Data Augmentation for Cross-domain Parsing via Lightweight LLM Generation and Tree Hybridization

Ziyan Zhang[*], **Yang Hou**[*], **Chen Gong**[†], **Zhenghua Li**
School of Computer Science and Technology, Soochow University
{zyzhang0509,yhou1}@stu.suda.edu.cn
{gongchen18,zhli13}@suda.edu.cn

Abstract

Cross-domain constituency parsing remains a challenging task due to the lack of high-quality out-of-domain data. In this paper, we propose a data augmentation method via lightweight large language model (LLM) generation and tree hybridization. We utilize LLM to generate phrase structures (subtrees) for the target domain by incorporating grammar rules and lexical head information into the prompt. To better leverage LLM-generated target-domain subtrees, we hybridize them with existing source-domain subtrees to efficiently produce a large number of structurally diverse instances. Experimental results demonstrate that our method achieves significant improvements on five target domains with a lightweight LLM generation cost.

1 Introduction

As a fundamental task in natural language processing (NLP), constituency parsing aims to analyze the syntactic structure of a sentence by representing the compositional relationships of its constituents in a hierarchical tree, which has been proven to be beneficial for various downstream tasks (Wang et al., 2018; Xu and Durrett, 2019). Though constituency parsing has made significant advancements in in-domain scenarios over the years (Kitaev and Klein, 2018; Zhang et al., 2020), cross-domain parsing remains challenging due to the scarcity of high-quality labeled data specific to each target domain.

Previous works usually enhance cross-domain parsing via automatic data augmentation approaches, which are more cost-effective than manually annotating target-domain data (Feng et al., 2021; Zhang et al., 2022). Recently, the large language model (LLM) has achieved great success with its powerful generative capabilities in many NLP tasks (Wei et al., 2022; Liu et al., 2023). Li et al. (2023) utilize LLM to generate target-domain sentences and use a small parser to obtain tree structures. Their method leverages LLM to generate a large number of complete sentences and requires multiple iterations of self-training, leading to relatively high computational costs.

In this paper, we propose a simple and effective data augmentation method for cross-domain constituency parsing. As tree structures are rarely presented in the pretraining, LLM parsing is hindered by flawed trees and faces challenges with longer constituents (Bai et al., 2023). To address this issue, we employ LLM to generate phrases instead of complete sentences or trees. Specifically, we leverage grammar rules as guidelines to produce high-quality phrases with corresponding subtree structures for the target domain.

To better utilize the subtrees generated by LLM for cross-domain parsing, we introduce a tree hybridization method. The core idea is that lexicalized trees with the same constituent label and lexical head are interchangeable. In this work, *hybridization* conveys dual significance: 1) Technologically, it refers to that we allow the newly hybrid-generated subtrees to be used in subsequent hybridizations, which is significantly differs from previous works and increases structural diversity and complexity. 2) Intuitively, it involves integrating phrases from the target domain into sentences from the source domain, effectively merging the two domains and minimizing their disparities.

To summarize, our contributions are three-fold:

- We propose a lightweight method to leverage LLM for generating high-quality phrases with corresponding subtree structures, which is responsible for introducing words for the target domain.
- We design a tree hybridization method to efficiently and continuously produce a large number of diverse instances for data augmentation, effectively leveraging subtrees generated by LLM, which mainly serves to create new structures.

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 11235–11247
January 19–24, 2025.

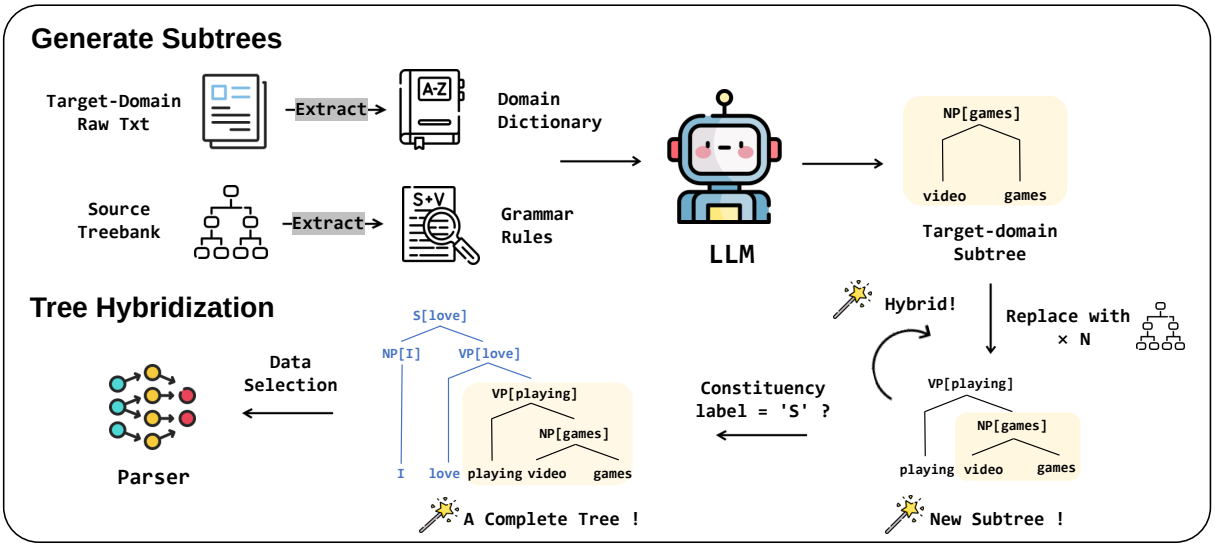


Figure 1: Overall workflow of our proposed method.

