LLM-Assisted Data Augmentation for Chinese Dialogue-Level Dependency Parsing

Meishan Zhang
Harbin Institute of Technology
(Shenzhen)
Institute of Computing and Intelligence
mason.zms@gmail.com

Gongyao Jiang
Tianjin University
School of New Media and
Communication
jianggongyao@gmail.com

Shuang Liu
Tianjin University
College of Intelligence and Computing
shuang.liu@tju.edu.cn

Jing Chen
Information Center of Ministry of
Science and Technology
chenjjing83@sina.com

Min Zhang*
Harbin Institute of Technology
(Shenzhen)
Institute of Computing and Intelligence
zhangmin2021@hit.edu.cn

Dialogue-level dependency parsing, despite its growing academic interest, often encounters underperformance issues due to resource shortages. A potential solution to this challenge is data augmentation. In recent years, large language models (LLMs) have demonstrated strong capabilities in generation, which can facilitate data augmentation greatly. In this study, we focus

* Corresponding author.

Action Editor: Carlos Gómez Rodríguez. Submission received: 10 September 2023; revised version received: 24 January 2024; accepted for publication: 28 February 2024.

https://doi.org/10.1162/coli_a_00515

on Chinese dialogue-level dependency parsing, presenting three simple and effective strategies with LLM to augment the original training instances, namely word-level, syntax-level, and discourse-level augmentations, respectively. These strategies enable LLMs to either preserve or modify dependency structures, thereby assuring accuracy while increasing the diversity of instances at different levels. We conduct experiments on the benchmark dataset released by Jiang et al. (2023) to validate our approach. Results show that our method can greatly boost the parsing performance in various settings, particularly in dependencies among elementary discourse units. Lastly, we provide in-depth analysis to show the key points of our data augmentation strategies.

1. Introduction

Dialogue-level dependency parsing, which extends vanilla dependency parsing (Marcus et al. 1994; Xue et al. 2005; Nivre 2005; McDonald et al. 2013) to dialogue texts, has attracted considerable attention in recent years (Afantenos et al. 2015; Asher et al. 2016; Davidson, Yu, and Yu 2019; Jiang et al. 2023). Given a piece of dialogue text, the task is to build a structural dependency tree covering not only the inner-sentence words but also the words across utterances by well-designed machine learning models. For the Chinese language, Jiang et al. (2023) present the initial work on dialogue-level dependency parsing. Figure 1 shows an example. A dialogue text is split into elementary discourse units (EDUs), where the inner-EDU dependencies reflect sentence-level syntax, and the inter-EDU dependencies reflect discourse structure.

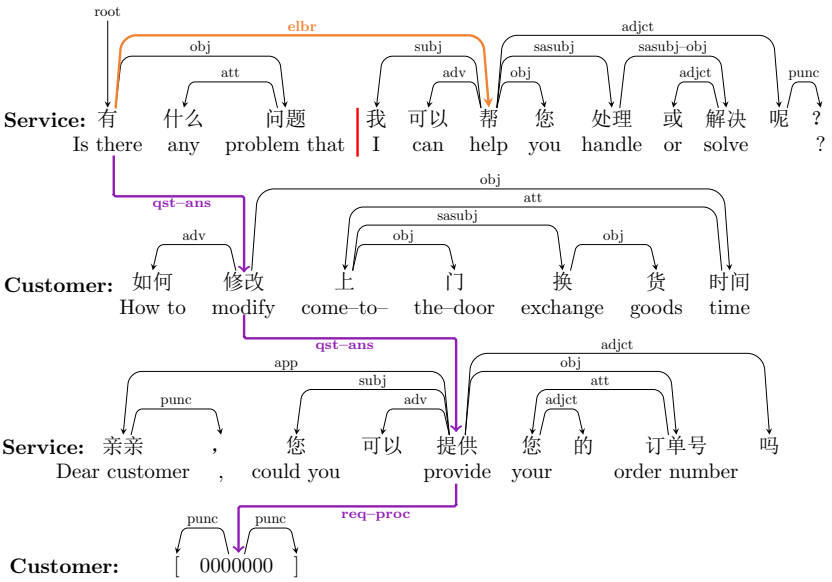


Figure 1
A fragment sample of dialogue-level dependency tree in Jiang et al. (2023), where the inter-EDU dependencies are emphasized by bold lines, and "elbr", "qt-ans", "req-proc" denote elaboration, question-answer, and requirement-process, respectively, as referred to Jiang et al. (2023). Note that the first utterance of the dialogue has two EDUs, as split by |.

A major problem with building a high-performance dialogue-level dependency parsing model is the relatively small amount of training corpora available. Such dependency treebank annotation is remarkably difficult and can be extremely expensive. It requires a high-degree background in linguistics, and long-distance global observations are needed to determine inter-utterance dependencies. Using expert annotation, Jiang et al. (2023) build a benchmark corpus containing only 850 dialogues with great effort. The small scale of the training corpus is insufficient for standard supervised learning. Jiang et al. (2023) exploited 50 instances as training and the remaining instances as evaluation, reporting a result of 88.20 and 55.73 by the inner-EDU and inter-EDU labeled attachment scores (LASs), respectively, which indicates that accurate dialogue understanding is still a long way away.

Data augmentation can be one prospective method to fix this problem. Given extremely limited (or even no) annotated instances, data augmentation aims to produce a number of pseudo training instances automatically (Scudder 1965; Tanner and Wong 1987). The line of work has been applied successfully to a number of NLP tasks (Liu et al. 2020; Feng et al. 2021; Shorten, Khoshgoftaar, and Furht 2021). The key to success is to ensure the diversity as well as the quality of the automatically generated training instances, enriching the training corpus effectively. Recently, large language models (LLMs) have shown great potential for data augmentation in NLP (Whitehouse, Choudhury, and Aji 2023; Dai et al. 2023) by their strong capabilities in text generation. With appropriate prompting, we can produce several transformed texts with controllable variations.

In this work, we make an initial attempt at data augmentation in Chinese dialogue-level dependency parsing, aiming to construct a number of pseudo instances automatically to supplement the training data. Our key idea is to leverage the generation ability of LLMs to obtain high-quality transformations of a gold-standard dependency tree. On the basis of the characteristics of dialogue-level dependency parsing, we transform the original dependency tree to new well-formed dependency trees gradually along three different levels: word, syntax, and discourse levels, which correspond to the alternations of surface word information, inner-EDU syntactic information, and inter-EDU discourse information, respectively.

We conduct experiments on the benchmark dataset of Jiang et al. (2023), following their work as the start-up baseline. Two settings are evaluated, namely, zero-shot and few-shot according to their work. The zero-shot setting is only with silver training instances that are constructed by rules, and the few-shot setting includes an extra 50 gold-standard training instances. We choose the LLM GPT-3.5-Turbo mainly to drive our data augmentation. The results show that our data augmentation methods are able to boost the performance in both settings, especially on inter-EDU dependencies. In the zero-shot setting, our method can achieve an improvement of 3.04 in inter-EDU LAS. Under the few-shot setting, the increase reaches to 3.85. We also conduct experiments based on Llama2-7B and Qwen-7B, and the results are consistent with GPT-3.5-Turbo. All our datasets as well as the source code are available for research purposes.[1]

2. Background

Given a text $\boldsymbol{x} = [w_1, w_2, \ldots, w_n]$, dependency parsing aims to establish a direct, labeled dependency tree between words in the text. Each word w_i ($i \in [1, n]$) has exactly

1 https://github.com/Zzoay/DialogDepAug.

one head word except the root word which has none, that is, dependency $w_i^{\frown}w_h (i < h)/w_h^{\frown}w_i (i > h)$. There is only one root word in the given text. Traditionally, dependency parsing handles mostly sentences. Recently, there is a growing interest in extending it to paragraphs and dialogues, uniting inner-sentence syntactic/semantic as well as discourse structures (Afantenos et al. 2015; Asher et al. 2016; Davidson, Yu, and Yu 2019; Jiang et al. 2023).

Here, we focus on dialogue-level dependency parsing in Chinese, as shown in Figure 1. The dependency trees also maintain the projective property, i.e., no dependencies cross when they are all depicted above the text. Jiang et al. (2023) present seminal work on this task. Given an input dialogue text, they divide it into a sequence of EDUs. For the inner-EDU dependencies, they use syntactic dependencies following the guideline of Jiang et al. (2018). While for the inter-EDU dependencies, they define a set of discourse labels according to the characteristics of Chinese dialogue.

3. Baseline Parser

It is feasible to solve Chinese dialogue-level dependency parsing with traditional sentence-level dependency parsing models directly. However, this straightforward method would be inefficient with respect to both speed and performance because of the increased numbers of input words as well as the dependency labels. Thus, a hierarchical decoding of inner-EDU and inter-EDU dependencies is more suitable. In this work, we extend the state-of-the-art biaffine parser (Dozat and Manning 2016) with the support of pretrained language model (PLM) into our Chinese dialogue-level dependency parser. The parser is a slightly modified version of Dozat and Manning (2016).

