Yue Yu, Hamid Reza Karimi, Peiming Shi, Rongrong Peng, and Shuai Zhao. 2024. A new multi-source information domain adaption network based on domain attributes and features transfer for cross-domain fault diagnosis. *Mechanical Systems and Signal Processing*, 211:111194.

Yuhao Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural crf constituency parsing. In *Proceedings of IJCAI*, pages 4046–4053. International Joint Conferences on Artificial Intelligence Organization.

Ziyan Zhang, Yang Hou, Chen Gong, and Zhenghua Li. 2025. Data augmentation for cross-domain parsing via lightweight LLM generation and tree hybridization. In *Proceedings of ICCL*, pages 11235–11247, Abu Dhabi, UAE. Association for Computational Linguistics.

System Prompt
[Role] dependency parsing expert.
[Task] Chunk sentences into syntactic phrases (5 words).
[Instructions]
• Single-line output, no explanations.
[Example]
Input: ABCDEF
Output: (NP A) (VP B) (NP CD) (PP E) (NP F)

User Prompt
[Input] Andrea Maisi ...

[Output] (NP Andrea Maisi) ...

Table 6: Chunking Prompt Structure

System Prompt
[Role] dependency parsing expert.
[Task] Identify main and subordinate clauses in sentences.
[Instructions]
• Output [Main Clause] and [Subordinate Clause] separately.
• If no subordinate clause, only return [Main Clause].
• No explanation, follow strict format.
[Example]
Input: AB Output: [Main Clause]: A [Subordinate Clause]: B

User Prompt
[Input] Andrea Maisi ...

[Output] [Main Clause]: Andrea Maisi... [Subordinate Clause]: ...

Table 7: Clause Prompt Structure

A Appendix

A.1 Prompt Template in Aligned Trees Generation

In this section, we provide the prompt templates used in the *Aligned Tree Generation*. Table 6 presents the prompt used for chunking, where the input sentence is segmented into syntactic phrases with a maximum length of five words, each enclosed in labeled brackets such as (NP ...), (VP ...), etc. Table 7 shows the prompt designed for clause identification, which extracts the main and subordinate clauses from the sentence in a structured format. Finally, Table 8 illustrates the prompt used to drive the first-round dependency tree generation using CoT reasoning. This prompt takes the original sentence along with its chunked phrases and clause structures as reference information and guides the model to generate a simplified CoNLL-U formatted dependency tree. All prompts emphasize format consistency, strict input-output alignment, and eliminate unnecessary explanations to ensure the model follows precise syntactic logic during parsing.

A.2 Prompt Template in Synthetic Data Optimization Based on LLMs

In this section, we present the prompt template used in the *Synthetic Data Optimization* phase driven by LLMs. As shown in Table 10, the prompt guides the model to perform structured comparison between the two trees, identify issues such as incorrect head assignments or invalid dependencies, and merge the strengths of both results to produce a high-quality dependency tree. The final output must follow the simplified CoNLL-U format with strict rules, including the presence of a single root, use of standard dependency labels, and structural validity of the tree.

A.3 Prompt Template in Fine-tuning

In this section, we present the fine-tuning template as table 9. Specifically, we provide segmented input sentences along with their corresponding annotated outputs. A similar template is used in the final evaluation phase, where the outputs are generated by the large language model.

System Prompt
[Role] dependency parsing expert.
[Task] Parse the input sentence and output in simplified CoNLL-U format.
[Reference Info]
• Raw sentence, Chunking, Clause structure, Model info
[Reasoning]
1. Identify main verb as root.
2. Use chunking to verify phrase boundaries.
3. Build dependency tree.
4. Refine with external parser.
5. Output in final format.
[Output Rules]
• Use tab-separated fields.
• Single root per sentence.
• All dependencies must be labeled and acyclic.
[Format]
1. ID 2. Word 3. UPOS 4. Head 5. Relation
Only output the final CoNLL-U. No explanation.
[Example]
Input: AB Output: [Main Clause]: A [Sub Clause]: B

User Prompt
[Input]
[Raw] Andrea Maisi ...
[Reference Info] ...

[Output]
.....

Table 8: Parsing Prompt Structure

System Prompt
[Role] dependency parsing expert.
[Task] Parse the segmented sentence into CoNLL-U syntactic universal format.

[Example]
Input: Mòể̂at \t hai \t cÕánh tay \t ,
\t mòẹ̃Ẽt \ mòể̂ắt \t .

[Output]
1\tM òể̂at\tVERB \t 0\troot
2\thai \tNUM \t 3\tnummod
...
7\t .\tPUNCT \t 1\tpunct

Table 9: Example of fine-tuning

System Prompt
[Role] dependency parsing expert.
[Task] Analyze and compare two dependency trees (from LLM and traditional parser), then output a final optimized tree in simplified CoNLL-U format.
[Reference Info]
• Raw sentence, LLM parse, Traditional parser parse, Parsing rules
[Reasoning Steps]
1. Preliminary Check: Examine both trees for correctness in label usage, head selection, tree integrity (non-cyclicity, single root).
2. Comparative Reasoning: Retain LLM outputs when valid; use traditional parser results when more linguistically accurate.
3. Final Synthesis: Merge the best parts of both trees to produce a consistent, valid, high-quality structure.
[Output Rules]
• Use tab-separated fields.
• Sentence must have exactly one root.
• Dependency labels must follow standard syntax roles.
• Output must be acyclic and structurally valid.
[Format]
1. ID 2. Word 3. UPOS 4. Head 5. Relation
Only output the final CoNLL-U. No reasoning, no explanation.
[Example]
Input: [LLM Tree]: ... [Traditional Tree]: ...

User Prompt
[Input]
[Raw] Andrea Maisi ...
[LLM Tree] ...
[Traditional Tree] ...

[Output]
.....

Table 10: Parsing Prompt Structure for Secondary Contemplation