- Experimental results demonstrate the efficacy of our approach in improving cross-domain parsing performance while maintaining a low cost.

Our code is available at https://github.com/zzy0509/LLM-Tree-Hybridization.

2 Related work

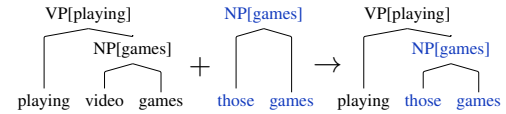
Cross-domain Constituency Parsing Constituency parsing, a fundamental task in NLP, has shown significant progress in recent years (Stern et al., 2017; Kitaev and Klein, 2018; Zhang et al., 2020; Xin et al., 2021; Yang and Tu, 2022). While achieving over 95% F-scores in the newswire domain using the Penn Treebank (PTB) (Marcus et al., 1993), the performance in the open domain remains struggling due to the lack of high-quality annotated domain-specific data.

Fried et al. (2019) trained parsers on in-domain corpora and evaluated them on out-domain corpora, presenting valuable insights into the generalization of neural parsers. Li et al. (2023) first applied self-training method to cross-domain constituency parsing and proposed enhancing it with LLMs by iteratively generating domain-specific raw corpora. Their approach differs from ours as they use LLMs to generate sentences for the target domain and derive tree structures with a self-trained parser. In contrast, we utilize LLMs to generate phrases with accurate subtree structures and without requiring an additional parser. Notably, our LLM generation process is more lightweight, generating only a few phrases instead of numerous sentences.

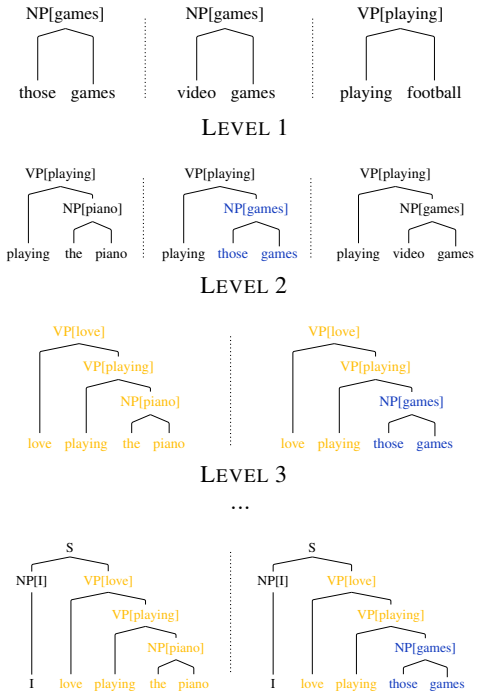
Data Augmentation for Constituency Parsing Due to the high cost of annotation, especially for tree structures, data augmentation has gained significant popularity in constituency parsing. Some studies (Shi et al., 2021b; Xu et al., 2021; Yang et al., 2022) have explored enhancing parsers through the use of partially annotated structures, such as entity spans or markups. Another augmentation method involves creating new substructures. Shi et al. (2020, 2021a) propose a method that generates trees by substituting subtrees with the same constituent label. Similarly, Zhang et al. (2022) use subtree replacement to create new sentences for downstream tasks, achieving significant performance on various text classification benchmarks.

However, previous methods may lead to semantic errors in the generated data. Our approach goes further by utilizing subtrees annotated with lexical heads from lexicalized trees to improve semantic accuracy. Additionally, we further consider the diversity of new data. Thus, we introduce tree hybridization to merge different subtrees.

LLM Parsing Recently, LLMs have achieved remarkable success in many NLP tasks (Brown et al., 2020; Wei et al., 2022; Liu et al., 2023). However, because tree structure data is rarely included in pretraining, LLMs may not be effective few-shot learners for constituency parsing. Bai et al. (2023) revealed that LLMs suffer from hallucinations and have limited ability to learn extremely long constituents. To address these challenges, we simplify the prompt requirements, enabling LLMs to generate phrases with accurate subtree structures.



(a) Generate a new subtree via one-time replacement.



The last LEVEL: the complete tree.

(b) Hybridization from the lower level to the higher level.

Figure 2: The process of tree hybridization. We use yellow color and blue color to represent the subtree structures from different trees, respectively.

3 Our Approach

To enhance cross-domain constituency parsing, in this work, we propose a data augmentation method. The basic idea is to first leverage the powerful generative capabilities of LLM to generate accurate and domain-specific constituency subtrees. Then, a tree hybridization method is proposed to efficiently produce a large number of diverse instances for further data augmentation, by making full use of both the LLM-generated target-domain subtrees and existing source-domain treebank. The whole workflow of our method is shown in Figure 1.