In more detail, given an input dialogue text $\boldsymbol{x} = [w_1, w_2, \ldots, w_n]$ and its EDU-level sequence $\boldsymbol{x} = [E_1, E_2, \ldots, E_m]$, where m is the number of EDUs and $\mathrm{E}_k = \left[w_{k,1}, \ldots, w_{k,s_k}\right]$ ($k \in [1, m]$) indicates the words covered by one EDU, we illustrate the baseline parser as follows by an encoding-decoding view.

Encoding. First, we let $\boldsymbol{x}$ go through a typical PLM, resulting in general contextualized word representations $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$. Then, we adopt a multi-layer BiLSTM to obtain high-level abstract word representations $\boldsymbol{h} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n]$. Based on these representations, we derive dependency-aware and head-aware features by multilayer perceptron (MLP) for each word, i.e., $\boldsymbol{z}^d = \left[\boldsymbol{z}_1^d, \boldsymbol{z}_2^d, \ldots, \boldsymbol{z}_n^d\right]$ and $\boldsymbol{z}^h = \left[\boldsymbol{z}_1^h, \boldsymbol{z}_2^h, \ldots, \boldsymbol{z}_n^h\right]$:

$$\begin{aligned}
\boldsymbol{e}_{1:n} \rightarrow \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n &= \mathrm{PLM}\left(w_1, w_2, \ldots, w_n\right) \\
\boldsymbol{h}_{1:n} \rightarrow \boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n &= \mathrm{BiLSTM}^{\times L}\left(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n\right) \\
\begin{cases} \boldsymbol{z}_{1:n}^d \rightarrow \boldsymbol{z}_1^d, \boldsymbol{z}_2^d, \ldots, \boldsymbol{z}_n^d = \mathrm{MLP}^{\times K}\left(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\right) \\ \boldsymbol{z}_{1:n}^h \rightarrow \boldsymbol{z}_1^h, \boldsymbol{z}_2^h, \ldots, \boldsymbol{z}_n^h = \mathrm{MLP}^{\times K}\left(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\right) \end{cases}
\end{aligned} \tag{1}$$

Decoding. Next, we come up the decoding of dialogue-level dependency parsing tree, which involves two steps. First, we perform inner-EDU dependency parsing. For each $\mathrm{E}_k = \left[w_{k,1}, \ldots, w_{k,s_k}\right]$, we obtain their corresponding dependency-aware and head-aware representations: $\boldsymbol{z}_{k,1:k,s_k}^d = \boldsymbol{z}_{k,1}^d, \ldots, \boldsymbol{z}_{k,s_k}^d$ and $\boldsymbol{z}_{k,1:k,s_k}^h = \boldsymbol{z}_{k,1}^h, \ldots, \boldsymbol{z}_{k,s_k}^h$ by

straightforward indexing, and then calculate the candidate head scores for each word $w_{k,j}$ by biaffine operation:

$$\begin{aligned} \boldsymbol{o}_{k,j}^{\mathrm{IN}} &= \boldsymbol{z}_{k,1:k,s_k}^{h} \boldsymbol{U}^{\mathrm{IN}} \boldsymbol{z}_{k,j}^{d} + \boldsymbol{z}_{k,1:k,s_k}^{h} \boldsymbol{u}^{\mathrm{IN}} \\ \boldsymbol{o}_{k,j}^{\mathrm{IN,ARC}} &= \sum_{l} \boldsymbol{o}_{k,j}^{\mathrm{IN}}[\cdot][l] \end{aligned} \tag{2}$$

where $\boldsymbol{U}^{\mathrm{IN}}$ and $\boldsymbol{u}^{\mathrm{IN}}$ are learnable parameters, the candidate heads of each word $w_{k,j}$ are limited inside the EDU, and dependency labels are restricted to syntactic labels only. Note that $\boldsymbol{o}_{k,j}^{\mathrm{IN}}$ includes scores for both head candidates and syntactic labels, that is, $\boldsymbol{o}_{k,j}^{\mathrm{IN}}[i]$ is a vector with scores across all candidate labels. Thus, $\boldsymbol{o}_{k,j}^{\mathrm{IN}}$ is a two-dimensional tensor here. During inference, we first exploit the minimum spanning tree algorithm based on $\boldsymbol{o}_{k,j}^{\mathrm{IN,ARC}}$ to build a well-formed dependency tree, and then assign each dependency with a label according to the second-dimensional values.

Second, for the inter-EDU dependency parsing, we also use the biaffine operation and the same inference strategy but with different dependency candidates and labels. We extract two sequences of features: $\boldsymbol{z}_{r_1:r_m}^{d} = \boldsymbol{z}_{1,r_1}^{d}, \ldots, \boldsymbol{z}_{m,r_m}^{d}$ and $\boldsymbol{z}_{r_1:r_m}^{h} = \boldsymbol{z}_{1,r_1}^{h}, \ldots, \boldsymbol{z}_{m,r_m}^{h}$, where r_* denotes the root word of a given EDU. Based on the root word sequence of EDUs, we build inter-EDU dependencies as follows:

$$\begin{aligned} \boldsymbol{o}_{k}^{\mathrm{IT}} &= \boldsymbol{z}_{r_1:r_m}^{h} \boldsymbol{U}^{\mathrm{IT}} \boldsymbol{z}_{r_k}^{d} + \boldsymbol{z}_{r_1:r_m}^{h} \boldsymbol{u}^{\mathrm{IT}} \\ \boldsymbol{o}_{k}^{\mathrm{IT,ARC}} &= \sum_{l} \boldsymbol{o}_{k}^{\mathrm{IT}}[\cdot][l] \end{aligned} \tag{3}$$

where $\boldsymbol{U}^{\mathrm{IT}}$ and $\boldsymbol{u}^{\mathrm{IT}}$ are learnable parameters, $\boldsymbol{o}_{k}^{\mathrm{IT}}$ is also a two-dimensional tensor: one spreading inter-EDU dependency heads and the other spreading the corresponding labels, and $\boldsymbol{o}_{k}^{\mathrm{IT,ARC}}$ is used to derive the skeleton of a dependency tree.

By using the above two-step hierarchical decoding, the efficiency is largely improved by filtering out the head and label candidates.

Training. We exploit the standard cross-entropy loss as the training objective, where the losses of dependency arc recognition and label classification are computed separately. Given the output $\boldsymbol{o}_{*}^{*}$ (either $\boldsymbol{o}_{k,j}^{\mathrm{IN}}$ or $\boldsymbol{o}_{k}^{\mathrm{IT}}$), we use softmax over $\boldsymbol{o}_{*}^{*,\mathrm{ARC}}$ and $\boldsymbol{o}_{*}^{*}[\boldsymbol{y}_h]$ ($\boldsymbol{y}_h$ is the correct dependency tree) to calculate the probabilities of all candidate dependency heads and syntactic/discourse labels, respectively. Our training strategy is essentially equivalent to that of Dozat and Manning (2016).

Particularly, the training of our baseline parser can be divided into two parts, that is, inner-EDU and inter-EDU dependency parsing. The inner-EDU parsing may receive supervised signals from the existing syntactic treebank, as well as the inner-EDU dependencies in the 50 gold-standard training instances provided by Jiang et al. (2023). This part could be trained adequately. While for the inter-EDU dependency parsing, there are only very few training instances. We follow Jiang et al. (2023), using their rule-based silver training corpus along with the same 50 gold-standard instances. For the construction of this benchmark corpus, one can refer to their paper for the details.

4. LLM-Assisted Data Augmentation

As mentioned in Jiang et al. (2023), they have annotated only a total of 850 gold-standard dialogue-level dependency trees for training and evaluation, at great cost. There are two main reasons for this. First, dependency-style treebank annotation requires a high degree of linguistic background, which limits the pool of annotators, and the cost of training an annotator is also expensive. Second, discourse-level dependencies often involve long-term deep understandings of dialogue texts, making the annotation process extremely challenging. As a result, dialog-level dependency parsing in a low-resource setting is more practical and desirable.

Data augmentation is one popular strategy in low-resource settings for a variety of NLP tasks (Liu et al. 2020; Feng et al. 2021; Shorten, Khoshgoftaar, and Furht 2021). The main idea is to produce a number of high-quality and high-diversity training instances by transforming the existing training instances. To transform the natural-language training instances, LLMs have been shown great potential because of their strong capability in sentence rewriting. The use of LLMs for data augmentation has been suggested in sentence classification (Dai et al. 2023) and commonsense reasoning (Whitehouse, Choudhury, and Aji 2023). In this work, we exploit the training dependency trees as the base, reforming them gradually by LLM prompting.

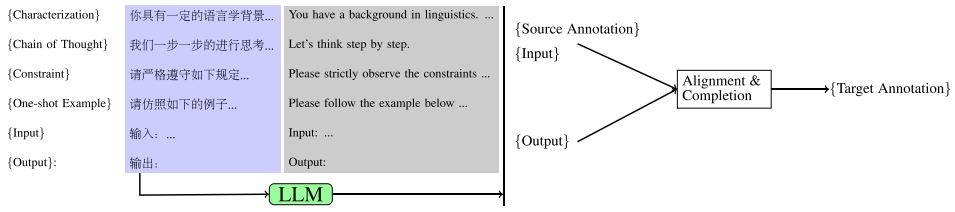
There are three types of information in a dialogue-level dependency tree: (1) words, (2) syntax dependencies and (3) discourse dependencies. Here we transform one dependency tree to generate new dependency trees by disturbing the three types of information. We call the three levels of alteration as follows: word-level, syntax-level, and discourse-level, respectively. The word-level alteration is the most basic, where the higher-level one may also include lower-level variations. Figure 2 shows the overall architecture of our method, accompanied by three examples to illustrate the three augmentation strategies.