3.1 Tree Hybridization

To better understand, we first introduce our tree hybridization method, which is responsible for creating a large number of new structures with diversity. The main idea is that the lexicalized subtrees with the same non-terminal constituency label and lexical head can be replaced with each other to produce new subtrees, as is shown in Figure 2(a). As presented in Algorithm 1, our tree hybridization method can be divided into 3 detailed steps:

Algorithm 1: Tree Hybridization

input : Source Lexicalized Treebank S
output : Pseudo Treebank T

$T = \{\}$
for *iter in* $[1, 2, 3]$ **do**
  Construct subtree dictionary D *with different levels based on* S
  for *level in* $[1, 2, ...]$ **do**
    for s *in* D_{level} **do**
      Choose alternatives: $v \in D_{level-1}$
      Replace subtrees: $s' \leftarrow$ replace the subtree of s with v
      Update corpus: $D_{level} = D_{level} \cup s'$
    end
    $S = S \cup D_{level}$
  end
end
Select $T \in S$ **whose label is 'S'**

1) Constructing Subtree Dictionary: As shown in Figure 2(b), we collect subtrees from the source lexicalized treebank and level them based on the number of leaf nodes to construct a subtree dictionary. Subtrees containing more leaf nodes are assigned a higher level and the last level consists of the complete lexicalized trees. Specifically, the source lexicalized treebank containing two parts, the existing source-domain lexicalized treebank, which is built by applying head-finding rules to the source-domain constituency treebank, and the target-domain lexicalized subtrees generated by LLM as illustrated in Section 3.2. Since hybridization continuously generates new structures, the dictionary contains two types of subtrees: the subtrees from the source lexicalized treebank and the subtrees generated from hybridization.

2) Continuously Replacing Subtrees: During each replacement, we first determine to choose the subtree generated from hybridization with the probability p and the subtree generated from LLM with the probability $1 - p$. Then, we randomly select an alternative subtree that has the same constituency label and the same lexical head as the subtree being replaced and then generate a new subtree. The

dictionary is updated after each replacement, allowing the newly generated subtrees to be used continuously in subsequent replacements.

Moreover, hybridization starts from the lower level and progresses to the last level, which is a bottom-up process. We use the subtrees contained in the previous level, which have been updated, to replace the subtrees of the current level. Taking Figure 2(b) as an example, the subtree "playing those games" in LEVEL 2 is used to generate a higher subtree "love playing those games" in LEVEL 3.

3) Obtaining pseudo treebank: The hybridization process continues until the replacement operation is completed for the last level, generating a large number of complete trees, concluding one iteration of hybridization. To generate more complex structures, our hybridization comprises three iterations. At the end of the hybridization process, we select trees whose constituency label is 'S', forming a pseudo treebank for cross-domain constituency parsing data augmentation.

Compared with the previous compositional constituency-based data augmentation method (Shi et al., 2021a; Zhang et al., 2022), our method has the following two advantages. 1) We allow the newly produced subtrees to participate in subsequent replacements continuously, rather than just one-time replacement. This not only introduces greater diversity to the composed tree structures but also significantly increases the number of trees produced for data augmentation. 2) We use lexicalized tree instead of constituency tree, requiring the replaced subtrees same in constituency label and lexical head. This enhances the replacement constraints and significantly improves semantic correctness, which is proved in Appendix A.1.

3.2 Generating Target-Domain Subtrees via LLM

LLMs exhibit impressive generative capabilities which have been successfully leveraged to produce new and diverse augmented data for enhancing various tasks (Wan et al., 2023; Xu et al., 2023; Abaskohi et al., 2023). However, LLMs still have limitations in handling constituency parsing, suffering from the hallucination of generating invalid constituency trees, especially when the input sentences are domain-specific and long (Bai et al., 2023). To address the above issue, in this work, we propose a lightweight data augmentation method that simplifies the generation task by requiring the LLM to generate only target-domain phrases based on given grammar rules, rather than generating full raw sentences. In this way, we can directly obtain high-quality target-domain phrases along with their constituency subtrees from the LLM, eliminating additional parsing to determine the constituency structure of the LLM-generated augmented data. As illustrated in the Figure 1, the target domain subtrees are generated by the following three stages:

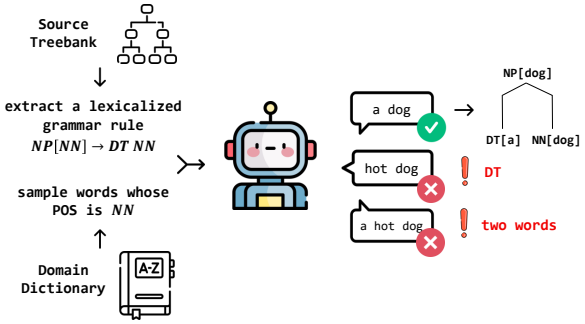


Figure 3: The process of LLM generating.

Stage 1: Extracting Domain Dictionary and Lexicalized Grammar Rules. At this stage, to mitigate the issues of hallucination and lacking flexibility in the LLM, we fully utilize existing large-scale target-domain raw texts and source-domain treebanks to extract a target-domain dictionary and lexicalized grammar rules, respectively. These resources serve as instructions to guide the LLM in generating high-quality target-domain phrases with the specified constituency subtree structures.

For the target-domain dictionary, we directly select high-frequency words from existing target-domain raw texts. We also obtain the corresponding part-of-speech (POS) tag for each word in the dictionary, which will be used in the second stage. For the lexicalized grammar rules, we extract rules from the source-doamin lexicalized treeban. Considering the differences in grammar rules across domains, especially for high subtrees (Yang et al., 2022), thus we only use subtrees whose heights are between three and eight. For instance, the subtree "video games" in Figure 1 with the height of 3 is considered as the rule "NP → NN NNS".