As shown in Figure 2a, all three-level data augmentation mechanisms adopt a universal style of LLM prompting to rewrite the source input text, that is, "{`Characterization`} {`Chain of Thought (CoT)`} {`Constraint`} {`One-Shot Example`} {`Input`} $\longrightarrow$ {`Output`}":

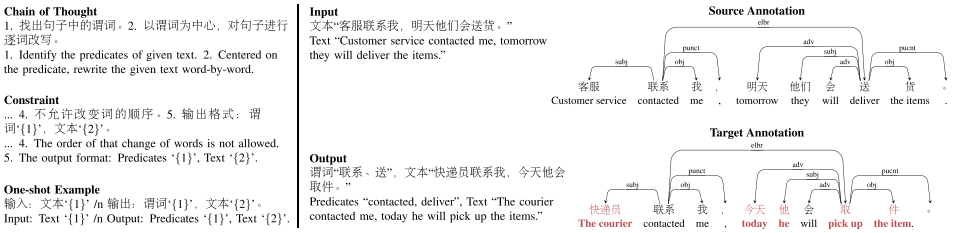
- `Characterization`: We initially provide a characterization to the LLM, aiming to stimulate the language understanding ability of the LLM. Concretely, we prompt the LLM: "你具有一定的语言学背景，精通中文文本理解，尤其是依存分析。 (You have a background in linguistics and are proficient in understanding Chinese text, especially dependency parsing.)". This part is the same over all augmentations.

- `CoT`: A CoT component is exploited to achieve a more reasonable rewriting goal, which is believed to significantly enhance the generative capability of the LLM (Wei et al. 2022). This part starts with "我们一步一步的进行思考 (Let's think step by step)," followed by more detailed instructions which are different across the three strategies.

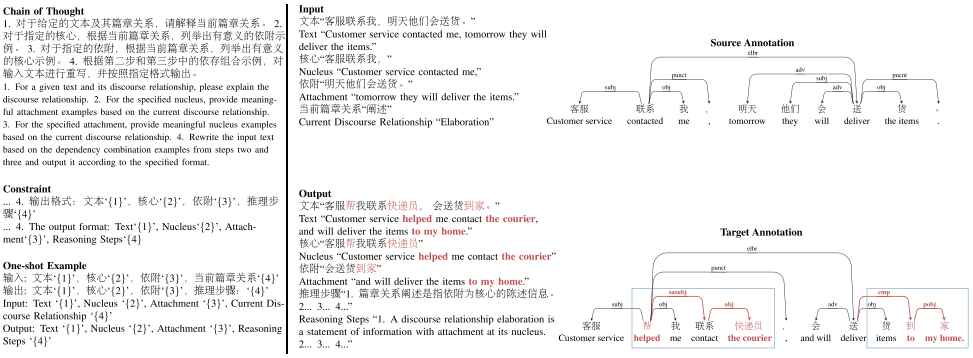
- `Constraint`: More importantly, we impose certain constraints to control the LLM output within a fixed language, style, and format, ensuring that the generated text does not undergo language or style shifts and meanwhile making the follow-up information extraction more



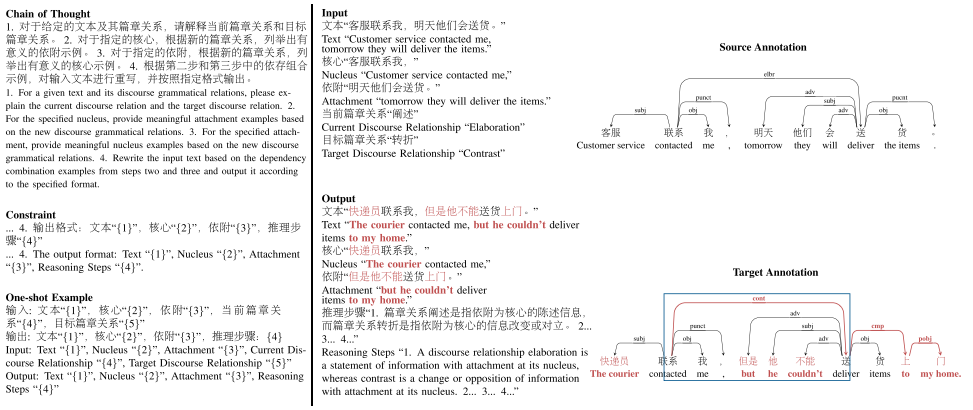
(a) Overall Architecture



(b) Word–Level Transformation



(c) Syntax–Level Transformation



(d) Discourse–Level Transformation

Figure 2
LLM-assisted data augmentation.

convenient. This part starts with "请严格遵守如下规定。 (Please strictly observe the constraints.)." There are three common constraints: "1. 使用中文并遵循原始文本的风格. 2. 上下文逻辑应当合理. 3. 不要对给定文本进行回复或续写 (1. Use Chinese and adhere to the style of the original text. 2. The contextual logic should make sense. 3. Do not respond to or continue the given text)."

- `One-Shot Example`: Although the aforementioned method can effectively enhance and standardize the generation of the LLM, it is still difficult to ensure the stability of the output due to the LLM's tendency to use random sampling during the decoding process. To address this issue, we manually design an example to guide the LLM generation in accordance with the above requirements. This part starts with "请仿照如下的例子。 (Please follow the example below.)". Here the selection is totally empirical and random, where the concrete examples are offered in Section 5.1.4. In this section, we only depict the format of the example for each augmentation strategy.

We notice that the results of rewriting can be inevitably different when prompts vary even slightly. The phenomenon is acceptable since data augmentation always encounters this issue: Picking up pseudo instances often results in such randomness due to the uncertainty of raw input selection (Feng et al. 2021). The key success of data augmentation is to ensure the high quality of generated outputs. Furthermore, all three strategies can be applied either to the individually segmented EDUs, to the entire utterance, or even to the whole dialogue. The only difference lies in the input and the one-shot example in terms of the prompt. However, our preliminary experimental findings indicate that the results are unsatisfactory when applied to the entire dialogue. The reason might be attributed to the fact that with the increase in sample length, the complexity of executing accurate augmentations escalates. In the following, we describe the three-level data augmentation mechanisms by LLM prompting in detail.

4.1 Word-Level Transformation

By substituting a word into an alternative word, while maintaining the same syntactic and discourse structure, we can obtain a transformed new dependency tree. For the word-level substitution, high-quality word substitution is the key to success. Previous studies have often exploited semantically closed words to replace the original words, largely ignoring the current contexts (Liu et al. 2020). With the help of LLMs, the issue can be greatly reduced. Figure 2b shows an example to illustrate our method.

Concretely, given a dialogue-level dependency tree, we sample a proportion of words in text, which are expected to be replaced by the LLM with a well-defined prompt. The updated text might be ill-formed. To alleviate this, we propose a straightforward and effective approach. It involves automatically verifying the alignment of punctuation marks such as commas, periods, and question marks in the rewritten text to ensure their positions match the original text. Additionally, the sequence of words separated by punctuation marks is checked to ensure that the word counts are equal. The entire rewriting process continues until these conditions are met. In this manner, the

entire dependency structure remains unchanged, thereby achieving a direct mapping of dependencies. The specific prompt definition is as follows:

- `CoT`: "1. 找出句子中的谓词。 2. 以谓词为中心，对句子进行逐词改写。 (1. Identify the predicates of given text. 2. Centered on the predicate, rewrite the given text word-by-word.)." In this case, we focus more on predicate-centered words as they are usually the core part of a text.
- `Constraint`: "4. 不允许改变词的顺序。 5. 输出格式：谓词'{1}', 文本'{2}'。 (4. The order of that change of words is not allowed. 5. The output format: Predicates '{1}', Text '{2}'.)" The former ensures the LLM does not disrupt the original word order and therefore maintains the syntactic structure. The latter prompts the LLM to generate a formatted output. These two constraints allow the dependencies of the original samples to be directly transferred to the rewritten samples.
- `One-Shot Example`: "输入：文本'{1}' /n 输出：谓词'{1}', 文本'{2}'。 ('Input: Text '{1}' /n Output: Predicates '{1}', Text '{2}'.)" The example takes the source text as input, and outputs the predicate words as well as the target text after being rewritten.

4.2 Syntax-Level Transformation

The word-level transformation maintains the unchanged syntactic and discourse structures, resulting in limited alterations. Although this alteration through simple word replacement can achieve a satisfactory quality of labels, it cannot yield substantial diversity. The low diversity limits the extra supervised signals. To increase diversity and, in turn, enhance performance more effectively, we implement alterations at a higher level, involving changes in syntax dependencies. Figure 2c illustrates an overview of this method.

The detailed prompt specific to this strategy is defined as follows:

- `CoT`: "1. 对于给定的文本及其篇章关系，请解释当前篇章关系。 2. 对于指定的核心，根据当前篇章关系，列举出有意义的依附示例。 3. 对于指定的依附，根据当前篇章关系，列举出有意义的核心示例。 4. 根据第二步和第三步中的依存组合示例，对输入文本进行重写，并按照指定格式输出。 (1. For a given text and its discourse relationship, please explain the discourse relationship. 2. For the specified nucleus, provide meaningful attachment examples based on the current discourse relationship. 3. For the specified attachment, provide meaningful nucleus examples based on the current discourse relationship. 4. Rewrite the input text based on the dependency combination examples from steps 2 and 3 and output it according to the specified format.)"
- `Constraint`: "4. 尽可能改变句法结构，但严格保留语篇结构。 5. 输出格式：文本'{1}'，核心'{2}'，依附'{3}'，推理步骤'{4}'。 (4. Alter the syntactic structure as much as possible, but strictly preserve the discourse structure. 5. Output format: Text '{1}', Nucleus '{2}', Attachment '{3}', Reasoning Step '{4}'.)" By the former instructions, the LLM has greater freedom in generating samples, while being constrained not to alter the

discourse structure. The latter requires the LLM to output in a fixed format for easy extraction.

- `One-Shot Example`: "输入:文本'{1}'，核心'{2}'，依附'{3}'，当前篇章关系'{4}'。 /n 输出: 文本'{1}'，核心'{2}'，依附'{3}'，推理步骤：'{4}'。 (Input: Text '{1}', Nucleus '{2}', Attachment '{3}', Current Discourse Relationship '{4}'. /n Output: Text '{1}', Nucleus '{2}', Attachment '{3}', Reasoning Steps '{4}'.) "

After generation, the baseline parser is used to assign syntactic dependencies to the altered text, ignoring discourse dependency predictions. Note that our data augmentation mainly aims at discourse-level dependencies, because the inner-EDU dependency parsing is already acceptable due to various available syntactic/semantic dependency treebanks. Thus, this transformation is executed from the perspective of keeping discourse dependencies unchanged, enriching the same discourse structure with abundant syntax contexts.

4.3 Discourse-Level Transformation

The syntax-level transformation changes the syntactic structure, resulting in a broader scope of sample variations. Nevertheless, the rigidity of discourse semantics can limit the model's ability to generalize outside the boundaries of the original discourse structure. Hence, we further propose a discourse-level transformation mechanism, as shown in Figure 2d.

The prompt of discourse-level augmentation is defined as follows:

- `CoT`: "1. 对于给定的文本及其篇章关系，请解释当前篇章关系和目标篇章关系。 2. 对于指定的核心，根据新的篇章关系，列举出有意义的依附示例。 3. 对于指定的依附，根据新的篇章关系，列举出有意义的核心示例。 4. 根据第二步和第三步中的依存组合示例，对输入文本进行重写，并按照指定格式输出。 (1. For a given text and its discourse relationship, please explain the current discourse relationship and the target discourse relationship. 2. For the specified nucleus, provide meaningful attachment examples based on the new discourse relationship. 3. For the specified attachment, provide meaningful nucleus examples based on the new discourse relationship. 4. Rewrite the input text based on the dependency combination examples from steps two and three and output it according to the specified format.)." We randomly choose a discourse relation for the LLM to explain and generate text based on it. Through the above chained reasoning, the LLM can generate and combine EDUs that fit the target discourse relationship.
- `Constraint`: "4. 输出格式：文本'{1}'，核心'{2}'，依附'{3}'，推理步骤'{4}'。 (4. The output format: Text '{1}', Nucleus '{2}', Attachment '{3}', Reasoning Steps '{4}'.)" This format constraint allows the LLM output to position EDUs and their relationships by fixed-form natural language, facilitating the alignment and completion of dependencies.
- `One-Shot Example`: "输入:文本'{1}'，核心'{2}'，依附'{3}'，当前篇章关系'{4}'，目标篇章关系'{5}'。 /n 输出: 文本'{1}'，核心'{2}'，依附'{3}'，推理步骤：'{4}'。 /n (Input: Text '{1}', Nucleus '{2}',

Attachment '{3}', Current Discourse Relationship '{4}', Target Discourse Relationship '{5}'. /n Output: Text '{1}', Nucleus '{2}', Attachment '{3}', Reasoning Steps '{4}'.) "

To assist the LLM in interpreting each relationship, we establish a well-defined description table that contains discourse relations and embed it into the prompt before the constraint part. To auto-annotate the generated text, we exploit the same method as Section 4.2 by the baseline parser to perform inner-EDU dependency parsing, while the inter-EDU dependencies are built upon the root words of EDUs with the labels specified by our prompting texts.

4.4 A Viewpoint from Self-Training

In this work, we exploit LLM to generate new dialogues by heuristic prompting strategies, where the inner-EDU and inter-EDU dependencies of these dialogues can be easily inferred from their source training instances. There is an interesting question regarding how this method relates to previously representative data augmentation approaches. Essentially, our method is a form of self-training with carefully chosen unlabeled examples. Many previous data augmentation studies assume that the newly added unlabeled instances can be easily labeled by heuristic strategies, e.g., rules (Sahin and Steedman 2018) and interpolation (Zhang et al. 2018). In contrast, our approach could be categorized as model-based as mentioned in Feng et al. (2021), and self-training is responsible for most of the annotations.

As compared to previous works of self-training for dependency parsing (Yu, Elkaref, and Bohnet 2015; Rotman and Reichart 2019; Guo et al. 2022), our approach is unique in two ways. First, part of the augmented dependencies (i.e., the inter-EDU dependencies) are not annotated by the basic parser. As our heuristic annotations produce better inter-EDU dependencies than the basic parser, it is reasonable to expect that our approach will be more effective, and our preliminary findings support this expectation. Second, we use LLMs to generate unlabeled instances instead of heuristic selection from a large-scale pool that was widely used before. Alternatively, we can use other text generators. As of yet, LLM is the best fit for this situation. Furthermore, with LLMs, we can produce high-quality dialogues with specifications that can be parsed most accurately, which is difficult with other tools.

There is another question that arises: Why not use LLM to perform dialogue-level parsing directly following a standard strategy of self-training? Currently, we find that LLMs are not well suited to dependency parsing at this point. Even using direct supervised fine-tuning based on an open-source Llama2-7B, the plain syntactic parsing (inner-EDU parsing) performs much worse than our baseline (the gap is greater than 4% in unlabeled attachment score [UAS]). As a result, we believe that a long-term investigation is needed in order to explore decoder-style LLMs for dependency parsing. Due to the indirect exploitation of syntax and discourse properties, our method can be considered a special case of distilling soft knowledge from LLMs.

5. Experiments

5.1 Settings

5.1.1 Dataset. We use the publicly available corpus released by Jiang et al. (2023), which is the only benchmark dataset for Chinese dialogue-level dependency parsing. The

Table 1
Data statistics. "#" and "avg. #" indicate "number of" and "average number of," respectively.

	# dialogue	avg. # turns	avg. # words	# inner-EDU	# inter-EDU
Train	50	23	194	9,129	1,671
Test	800	25	212	159,803	29,200

dataset unites syntactic dependencies and discourse dependencies as a whole over dialogue texts. The syntactic dependency structure is sourced from Jiang et al. (2018), and the dependency-based discourse structure is reorganized according to the characteristics of dialogue and previous RST-based discourse parsing (Li et al. 2014; Carlson, Marcu, and Okurovsky 2001). Table 1 shows the statistics of this dataset.

5.1.2 Evaluation. We assess the performance using the UAS and LAS ignoring the punctuation words, following the standard evaluation of dependency parsing. To provide a detailed analysis, we report the scores of inner-EDU and inter-EDU dependencies separately. The inter-EDU performance is calculated based on the concrete dependency arcs over words, not EDUs, which means that the correctness of EDU heads is a requirement for the correctness of inter-EDU dependencies. In situations where a development set is not available due to limited resources, we choose the last training checkpoint for evaluation purposes. All experiments are conducted on a single RTX 2080 Ti GPU.

5.1.3 Hyperparameters. For the baseline parser, we utilize the base scale discriminator of the Chinese version ELECTRA (Clark et al. 2019; Cui et al. 2020) as the PLM.[2] The hidden sizes of the subsequent neural modules are all 200, and the dropout ratio is set to 0.2. The AdamW (Loshchilov and Hutter 2019) optimizer is used for objective optimization, and the weight decay is 0.01. We use the linear warmup for the first 10% training steps, setting the initial learning rate of the PLM to 2e-5 and of the subsequent modules to 1e-4. To alleviate gradient explosion, we apply the gradient clipping mechanism by a maximum value of 2.0. The training batch size is set to 32 for both the syntactic treebank and pseudo-labeled dialogue, whereas it is set to 1 for the altered data based on the LLM. The number of epochs for training is set at 10.