Stage 2: LLM Prompting with Target-domain Words and Lexicalized Grammar Rules. After extracting the target-domain dictionary and lexicalized grammar rules, we prompt LLM to generate target-domain phrases using the given target-domain words and grammar rules. To ensure the generation quality and diversity, during each generation, we select a grammar rule and sample three

LLM Prompt

As a language assistant, you excel at selecting appropriate words from the vocabulary and creating phrases based on the grammar rules. Please strictly observe the constraints.

1. The phrase must follow the given grammar rules and make sense logically.
2. The phrase does not have to be a complete sentence.
3. The generated phrase must be different from the example.
4. The POS of the words in the vocabulary are consistent, so you can only use one word from the vocabulary.

Grammar rules: *NP[NN]* → *DT JJ NN*

Candidate words: ['ring', 'song', 'bicycle']. The part of speech of all candidate words is *NN*.

The step of constructing a phrase based on the grammar rules "*NP[NN]* → *DT JJ NN*" is the following:

determiner "an" and adjective "excellent" and singular noun "toy" combine and get the noun phrase "an excellent toy"

Examples: an excellent toy, a cute movie...

Based on the above information, generate a phase of 3 words.

Figure 4: Prompt template used in LLM generation.

candidate words from the target-domain dictionary whose POS tag is the same as the lexical head of the rule. This not only ensures LLM can select one appropriate word and expand it into a phrase but also guarantees that the generated phrase resembles the style of the target domain since the lexical head is definitely from the target domain. Taking Figure 3 as an example, when the given lexicalized grammar rule is "NP[NN]→ DT NN", we only sample words whose POS tag is "NN" as candidate words.

The detailed prompt template is shown in Figure 4. It includes a series of step-by-step instructions, specified grammar rules, candidate words, and illustrative examples. Notably, we first utilize gpt-3.5-turbo to generate a few phrases under the zero-shot setting and then select high-quality phrases using the same method described as follows.

Stage 3: Checking Phrase Quality. In order to guarantee the quality of the LLM-generated data used for subsequent tree hybridization, further verification is conducted on the generated phrases, considering their length and POS tags. As shown in Figure 3, we only retain phrases that meet the length requirements which are explicitly specified in the prompt. Additionally, the domain dictionary is utilized to confirm that the POS of the words in generated phrases adhere to the given grammar rules. Through these two steps, we ensure that the obtained phrases fully comply with the given grammar rules, enabling accurate reconstruction of the phrase structures based on these rules.

3.3 Data Selection Strategy

To ensure the augmented data used for cross-domain parsing is high-quality and aligns with the style of the target domain, we further select data from the generated pseudo treebank based on three strategies from different perspectives: token, grammar rule, and confidence. Moreover, we further combine the two best-performing strategies, surprisingly resulting in a better performance.

Token-based selection strategy: To maintain the pseudo data resembling the style of the target domain, we use word distribution as the selection strategy. We define the frequency of the words in the domain dictionary as the word distribution and calculate the average frequency of leaf nodes in the constituency tree and set a threshold.

Grammar-rule-based selection strategy: To guarantee the correctness of the newly generated structure after hybridization, we compare the structure of hybrid data with the source-domain treebank and filter out the data whose subtree structures have never existed in the source-domain treebank.

Confidence-based selection strategy: To maintain the semantic correctness of data, we compute confidence for the sentences using the log probability computed by the language model GPT-2 (Radford et al., 2019) and set a threshold.

Token-Grammar-based selection strategy: To maintain both the style and structural information of data, we combine Token-based selection strategy and Grammar-rule-based selection strategy.

4 Experiments

4.1 Experimental Settings

Datasets. Following previous work (Li et al., 2023), we use PTB as the source-domain (newswire) training and dev data. For the target-domain test data, we use the Multi-domain Constituent TreeBank (MCTB) (Yang et al., 2022), including Dialogue, Forum, Law, Literature, and Review domains. To construct the domain dictionary, we collect raw texts from the same sources as the test set, including Wizard (Dinan et al.), Reddit (Völske et al., 2017), ECtHR (Stiansen and Voeten, 2019), Gutenberg[1], and Amazon (He and McAuley, 2016). Each domain dictionary consists of the top K = 10k high-frequency words from the raw texts.

Model Settings. We use Berkeley Neural Parser (Kitaev and Klein, 2018) as the backbone, which is a chart-based parser adopting a self-attentive encoder and a chart-based decoder. Following Fried et al. (2019), all the experiments are based on

[1]https://www.gutenberg.org/

Method	Dialogue	Forum	Law	Literature	Review	Average
GPT-3.5-turbo	70.70	71.56	80.72	72.83	71.24	73.42
Liu and Zhang (2017)	85.56	86.33	91.50	84.96	83.89	86.45
Li et al. (2023)	**87.59**	**87.55**	**93.29**	**87.54**	**85.58**	**88.31**
Kitaev and Klein (2018)	86.30	87.04	92.06	86.26	84.34	87.20
Our method	87.45†	87.50†	92.86†	87.16†	85.44†	88.08†

Table 1: The overall result of our method. The "**Bold**" identifies the best performance, while "Underline" identifies the second-best performance. The † represents the statistical significance compared to baseline with $p < 0.005$. All the results of our experiments are based on pretrained-bert-large and averaged on three distinct seeds.