For data augmentation, we utilize GPT-3.5-Turbo-0613[3] as the main LLM for its impressive performance in various NLP tasks. We also examine two open-source LLMs, namely, Llama2-7B (Touvron et al. 2023b) and Qwen-7B (Bai et al. 2023). Particularly, Qwen-7B has been optimized much for the Chinese language. We use the default temperature setting and empirically set top_p to 0.5, ensuring the stability of the generated output from the LLM while retaining diversity. For each type of transformation, we use LLM to generate three different augmentations based on a given instance in the training dataset.

5.1.4 One-Shot Example. The one-shot examples of word-level, syntax-level, and discourse-level transformations are designed according to the prompting descriptions

2 https://huggingface.co/hfl/chinese-electra-180g-base-discriminator.
3 https://platform.openai.com/docs/api-reference/chat.

Table 2
Concrete one-shot examples, where wrd, syn, and dis indicate word-level, syntax-level, and discourse-level transformations, respectively.

wrd	输入: 文本“这款笔记本我要退了，明天买其它样式。” /n 输出: 谓词“退、买”， 文本“那件衣服我要买了，今天下订单。” Input “I would like to refund this laptop, and I will buy other things tomorrow.” /n Output: Predicates “refund, buy,” Text “I am going to buy that dress, and I will order this item today.”
syn	输入: 文本“这款笔记本我要退了，质量实在太差了。”，核心“这款笔记本我要退了”，依附“质量实在太差了”，篇章关系“原因” /n 输出: 文本“我准备购买其它笔记本，因为它无法正常使用。”, 核心“我准备购买其它笔记本，”，依附“因为它无法正常使用。”。推理步骤“1. 篇章关系“原因”是指依附为核心的原因。2. “这款笔记本我要退了”，对于篇章关系“原因”，依附的例子可以为“笔记本不符合我购买的需求”，“这件商品质量不好”，“它无法正常使用”。3. “质量实在太差了。”，对于篇章关系“原因”，核心的例子可以为“我无法正常使用该笔记本”，“我准备购买其它笔记本”，“我要退款”。4. 综上依存组合，根据“ 因为笔记本无法正常使用。”，核心“我准备购买其它笔记本”和“因为它无法正常使用”，生成了新的文本。 Input: Text “I want to refund this laptop, because the quality is really bad.”, Nucleus “I want to refund this laptop,”, Attachment “because the quality is really bad.”, Discourse Relationship “Cause” /n Output: Text “Because the laptop cannot function properly, I plan to purchase other products.”, Nucleus “I plan to purchase other products.”, Attachment “Because the laptop cannot function properly,”, Reasoning Steps “1. The discourse relation discourse relation “Cause” refers to reasons with attachment as the nucleus. 2. For the nucleus “I want to refund this laptop,”, in the context of discourse relation “Cause”, examples of attachments could be “the laptop doesn’t meet my purchasing needs,” “this quality of laptop is bad,” “the laptop cannot function properly”. 3. For the attachment “because the quality is really bad.”, examples of nucleus elements related to the discourse relation “Cause” could be “I won’t be able to use the product properly”, “I’m considering buying another laptop”, “I want a refund”. 4. Based on the dependency combinations mentioned above, a new text has been generated using “Because the laptop cannot function properly,” and “I’m considering buying another laptop”.
dis	输入: 文本“如果这款笔记本明天还没有到货，我就要退了”，核心“我就要退了”，依附“如果这款笔记本明天还没有到货”，当前篇章关系“条件”，目标篇章关系“因果” /n 输出: 文本“因为这款笔记本没有到货，我要退款”，核心“我要退了”，依附“因为这件商品没到货”， 推理步骤：1. 篇章关系“ 条件”是指依附为核心的前提，而篇章关系“原因”是指依附为核心的原因。2. “我就要退了”，对于篇章关系“原因”，依附的例子可以为“笔记本不符合我购买的需求”，“因为这款笔记本没有到货”，“ 笔记本无法正常使用”。3. “如果这款笔记本明天还没有到货”，对于篇章关系“原因”核心的例子可以为“我明天无法使用该商品”，“我准备购买其它商品”，“我要退款”。4. 综上依存组合，根据“因为这款笔记本没有到货”和“我要退款”，生成了新的文本。 Input: Text “If this laptop doesn’t arrive tomorrow, I am going to refund this merchandise.”, Nucleus “I am going to refund this merchandise”, Attachment “If this laptop doesn’t arrive tomorrow”, Current Discourse Relationship “Condition”, Target Discourse Relationship “Cause” /n Output: Text “Because this laptop hasn’t arrived, I want a refund.”, Nucleus “I want a refund.”, Attachment “Because this laptop didn’t arrive,”, Reasoning Steps “1. The discourse relation “Condition” refer to premises with attachment as the nucleus, while discourse relation “Cause” refers to reasons with attachment as the nucleus. 2. For the nucleus “I am going to refund this merchandise”, in the context of discourse relation “Cause”, examples of attachments could be “the laptop doesn’t meet my purchasing needs,” “Because this laptop hasn’t arrived,” “Because the laptop can’t be used properly.” 3. For the attachment “If this laptop doesn’t arrive tomorrow,” examples of nucleus elements related to the discourse relation “Cause” could be “I won’t be able to use the laptop tomorrow”, “I’m considering buying another laptop”, “I want a refund”. 4. Based on the dependency combinations mentioned above, a new text has been generated using “Because this laptop hasn’t arrived” and “I want a refund” as the current inputs.”.

in `CoT` and `Constraint`. Table 2 presents specific details. The one-shot example of word-level transformation consists of identifying the predicates first, and then rewriting the sentence word-by-word. In syntax-level transformation, the one-shot example first outputs the rewritten text along with the discourse nucleus and attachment in the text, followed by a CoT that details the reasoning steps. The CoT is achieved by first describing the discourse relationship, then generating several rewritten nuclei and attachments

in line with the relationship, and finally combining one specific pair of the produced candidates. The one-shot example of discourse-level transformation is similar to that at the syntax level. The key difference lies in the goal of discourse-level transformation, which aims to change discourse relationships rather than preserve them.

5.2 Results

We consider two different settings during evaluation: (1) the zero-shot setting consistent with Jiang et al. (2023), where a set of rule-based silver instances is used as the initial training dataset, and (2) the few-shot setting, where the 50 human-annotated instances together with the silver corpus are used for training. For each setting, we evaluate the baseline method, each augmentation strategy alone, pairwise combinations, and the full combination of all three data augmentations. In this way, we can examine the potential of all our data augmentations comprehensively. Table 3 shows the main results. We conduct significant tests between the baseline and our methods using pairwise t-test.

Table 3
Zero-shot and few-shot results.

Training Data	Zero-shot				Few-shot			
	Inner-EDU		Inter-EDU		Inner-EDU		Inter-EDU	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Jiang et al. (2023)	88.22	84.34	66.48	50.78	91.74	88.20	71.09	55.73
baseline	88.20	84.40	66.41	50.85	91.66	89.12	71.59	56.32
GPT-3.5-Turbo-0613								
+ wrd	88.12	84.23	67.73	52.14	**92.37**	90.01	73.06	58.50
+ syn	88.09	**84.36**	68.19	52.81	92.13	89.94	73.22	59.33
+ dis	**88.27**	84.31	**68.57**	**53.41**	92.35	**90.11**	**73.57**	**59.68**
+ wrd & syn	88.32	84.43	68.44	53.39	92.38	90.16	73.52	59.47
+ wrd & dis	88.24	84.12	68.64	53.52	92.19	90.04	73.84	59.81
+ syn & dis	88.08	84.17	68.77	53.62	92.23	90.18	**73.88**	59.94
+ wrd & syn & dis	**88.33**	**84.51**	**68.82**	**53.89**	**92.46**	**90.35**	73.81	**60.17**
Llama2-7B								
+ wrd	87.79	83.99	66.83	51.28	**91.91**	89.73	72.33	57.63
+ syn	87.75	84.00	67.17	51.67	91.65	89.51	72.31	58.28
+ dis	**88.01**	**84.02**	**67.75**	**52.27**	91.90	**89.85**	**72.76**	**58.45**
+ wrd & syn	88.02	84.13	67.52	52.22	91.87	89.81	72.56	58.38
+ wrd & dis	**88.05**	83.89	67.78	52.46	91.82	89.63	**73.13**	58.75
+ syn & dis	87.85	83.90	67.80	52.59	91.76	**89.91**	72.92	58.79
+ wrd & syn & dis	88.01	**84.32**	**68.15**	**52.69**	**91.97**	89.89	72.95	**59.01**
Qwen-7B								
+ wrd	87.87	84.12	67.22	51.43	**92.03**	89.88	72.68	57.94
+ syn	87.88	**84.14**	67.63	52.03	91.94	89.69	72.80	58.46
+ dis	**88.16**	84.09	**68.11**	**52.51**	92.01	**89.97**	**73.19**	**58.85**
+ wrd & syn	88.15	84.21	67.92	52.64	91.84	89.97	73.05	58.74
+ wrd & dis	88.11	84.00	68.23	52.86	91.87	89.76	73.47	59.05
+ syn & dis	87.98	84.02	68.18	53.02	**92.07**	**89.99**	73.42	59.14
+ wrd & syn & dis	**88.18**	**84.31**	**68.41**	**53.12**	91.96	89.85	**73.52**	**59.31**

First, we examine the results of the zero-shot setting as a whole. The baseline method achieves 88.20 UAS and 84.40 LAS on inner-EDU dependencies, but only 66.41 UAS and 50.85 LAS for the inter-EDU dependencies, indicating that the inter-EDU dependency parsing is still underperforming. With our two-step parsing of inner-EDU and inter-EDU dependencies, which considers the hierarchical structure of dialogue-level dependency parsing, the baseline achieves better performance than Jiang et al. (2023). Through our word-level, syntax-level, and discourse-level data augmentations, both the inner-and inter-EDU performance can be improved, and the inter-EDU performance can be improved even more. As shown, the final model has $84.51 - 84.40 = 0.11$ improvement in inner-EDU dependencies, and $53.89 - 50.85 = 3.04$ ($p < 0.001$) improvements in inter-EDU dependencies. The marginal gain on inner-EDU dependencies can be attributed to the sufficiently large scale of the dependencies during training provided by a syntactic treebank.

Furthermore, we examine the performance of word-level, syntax-level, and discourse-level data augmentation separately, as well as their pairwise combinations. The overall tendency is that discourse-level $>$ syntax-level $>$ word-level in terms of performance. Among the single augmentation strategies, the discourse-level method shows the best performance, and the word-level approach is the worst. Among the pairwise combinations, the combination of syntax- and discourse-level strategies yields the highest LAS, whereas the word- and syntax-level combination performs the poorest. The possible reason for this observation might be that high-level substitutions can actually cover the low-level alternations to some degree. Our results also show that the three methods can be supplementary to one another because the addition of another augmentation strategy can always bring improved performance. The reason might be that the low-level data augmentation can obtain higher-quality dependency trees because of the relatively smaller variations.

Third, we shift our focus to the few-shot results with an extra 50 human-annotated dialogue-level dependency trees. As shown, the baseline performance of both inner-EDU and inter-EDU dependencies has been greatly boosted. There are two aspects that contribute to the significant improvements: in-domain dialogue data for inner-EDU syntactic parsing, and supervised data for inter-EDU discourse parsing. In addition, we observe completely consistent results in the few-shot setting compared to the zero-shot. The discourse-level data augmentation can bring the highest gains, whereas the word-level one is the lowest. The pairwise combination is always better than single augmentation alone. The final method, which combines all three strategies, obtains the best performance, leading to improvements of $90.35 - 89.12 = 1.23$ ($p < 0.001$) in inner-EDU LAS and $60.17 - 56.32 = 3.85$ ($p < 0.001$) in inter-EDU LAS. Interestingly, we find that larger improvements can be achieved in this setting by our data augmentation, despite a stronger baseline. The reason might be that by adding higher-quality source instances, the produced pseudo instances after data augmentation are less noisy.

Finally, we compare the zero-shot and few-shot performance across different LLMs. The above results are based on the closed-source GPT-3.5-Turbo, and here we further verify our method based on two open-source LLMs: (1) Llama2-7B and (2) Qwen-7B. As shown, GPT-3.5-Turbo exhibits the highest performance on our task among the three LLMs. Compared with Llama2-7B, the difference is significant in the inter-EDU results. We can see that the performance gaps are $53.89 - 52.69 = 1.20$ and $60.17 - 59.01 = 1.16$ for the zero-shot and few-shot settings, respectively. In addition, Qwen-7B performs better than Llama2-7B. The reason might be that Qwen-7B involves Chinese-oriented optimization during pretraining.

6. Analysis

6.1 Prompt Design

6.1.1 Characterization. The integration of a specialist role position enables the LLM to comprehend and adapt to a new task more accurately. Given that the generation of coherent text and rational dependency structures necessitates expertise in language, we position the LLM as a natural language specialist, as illustrated in Figure 2.

To probe the influence of prompt design on the LLM's generation performance, we select a sample as a case study and compare the generation results without and with characterization. As depicted in Figure 3, characterization empowers the LLM to produce samples that better match our requirements. In the syntax-level strategy, we aim for the LLM to maintain the original syntactic and discourse structures. Without characterization, the LLM might lack necessary prior knowledge, causing difficulties in understanding syntactic and discourse structures. By meticulously defining roles for the LLM, it can potentially assimilate NLP field-specific prior knowledge effectively, thereby circumventing this issue.

6.1.2 CoT. A CoT encompasses a series of intermediate logical steps, notably enhancing the capability of LLMs to execute intricate reasoning tasks (Wei et al. 2022). Following this work, we guide the LLM to produce logical inference results progressively, ultimately obtaining outcomes that fulfill the generation specifications. To assess the influence of CoT on LLM-based data augmentation, we select the same sample as previously discussed as a case study. We then observe the effects on the discourse-level data augmentation with and without the utilization of CoT. As illustrated in Figure 4, CoT effectively guides the generation of samples that meet our criteria. Our discourse-level transformation aims to alter the discourse structure of the original sample, whereas this objective is not fulfilled in the absence of CoT. One plausible explanation for this

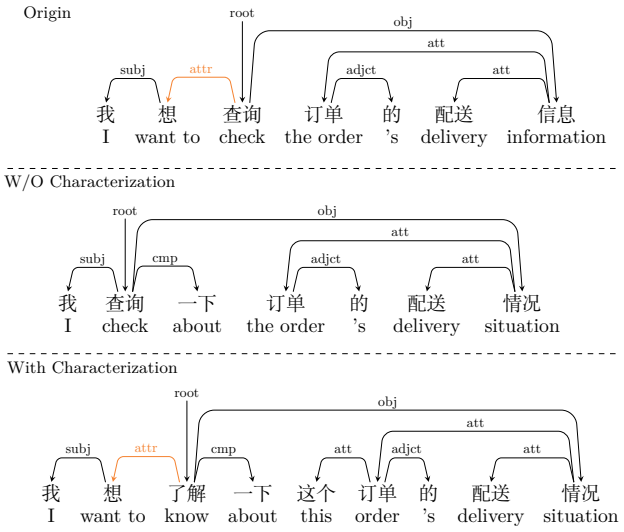


Figure 3
A case to compare the syntax-level transformation with and without characterization.

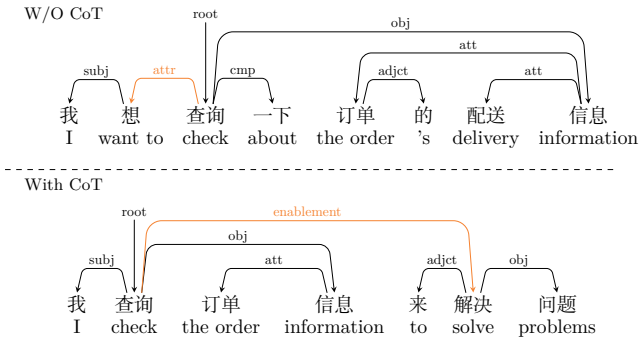


Figure 4
A case to compare the discourse-level transformation with and without CoT. Clearly, the discourse structure is not altered without CoT, opposing this strategy.

can be that without step-by-step inference for complex tasks, the LLM might struggle to accurately comprehend the task and generate logical outcomes.

6.1.3 Constraint. Despite the impressive language comprehension capability of LLMs, it is still a major challenge to generate stable and reliable texts consistently. Fortunately, leveraging natural language to instill constraints into the LLM's generation process has shown to be effective. This method capitalizes on the LLM's language understanding aptitude by conveying constraint information to the LLM in the form of natural language directives, thereby enabling the LLM to comprehend and adhere to these constraints. Following the line of these works, we delineate a set of constraints to steer the LLM in its generation process. Taking the syntax-level approach as an example, the specific constraints are shown in Figure 2c.

When constraints are not provided, the LLM may produce unpredictable results. We illustrate the effects of these constraints on the LLM-generated results by using several key constraints. First, it is essential to specify that the LLM should output text in Chinese; without this instruction, it may default to English responses. Second, word segmentation, a crucial step for Chinese dependency analysis, must be explicitly required; otherwise, the LLM will output unsegmented results, leading to misalignment in dependency relations. Third, the LLM must be prompted to stop generating after providing a response; otherwise, it may continue generating unrelated content. Fourth, the LLM must be explicitly directed not to reply to or extend the given content; without this directive, it may respond to some interrogative sentences, thereby invalidating the generated results. Thus, the LLM needs to identify EDUs and clearly delineate their boundaries.

6.1.4 One-Shot Instruction. Owing to its vast scale, fine-tuning the LLM poses a challenge, rendering it difficult to supply supervision signals for the LLM's adaptation to downstream tasks. Fortunately, the introduction of supervision signals into the LLM via in-context learning has been demonstrated to be straightforward and effective (Brown et al. 2020). Using the method, the LLM can produce reliable and desired text by mimicking the given examples. We manually select a sample at random from the training set and meticulously craft a generation example in accordance with the generation strategy, subsequently appending it to the prompt. In addition, we provide a reason for the

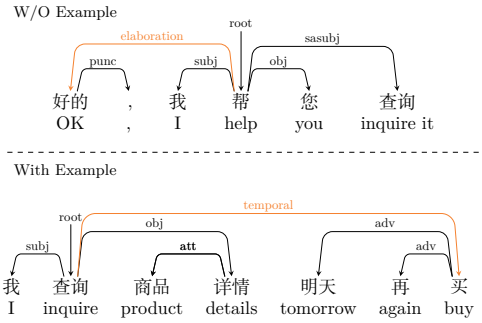


Figure 5
A case to compare our discourse-level transformation with and without one-shot example.

generation to support CoT. Figure 2b provides a demonstration of the prompt utilized in the word-level transformation. Here, an initial text sample is supplied, followed by the generation of sample outcomes, accompanied by the elucidation of the reasoning behind these outcomes.