BERT (Devlin et al., 2019)[2] and we adopt Dan Bikel's randomized parsing evaluation comparator (Noreen, 1989) for statistical significance test.

For each domain, LLM generates 10,000 phrases. During each replacement, we set $p = 0.5$. Our source lexicalized treebank involves subtrees generated from LLM and top $K = 2{,}000$ data from PTB that are closest to the target domain based on the Token-based selection strategy. After three iterations, we choose trees whose constituency label is 'S', containing 20,000 augmented data for each domain and select top $K = 8{,}000$ data based on our selection strategy. We compare 8,000 hybrid data with source-domain data in Appendix A.2.

Comparison Models. In addition to comparing with the Berkeley Neural Parser baseline, we also compare with three competitive models: GPT-3.5-turbo, In-order Parser (Liu and Zhang, 2017), and LLM-enhanced ST (Li et al., 2023). The performance of GPT-3.5-turbo is reported by Li et al. (2023) on few-shot settings, by generating bracketed trees for target domain sentences. We also compare with an additional cross-domain baseline model from Shi et al. (2021a) in the few-shot learning scenarios in Appendix A.4.

4.2 Main Results

The main experimental results are listed in Table 1. Our proposed approach demonstrates substantial and consistent improvements over the Berkeley Neural Parser baseline across all domains, with an average increase of 0.88 points, verifying the effectiveness of our data augmentation method.

Particularly, the most significant improvements are observed in the Dialogue and Review domains, with the increase of 1.15 and 1.10. This is not surprising, as the original sentences in these domains are relatively shorter than in others, aligning well with the characteristics of our generated data. Our approach also demonstrated consistent performance across domains with longer sentence (ranging from 22.01 to 25.59), maintaining an average improvement of 0.72. This consistency highlights the robustness of our method in effectively handling different domains with varying features.

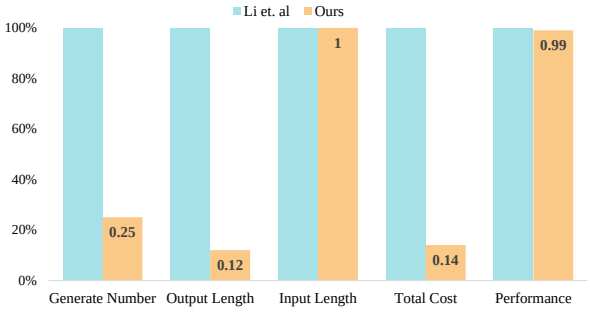


Figure 5: Comparative analysis of costs and performance between Li et al. (2023) and ours. *Generate Number*, *Output Length*, *Intput Length*, and *Total Cost* of generating stage averaged on five target domains.

Notably, GPT-3.5-turbo consistently underperforms relative to all the comparison models across all domains, suggesting that LLMs still have limitations in parsing. Our approach successfully mitigates the limitations of LLMs in constituency parsing, leveraging their strengths to enhance cross-domain parsing performance. When comparing our method to LLM-enhanced ST (Li et al., 2023), our approach achieves comparable performance at a significantly lower cost. We conduct a further discussion of the cost comparison in the next part.

4.3 Cost Analysis

Li et al. (2023) shares a similar idea of leveraging LLMs to enhance cross-domain parsing with data augmentation. They prompts the LLM to generate target-domain raw sentences, which are then incorporated into the self-training process of a small con-

[2]https://huggingface.co/bert-large-uncased

Method	Dialogue	Forum	Law	Literature	Review	Average
Kitaev and Klein (2018)	86.30	87.04	92.06	86.26	84.34	87.20
Substitution	86.93	87.14	92.47	86.85	84.98	87.67
Our method	**87.45**	**87.50**	**92.86**	**87.16**	**85.44**	**88.08**
w/o *Hybridization*	86.76	87.25	92.41	86.56	84.88	87.57
w/o *Generation*	86.25	86.82	92.08	86.23	84.19	87.11
w/o *Selection*	86.81	87.26	92.49	86.73	84.89	87.64
+ *Token-Based*	87.20	87.40	92.71	86.86	85.19	87.87
+ *Rule-Based*	87.13	87.35	92.67	87.05	85.03	87.85
+ *Conf-Based*	86.91	87.33	92.60	86.87	84.93	87.73

Table 2: Results of the effects of substitution, hybridization, generation, and different data selection strategies on the five target domains. Substitution represents using only the subtrees generated from LLM during replacement. "w/o Hybridization" represents using only the LLM-generated subtrees as additional augmented data. "w/o Generation" represents hybridization only on the source-domain treebank. The rest of the results represent different selection strategies, and our method is experimented with based on the Token-Grammar-based strategy.

stituency parser. While their approach has achieved leading results across various fields, we think it has come upon a significantly larger cost.

Figure 5 provides an intuitive comparison of the associated costs. Firstly, their approach involves 4 iterations of self-training and utilizes LLM to generate 10,000 raw sentences in each iteration, whereas ours only requires 10,000 phrases at a time. So our generation number is just 25% of theirs. Furthermore, their approach generates complete sentences, while ours only generates phrases. This results in our average output length being just 12% of theirs (3.82 tokens compared to 31.93 tokens). For input length, we only consider prompt instead of instruction, which in essentially the same for both of us (about 100 tokens). These differences lead to a significant reduction in overall costs for our method. Theoretically, according to the prices given by OpenAI[3], our LLM-related expanses for generating amount to only 14% of those incurred by theirs. Despite this substantial efficiency gain, our method maintains competitive performance, with an average accuracy decrease of less than 1%. The efficiency gains make our method more practical and scalable for real-world applications where computational resources and time are often limited.