As depicted in Figure 5, we observe noticeable disparities in generation with and without the provision of an example. In the absence of example guidance, an erroneous "elaboration" direction is generated, which can potentially be attributed to the LLM's inability to comprehend the dependency structure and necessary structural modifications. When provided with an example, the LLM can mimic the existing sample, subsequently generating stable and reliable structures.

6.2 Different Input Granularity

In Section 4, we mention that our data augmentation methods can take either EDUs or complete utterances as inputs. Here, we compare the performance of the two to demonstrate the differences, as shown in Table 4. We observe that using utterances as inputs generally achieves higher performance. One possible reason is that this approach provides a larger receptive field for the LLM, allowing it to balance fluency and diversity. The highest performance is still achieved using the discourse-level transformation

Table 4
The impact of different input granularity. "w/i" denotes "within."

Augmented Data	Zero-shot				Few-shot			
	Inner-EDU		Inter-EDU		Inner-EDU		Inter-EDU	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
wrd w/i EDUs	88.08	84.10	67.34	51.86	92.18	89.95	72.87	58.22
wrd w/i utterances	88.12	84.23	67.73	52.14	92.37	90.01	73.06	58.50
syn w/i EDUs	88.16	84.21	67.91	52.42	92.24	90.01	73.33	58.89
syn w/i utterances	88.09	84.36	68.19	52.81	92.13	89.94	73.22	59.33
dis w/i EDUs	88.21	84.33	68.18	53.07	92.21	89.97	73.43	59.27
dis w/i utterances	88.27	84.31	68.57	53.41	92.35	90.11	73.57	59.68

method, consistent with previous experiments. Both methods can lead to significant performance improvements, underscoring the superiority of the methods we propose. Furthermore, in our preliminary experiments, we observed that when LLM rewriting with dialogue-level input fails to follow the provided instructions, the rewritten samples cannot be assigned or filled with labels. One possible reason is that the difficulty in accurately rewriting arises when the input text becomes longer.

6.3 Instance Diversity

We calculate the overlap between the original dataset ($\mathcal{D}^f$) and the augmented dataset ($\mathcal{D}^e$) by averaging the Rouge-1 score for each pair of the original and generated instances. Intuitively, samples exhibiting lower overlap with the original ones are indicative of greater diversity. As shown in Figure 6, the diversity of data samples augmented through word-level, syntax-level, and discourse-level methods exhibits an increasing trend, corroborating our intuitive expectations. Simultaneously, the overlapping of rewritten or generated samples at the utterance level is frequently lower than that at the EDU level. This result suggests that the reconstruction of the entire utterance can introduce a higher degree of diversity. By correlating with the experimental results in few-shot and zero-shot settings, we observe a consistency between the increase in diversity and the improvement in performance. Thus, the diversity of the augmented samples contributes positively to the efficacy of the parser.

6.4 Influence of Data Mixture

In Section 5.2, our best results are obtained by a combination of data generated by word-level, syntax-level, and discourse-level transformations. We set the mixture ratio as equally distributed, namely, 1:1:1. This might not be the optimal ratio. Here we conduct experiments to study the influence of the mixing ratio of augmented data on parsing performance. For simplicity, we set the data ratio as three cases: 0%, 50%, and 100%, and report the inter-EDU LAS. The results are presented in Figure 7 in a three-dimensional graph with larger and darker bubbles representing the better-performing mixtures. It can be observed that mixing all three methods results in a stronger performance improvement than mixing only two methods or using a single method. Additionally, we

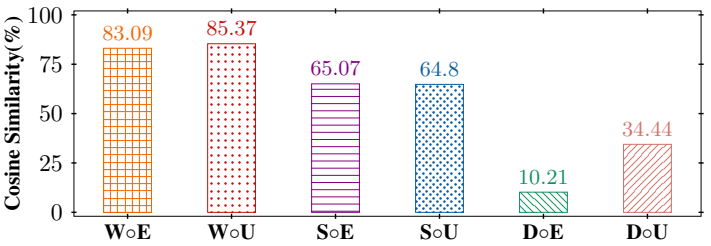


Figure 6
Diversity of augmented samples, where the Rouge-1 score is used to measure the diversity between the original samples and the augmented samples: the smaller the score, the higher the diversity. "W," "S," "D" denote the "word-level," "syntax-level," and "discourse-level," respectively. "E" and "U" indicates within the EDUs or utterances.

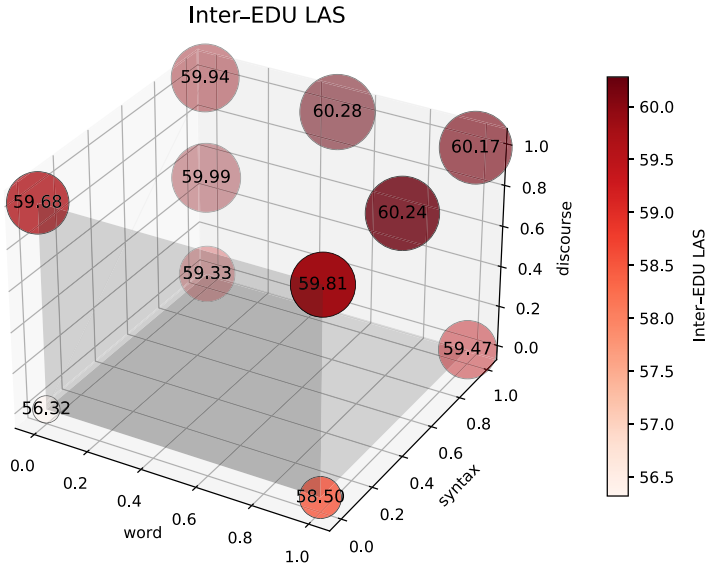


Figure 7
The performance with respect to data mixture ratio, where the values of inter-EDU LAS are reported.

find that a 1:1:1 ratio indeed is not optimal. A better performance can be achieved when one part is set to 0.5. When the ratio of word-level, syntax-level, and discourse-level augmentations is 1:0.5:1, the performance is the best among our investigated cases. This is possibly due to the word-level transformation adding the most basic and accurate data, whereas the discourse-level one is the most diverse, and the syntax-level approach is in between. When all three are equally mixed, this may introduce significant overlap, which can be alleviated by simply reducing the data in the middle part, while balancing accuracy and diversity.

6.5 Influence of Augmentation Ratio

According to the main experimental setup, each training instance is augmented into nine new ones by our three types of transformation using LLMs. Each type of transformation offers three different augmentations, defined by the augmentation ratio. Here, we examine how this ratio affects the final performance.

Figure 8 shows the experimental results based on the setting of few-shot learning, where the inner-EDU as well as the inter-EDU LASs are depicted separately. The three single-augmentation strategies and their various combinations are all examined. When the ratio is 3, almost all data augmentation methods achieve their peak, while minimal gains are obtained as the ratio is increased further. The inner-EDU LAS shows very marginal improvements when the ratio goes over 1. Therefore, the observation indicates that there is an upper bound to the capabilities of our data augmentation. With a three-time augmentation, we can make the most of our approach. Furthermore, our data augmentation does not degrade significantly as the augmentation ratio increases after the peak is reached, indicating the robustness of our method.

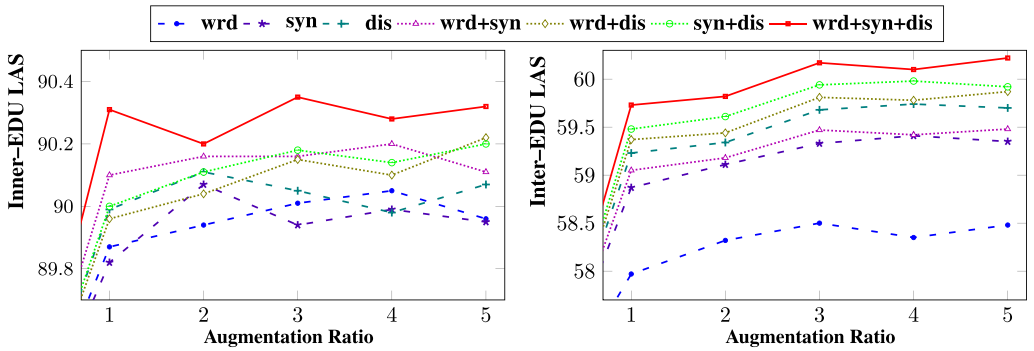


Figure 8
Performance in terms of different data augmentation ratios. The dashed lines denote single-strategy data augmentation, the dotted lines demonstrate pairwise combinations, and the solid lines indicate the combination of our three-strategy together.