5 Analysis

5.1 w/ Hybridization vs. w/o Hybridization

To analyze the effectiveness of introducing tree hybridization, we report the results of using only the LLM-generated subtrees as additional augmented data, without generating more data via hybridization, as shown in Table 2. We find that removing hybridization consistently decreases performance across all domains. This demonstrates that our tree hybridization method effectively utilizes newly LLM-generated data and the existing treebank to produce more diverse parsing trees. We provide a further analysis in Analysis 5.8 and Appendix A.3.

5.2 w/ Generation vs. w/o Generation

We also observe that compared to the baseline (Kitaev and Klein, 2018), the results of "w/o Hybridization" still show some improvement. This indicates that our method of generating subtrees via LLM can effectively provide additional guidance for cross-domain constituency parsing by introducing words in the target domain. Notably, since the LLM generating directly utilizes the structures from the source-domian, it will not bring new structures. To investigate the effectiveness of generation further, we report the results of performing hybridization on the source-domain treebank without LLM generating, as is shown in Table 2. We find that removing LLM generation results in degradation to performance consistent with baseline.

5.3 Substitution vs. Hybridization

We compare our method with the substitution method. Concretely, we set the probability p of selecting subtrees generated from hybridization to 0, so that only subtrees generated from LLM can be used as alternatives during each replacement. As is shown in Table 2, our method outperformed it by

[3]According to the prices given on the official website of OpenAI, the cost of output is three times that of input.

Method	Dialogue	Forum	Law	Literature	Review	Average
Kitaev and Klein (2018)	86.30	87.04	92.06	86.26	84.34	87.20
Our method w/ gpt-3.5-turbo	**87.45**	**87.50**	**92.86**	**87.16**	**85.44**	**88.08**
Our method w/ LLaMa-3-70B	87.38	87.48	92.75	87.09	85.34	88.01
Our method w/ QWEN-2-72B	87.33	87.47	92.71	87.05	85.33	87.98

Table 3: Results of different LLMs on the five target domains.

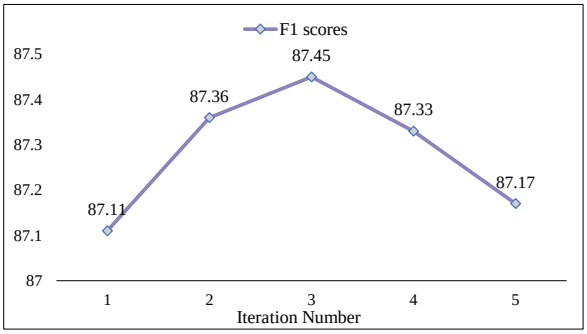


Figure 6: Results of different iteration numbers in hybridization stage on the Dialogue domain.

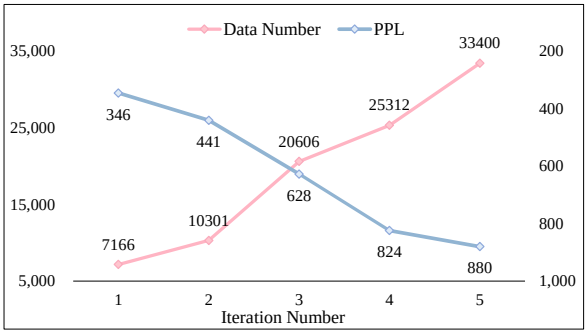


Figure 7: Statistic and semantic correctness over iterations on the Dialogue domain.

0.41 points averaged on five domains. The result demonstrates that hybridization can generate more complex structures with diversity for cross-domain constituency parsing, aiding the parser in learning more accurate and domain-specific information.

5.4 Impact of Different Data Selection Strategies

To investigate the effect of data selection strategy, we use three data selection strategies from different perspectives: token, grammar rule, and confidence. Moreover, we combine the two best-performing strategies, resulting a better performance. As is shown in Table 2, we find that when only one selection strategy is used, the token-based and rule-based selection strategy shows the most improvement, which ensures the style of the target domain and the correctness of syntactic constituency parsing tree structures, respectively. Combining two strategies can make complementary contributions, resulting in the optimal selection strategy.

5.5 The Iteration of Hybridization

We investigate the effect of each iteration of hybridization on the Dialogue domain. As shown in Figure 6, during the first three iterations, the performance exhibits a continuous improvement. However, it declines with continued iterations. For further investigation, we analyze the semantic correctness of data produced by hybridization as iterations increase. Specifically, we randomly sample 1K hybrid data generated in each iteration and use GPT-2 to compute the average perplexity. As shown in Figure 7, the perplexity increases over iterations, indicating the decline of semantic correctness. Since both the quality and quantity of the data are key factors affecting performance, the performance improves at first due to the increase in data quantity, but subsequently declines as data quality significantly diminishes. We also provide some examples from different iterations in Appendix A.5.

5.6 Results on Other LLMs

In addition to closed-source LLMs, we also verify the effectiveness of our method with two popular open-sourced LLMs (LLaMA-3-70B and QWEN-2-72B). As indicated in Table 3, the results show consistently and substantial improvements with these LLMs, indicating the effectiveness of using our method with open-sourced LLMs.

5.7 Results on the Other Dataset

We also report our results on the English Web Treebank (EWT) (Silveira et al., 2014), containing data from five genres of web media: Yahoo! answers, emails, newsgroups, reviews, and weblogs. As is shown in Table 4, our results show substantial improvements using our method on the five domains

Method	Answers	Email	Newsgroup	Reviews	Weblog
Kitaev and Klein (2018)	88.47	87.55	90.57	89.27	91.08
Our method	89.14	87.89	91.26	90.19	91.96

Table 4: Results on the EWT dataset.

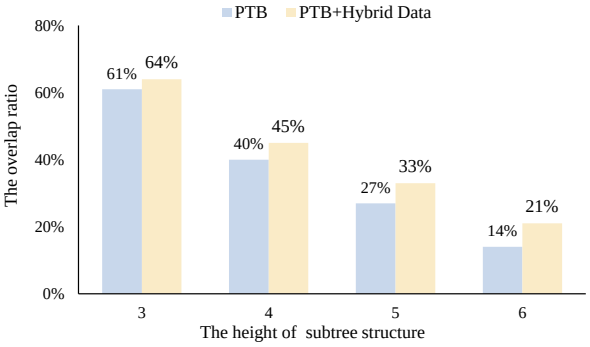


Figure 8: The ratio of overlapping subtree structures between train set and test set on the Dialogue domain.

of the EWT treebank, which further verify the effectiveness of our method.

5.8 Structures Generated via Hybridization

To further investigate the impact of hybridization, we examine whether it can produce more diverse structures that align with the target domain. Specifically, we compare the proportion of overlapping subtree structures between the training set and the test set before and after adding augmented data produced via tree hybridization. As shown in Figure 8, with our tree hybridization method, the overlap ratio of subtree structures with heights ranging from 3 to 6 increases consistently. This indicates that our tree hybridization approach can generate new structures that meet the target domain, and as the height of subtree structure increases, the overlap ratio also increases continuously.

6 Conclusion

In this paper, we introduce a lightweight data augmentation method for cross-domain constituency parsing with LLM generation and tree hybridization. First, we guide the LLM to generate high-quality target-domain constituency subtrees by providing grammar rules and words in the target-domain as instructions. Then, we propose a tree hybridization method to further produce a large number of diverse instances by fully leveraging both target-domain LLM-generated subtrees and the existing source-domain treebank. Experimental results demonstrate that our method consistently improves performance across five target domains while maintaining high efficiency.

7 Limitations

There remain unexplored experiments of interest, such as investigating the differences between zero-shot and few-shot approaches in LLM generation, which we plan to address in future studies.

Acknowledgments

We thank the anonymous reviewers for their valuable comments and suggestions. We are very grateful to Houquan Zhou to discuss with us and provide valuable suggestions. This work was supported by National Natural Science Foundation of China (Grant No. 62306202 and 62176173), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 23KJB520034), and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Amirhossein Abaskohi, Sascha Rothe, and Yadollah Yaghoobzadeh. 2023. LM-CPPF: Paraphrasing-guided data augmentation for contrastive prompt-based few-shot fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 670–681, Toronto, Canada. Association for Computational Linguistics.

Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. 2023. Constituency parsing using llms. *Preprint*, arXiv:2310.19462.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020.

Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Daniel Fried, Nikita Kitaev, and Dan Klein. 2019. Cross-domain generalization of neural constituency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Jianling Li, Meishan Zhang, Peiming Guo, Min Zhang, and Yue Zhang. 2023. LLM-enhanced self-training for cross-domain constituency parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8174–8185, Singapore. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. In-order transition-based constituent parsing. *Transactions of the Association for Computational Linguistics*, 5:413–424.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

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.

Eric W. Noreen. 1989. Computer-intensive methods for testing hypotheses : an introduction.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2020. On the role of supervision in unsupervised constituency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7611–7621, Online. Association for Computational Linguistics.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2021a. Substructure substitution: Structured data augmentation for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3494–3508, Online. Association for Computational Linguistics.

Tianze Shi, Ozan İrsoy, Igor Malioutov, and Lillian Lee. 2021b. Learning syntax from naturally-occurring bracketings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2941–2949, Online. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Øyvind Stiansen and Erik Voeten. 2019. ECtHR judgments.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Association for Computational Linguistics.

Fanqi Wan, Xinting Huang, Tao Yang, Xiaojun Quan, Wei Bi, and Shuming Shi. 2023. Explore-instruct: Enhancing domain-specific instruction coverage through active exploration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9435–9454, Singapore. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Pengcheng Yin, and Graham Neubig. 2018. A tree-based decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4772–4777, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Xin Xin, Jinlong Li, and Zeqi Tan. 2021. N-ary constituent tree parsing with recursive semi-Markov model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2631–2642, Online. Association for Computational Linguistics.