7. Related Work

7.1 Dependency Parsing

Dependency parsing has received widespread academic attention (Kübler, McDonald, and Nivre 2009). Up to the present, various Chinese dependency paradigms and their associated treebanks have been established (Xue et al. 2005; Che, Li, and Liu 2012; McDonald et al. 2013; Qiu et al. 2014). The majority of these studies are devoted to sentence-level dependency parsing, whereas document-level parsing is noticeably underrepresented in the literature. A dependency parsing paradigm has been proposed for discourse parsing (Li et al. 2014). This paradigm, characterized by an EDU-centric pattern, overlooks parsing operations within the EDUs themselves. Jiang et al. (2023) have undertaken preliminary studies on dialogue-level dependency parsing in Chinese, taking into account both inner- and inter-EDU considerations. However, the proposed methodology is not fully integrated, or end-to-end, and exhibits a shortfall in comprehensive data exploitation investigations.

7.2 Data Augmentation

Data augmentation has been a topic of focus from an early stage (Scudder 1965; Tanner and Wong 1987; Van Dyk and Meng 2001; Feng et al. 2021). Utilizing existing treebanks to train models, and assigning pseudo-labels to unlabeled data, is a common approach during the annotation stage of dependency parsing (Jiang et al. 2018; Li et al. 2019). Additionally, pseudo-labeled data within the target domain can provide weak supervision signals, effectively enhancing the generalization ability of models (Scudder 1965; Lee et al. 2013; Guo et al. 2022; Li et al. 2023). On the basis of these studies, our approach leverages both a syntactic treebank (Jiang et al. 2018) and a pseudo-labeled dialogue dataset, aiming to improve model performance within a few-shot learning environment. In the present era, LLMs have demonstrated profound comprehension and generative abilities (OpenAI 2023). Given this context, it is natural to utilize LLMs for the generation of pseudo-samples that are both natural and logically consistent, thereby facilitating the training of more proficient models (Whitehouse, Choudhury, and Aji 2023; Dai et al. 2023).

7.3 Large Language Models

As of now, the field of NLP has experienced the emergence and growing prominence of LLMs, such as PaLM (Chowdhery et al. 2023), ChatGPT (OpenAI 2023), and GPT-4 (Achiam et al. 2023). After instruction tuning, LLMs can accurately comprehend user instructions and generate text in accordance with user preferences (Ouyang et al. 2022; Wang et al. 2022; Peng et al. 2023). The recent breakthroughs achieved by GPTs (OpenAI 2023; Achiam et al. 2023) present significant opportunities to enhance the capabilities of open-source LLMs such as LLaMA (Touvron et al. 2023a), Standford Alpaca (Taori et al. 2023), and Vicuna (Vicuna 2023) via instruction-tuning methodologies. Based on the powerful instruction comprehension and text generation abilities of LLMs, several studies have attempted to use LLMs for data augmentation (Whitehouse, Choudhury, and Aji 2023; Dai et al. 2023). Nonetheless, current LLM-based data augmentation methods are primarily applied to text classification tasks, and would face issues related to untranslatable labels in structural analysis such as dependency parsing. Our approach can accurately map the dependency structure from the original text to the altered text, while also ensuring the diversity of the data.

8. Conclusion

In this study, we focused on dialogue-level dependency parsing in Chinese. To address the low-resource challenges posed by this task, we proposed using LLM assistance for data augmentation to provide more supervised signals. Considering the hierarchical structure of dialogue dependencies, we implemented data augmentation at different levels: from the lowest word-level, to the intermediate syntax-level, and then to the discourse-level. To meet the requirements of these strategies, we integrated multiple prompt design methods, including characterization, CoT, constraint, and a one-shot example, and meticulously designed prompts accordingly. Experimental results demonstrated that our approach effectively improves the performance of dialogue-level dependency parsing. We also provided in-depth analysis covering the impact of prompt design, the mixture of augmented data by different level of transformations, the augmentation ratio, and so forth.

The limitations of this study are primarily manifested in two ways. Although carefully designed prompt engineering is exploited for different levels of instance transformation, our approach still relies on manual prompt design, which could introduce subjectivity and potentially limit the scalability of our method. Moreover, the effectiveness of our method has only been demonstrated in the context of dialogue-level dependency parsing. It remains unclear whether it can be generalized across different levels and languages of dependency parsing, and further to broader NLP tasks. In the future, given the flexibility of our method, we intend to explore its application to a broader range of NLP tasks in diverse languages.

Acknowledgments

We sincerely thank the reviewers for their invaluable feedback, which significantly improved the quality of this work. This work is supported by the National Natural Science Foundation of China (NSFC) grant nos. 62336008 and 62176180.

References

Achiam, Josh, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*. 1–100.

Afantenos, Stergos, Éric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multi-party chat dialogues. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 928–937. https://doi.org/10.18653/v1/D15-1109

Asher, Nicholas, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: The STAC corpus. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2721–2727.

Bai, Jinze, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*, pages 1–59.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Carlson, Lynn, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10. https://doi.org/10.3115/1118078.1118083

Che, Wanxiang, Zhenghua Li, and Ting Liu. 2012. Chinese Dependency Treebank 1.0 LDC2012T05. Philadelphia: Linguistic Data Consortium.

Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2019. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, pages 1–18.

Cui, Yiming, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668. https://doi.org/10.18653/v1/2020.findings-emnlp.58

Dai, Haixing, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. AugGPT: Leveraging chatGPT for text data augmentation. *arXiv preprint arXiv:2302.13007*, pages 1–12.

Davidson, Sam, Dian Yu, and Zhou Yu. 2019. Dependency parsing for spoken dialog systems. In *Proceedings of the EMNLP-IJCNLP 2019*, pages 1513–1519. https://doi.org/10.18653/v1/D19-1162

Dozat, Timothy and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*, pages 1–8.

Feng, Steven Y., 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. https://doi.org/10.18653/v1/2021.findings-acl.84

Guo, Peiming, Shen Huang, Peijie Jiang, Yueheng Sun, Meishan Zhang, and Min Zhang. 2022. Curriculum-style fine-grained adaption for unsupervised cross-lingual dependency transfer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:322–332. https://doi.org/10.1109/TASLP.2022.3224302

Jiang, Gongyao, Shuang Liu, Meishan Zhang, and Min Zhang. 2023. A pilot study on dialogue-level dependency parsing for Chinese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9526–9541. https://doi.org/10.18653/v1/2023.findings-acl.607

Jiang, Xinzhou, Zhenghua Li, Bo Zhang, Min Zhang, Sheng Li, and Luo Si. 2018. Supervised treebank conversion: Data and approaches. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2706–2716. https://doi.org/10.18653/v1/P18-1252

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127. https://doi.org/10.1007/978-3-031-02131-2

Lee, Dong Hyun, et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, pages 896–901.

Li, Jianling, 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. https://doi.org/10.18653/v1/2023.emnlp-main.508

Li, Sujian, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35. https://doi.org/10.3115/v1/P14-1003

Li, Zhenghua, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2386–2395. https://doi.org/10.18653/v1/P19-1229

Liu, Pei, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195. https://doi.org/10.1109/CCNS50731.2020.00049

Loshchilov, Ilya and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*, pages 1–19. https://doi.org/10.3115/1075812.1075835

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop*, pages 114–119.

McDonald, Ryan, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97.

Nivre, Joakim. 2005. Dependency grammar and dependency parsing. *MSI report*, 5133(1959):1–32.

OpenAI. 2023. Introducing chatGPT. https://openai.com/blog/chatgpt

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744.

Peng, Baolin, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*, pages 1–12.

Qiu, Likun, Yue Zhang, Peng Jin, and Houfeng Wang. 2014. Multi-view Chinese treebanking. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 257–268.

Rotman, Guy and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713. https://doi.org/10.1162/tacl_a_00294

Sahin, Gözde Gül and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low resource languages. In *2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009. https://doi.org/10.18653/v1/D18-1545

Scudder, H. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371. https://doi.org/10.1109/TIT.1965.1053799

Shorten, Connor, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of Big Data*, 8:1–34. https://doi.org/10.1186/s40537-021-00492-0, PubMed: 34306963

Tanner, Martin A and Wing Hung Wong. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540. https://doi.org/10.1080/01621459.1987.10478458

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. pages 1–27.

Touvron, Hugo, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. LLAMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*. pages 1–77.

Van Dyk, David A. and Xiao-Li Meng. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50. https://doi.org/10.1198/10618600152418584

Vicuna. 2023. https://lmsys.org/blog/2023-03-30-vicuna/

Wang, Yizhong, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-Natural Instructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109. https://doi.org/10.18653/v1/2022.emnlp-main.340

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837.

Whitehouse, Chenxi, Monojit Choudhury, and Alham Fikri Aji. 2023. LLM-powered data augmentation for enhanced crosslingual performance. *arXiv preprint arXiv:2305.14288*. pages 1–16. https://doi.org/10.18653/v1/2023.emnlp-main.44

Xue, Naiwen, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11(2):207–238. https://doi.org/10.1017/S135132490400364X

Yu, Juntao, Mohab El-karef, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10. https://doi.org/10.18653/v1/W15-2201

Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.