Benfeng Xu, Chunxu Zhao, Wenbin Jiang, PengFei Zhu, Songtai Dai, Chao Pang, Zhuo Sun, Shuohuan Wang, and Yu Sun. 2023. Retrieval-augmented domain adaptation of language models. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 54–64, Toronto, Canada. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Zhiyang Xu, Andrew Drozdov, Jay Yoon Lee, Tim O'Gorman, Subendhu Rongali, Dylan Finkbeiner, Shilpa Suresh, Mohit Iyyer, and Andrew McCallum. 2021. Improved latent tree induction with distant supervision via span constraints. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4818–4831, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sen Yang, Leyang Cui, Ruoxi Ning, Di Wu, and Yue Zhang. 2022. Challenges to open-domain constituency parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 112–127, Dublin, Ireland. Association for Computational Linguistics.

Songlin Yang and Kewei Tu. 2022. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2403–2416, Dublin, Ireland. Association for Computational Linguistics.

Le Zhang, Zichao Yang, and Diyi Yang. 2022. TreeMix: Compositional constituency-based data augmentation for natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5243–5258, Seattle, United States. Association for Computational Linguistics.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural crf constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI-PRICAI-2020. International Joint Conferences on Artificial Intelligence Organization.

Models	**Constituency**	**Lexicalized**
Our method	87.17	87.45

Table 5: Results of using constituency tree VS lexicalized tree on the Dialogue domain.

A Appendix

A.1 Constituency Tree vs. Lexicalized Tree

To prove the effectiveness of using the lexicalized tree compared to the constituency tree in our method, we perform our tree hybridization method based on the constituency tree on the Dialogue domain, as shown in Table 5. The result demonstrates that using the lexicalized tree, which enhances the constraints of replacement, significantly guarantees the semantic correctness of augmented data.

A.2 Equivalence between Source-domain Data and Augmented Target-domain Data

Evaluating the equivalence between augmented target-domain data and source-domain data is crucial for assessing data augmentation methods. Specifically, we establish a baseline using a parser trained on the full set of labled source-domain data. We then determine how much source-domain data, in combination with 8,000 target-domain data via hybridization, is necessary to match this baseline performance. The results on the Dialogue domain indicate that using 25,000 source-domain data alongside 8,000 augmented target-domain data achieves results comparable to the baseline (trained on 40,000 source-domain data). This suggests that the effectiveness of 8,000 augmented target-domain data is equivalent to approximately 15,000 source-domain data (computed as 40,000 - 25,000).

A.3 Improvements on Constituency Labels

To provide a more in-depth analysis, we investigate the recall and precision of the 6 most common types of constituents before and after data augmentation on the Dialogue domain. As is shown in Table 6 and Table 7, the recall and precision improves consistently across different types of constituents, indicating the effectiveness and robustness of our data augmentation method.

A.4 Additional Cross-domain Baseline

We also compare our method with Shi et al. (2021a), which proposes a substructure substitution method for data augmentation. To adapt their methods which designed for few-shot learning scenarios to our zero-shot learning scenarios, we maintain the same settings as in their paper, sampling 50 sentences from MCTB for training and use the remaining 950 sentences for testing. As indicated in Table 8, our results significantly surpass theirs on the five target domains.

A.5 Examples from Different Iterations

To further investigate the quality of augmented data produced in different iterations of tree hybridization, we also provide examples from the first, third, and fifth iteration on the Dialogue domain. As is shown in the Table 9, the sentences become longer and more complex over the iterations, while at the same time becoming less coherent or containing some ambiguities. It is like the famous linguistic example, "Colorless green ideas sleep furiously" by Noam Chomsky, which illustrates a sentence that is grammatically correct but semantically nonsensical, making it difficult to understand. We speculate that such issues will become more prevalent as iterations progress.

Models	S	NP	VP	PP	ADVP	SBAR
Kitaev and Klein (2018)	89.55	89.55	78.40	88.11	80.40	89.17
Our method	89.86	90.49	80.65	89.07	83.74	90.16

Table 6: The recall of different types of constituents before and after augmentation on the Dialogue domain.

Models	S	NP	VP	PP	ADVP	SBAR
Kitaev and Klein (2018)	85.27	92.31	85.26	89.83	87.42	89.35
Our method	85.91	92.90	85.36	92.59	87.96	89.98

Table 7: The precision of different types of constituents before and after augmentation on the Dialogue domain.

Method	Dialogue	Forum	Law	Literature	Review	Average
Kitaev and Klein (2018)	86.22	86.96	92.08	86.35	84.43	87.21
Shi et al. (2021a)	86.64	87.17	92.36	86.55	84.74	87.49
Our method	87.37	87.46	92.86	87.20	85.50	88.08

Table 8: Results of Shi et al. (2021a) and our method in the few-shot learning scenarios.

Iter	Sentence
1	1. The popcorn of this tastes quite delicious. 2. It is a very talented artist. 3. It causes a surprisingly difficult problem. 4. Her movie plays quite well in concerts. 5. Excited students in the classroom attend the event.
3	1.A depth in science is incredibly impressive. 2. A painting in Rome inspired her own success. 3. The ideas on creativity build a very modern house. 4. Helpful books sold the total supply of the stuff last year and again this year. 5. Interested students in university hold my favorite book in library.
5	1. Students developing the software hang an expensive painting in museum. 2. I all of the cities finally put the fantastic book in library. 3. The wizards have appeared in four nba finals, and sadly it had the scripts! 4. He published a always fascinating book on a rainy afternoon. 5. Smarter doctors question me all of their diseases about math.

Table 9: Exampls from different iteration. The semantic nonsense in the sentences is highlight.