Journal of Artificial Intelligence Research 81 (2024) 771-791 Submitted 11/2023; published 12/2024

Cross-domain Constituency Parsing by Leveraging Heterogeneous Data

Peiming Guo GUOPEIMING.GPM@GMAIL.COM
Meishan Zhang MASON.ZMS@GMAIL.COM
Institute of Computing and Intelligence,
Harbin Institute of Technology (Shenzhen), Shenzhen, China

Yulong Chen YULONGCHEN1010@GMAIL.COM
School of Engineering,
Westlake University, Hangzhou, China

Jianling Li JIANLINGL@TJU.EDU.CN
School of New Media and Communication,
Tianjin University, Tianjin, China

Min Zhang ZHANGMIN2021@HIT.EDU.CN
Institute of Computing and Intelligence,
Harbin Institute of Technology (Shenzhen), Shenzhen, China

Yue Zhang YUE.ZHANG@WIAS.ORG.CN
School of Engineering,
Westlake University, Hangzhou, China

Abstract

Knowledge transfer is investigated in various natural language processing tasks except cross-domain constituency parsing. In this paper, we leverage heterogeneous data to transfer cross-domain and cross-task knowledge to constituency parsing. Concretely, we first select language modeling, named entity recognition, CCG supertagging and dependency parsing as auxiliary tasks and collect the corpora of these tasks covering various domains as cross-domain and cross-task heterogeneous data. Second, we exploit three types of prefixes: shared, task and domain prefix, to merge cross-domain and cross-task data and decompose the general, task and domain representation in the pretrained language model. Third, we convert the data formats of multi-source heterogeneous datasets and loss objectives of the auxiliary tasks into a consistent formalization closer to constituency parsing. Finally, we jointly train the model to transfer task and domain knowledge to cross-domain constituency parsing. We verify the effectiveness of our proposed model on five target domains of MCTB. Experimental results show that our knowledge transfer model outperforms various baseline models, including conventional chart-based and transition-based parsers and the current large-scale language model for zero-shot and few-shot settings.

1. Introduction

Constituency parsing (CP) is a fundamental task in computational linguistics, which aims to build a hierarchical syntax tree for the given sentence. The current state-of-the-art is a chart-based parser (Stern, Andreas, & Klein, 2017; Kitaev & Klein, 2018; Tian, Song, Xia, & Zhang, 2020; Cui, Yang, & Zhang, 2022), which assigns scores to all the spans within a sentence and employs the CKY algorithm (Cocke, 1969; Kasami, 1966; Younger, 1967) to search for the optimal parse tree. Such a parser requires supervised training over manually labeled data (i.e., treebanks). However, treebank annotation can be extremely expensive and time-consuming, and only a few domains have large annotated treebanks. As a result, parsing performances over low-resource domains are still low, leaving constituent parsing as an unresolved problem.

Knowledge transfer is a key problem for cross-domain constituency parsing. In general, both task knowledge and domain knowledge can be transferred to improve constituency parsing. For task knowledge transfer, it has been shown that constituency parsing can be improved by named entity recognition (Finkel & Manning, 2009) and dependency parsing (Sun & Wan, 2013; Zhou & Zhao, 2019). However, they only focus on the in-domain setting. Existing work on domain knowledge transfer for constituency parsing (McClosky, Charniak, & Johnson, 2010; Fried, Kitaev, & Klein, 2019), on the other hand, does not consider making knowledge transfer from relevant tasks. Intuitively, for maximizing the utility of manually-labeled resources, a constituency parser should learn knowledge from both multi-domain constituency corpora and multi-domain corpora for related tasks. In addition, it should make use of knowledge from unlabeled data.

The above goal poses a significant challenge to modeling, since it requires information integration from heterogeneous data sources. To this end, it has been shown that a standard representation model can fail to achieve the most effective knowledge transfer over multiple loss sources (Søgaard & Goldberg, 2016; Bingel & Søgaard, 2017; Crawshaw, 2020). Existing work has considered shared-private structure (Liu, Johns, & Davison, 2019a; Wu, Zhang, Jin, Xue, & Wang, 2019), adversarial loss (Ganin & Lempitsky, 2015; Liu, Qiu, & Huang, 2017), feature transformation and selection (Zhang & Yang, 2022) for transfer learning. However, most existing methods consider only one dimension in knowledge transfer (i.e., either cross-task transfer or cross-domain transfer), and therefore cannot be directly applied to our setting. Large language models (LLMs) can be strong transfer learners, yet their performances on structured tasks are not as strong as encoder-only models (Qin, Zhang, Zhang, Chen, Yasunaga, & Yang, 2023; Li, Zhang, Guo, Zhang, & Zhang, 2023)[1]. We investigate a novel multi-prefix encoder-only representation model by adapting the prefix tuning method for text generation (Li & Liang, 2021) to our parsing problem. The basic idea is to augment the input token sequence with two types of prefix tokens, which allow a generic neural representation model to gain domain-specific and task-specific information respectively.

We choose the method of Kitaev and Klein (2018) as our baseline, which gives the current state-of-the-art results for constituency parsing. It makes use of a pre-trained BERT (Devlin, Chang, Lee, & Toutanova, 2019) model, which contains generic knowledge from masked language model pre-training. On top of this baseline, we add auxiliary output layers to the representation model, which allow knowledge from labeled data for related tasks over multiple domains to be injected into the representation model by specific loss functions. Since our goal is to optimize parsing performance, we do not treat each task equally, but instead transform auxiliary task data into forms that facilitate knowledge transfer for constituent parsing (§3.2), and make loss design for parsing optimization only (§3.3).

[1] LLMs, such as Open AI ChatGPT and GPT4, still underperform the standard chart-based parser under supervised training, as we will show in §4.

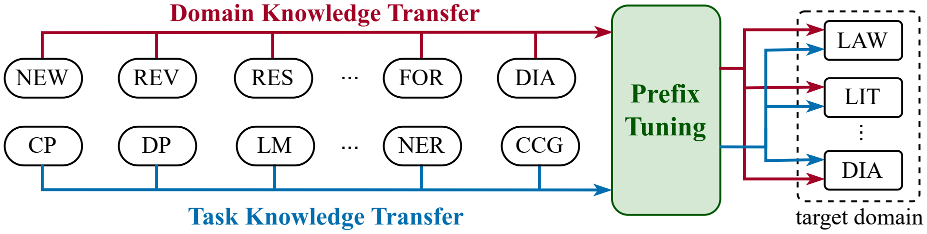


Figure 1: Cross-domain and cross-task knowledge transfer from multiple source domains and various heterogeneous tasks to the target domain constituency parsing based on prefix-tuning.

As shown in Figure 1, we select four auxiliary tasks to improve constituency parsing, including language modeling (LM) (Devlin et al., 2019), named entity recognition (NER) (Finkel & Manning, 2009), combinatory categorial grammar (CCG) supertagging (Steedman, 2001) and dependency parsing (DP) (Kübler, McDonald, & Nivre, 2009). We collect the datasets of these auxiliary tasks covering various domains for knowledge transfer on top of the source domain constituency treebank. The task domains include dialogue, forum, law, literature and review for LM, news and restaurant for NER, news for CCG and web for DP, for which existing labeled corpora are available. Given each training instance, we set the prefix values according to the task and domain for informing the representation model. After joint training, our parser effectively integrates both domain and task knowledge into a unified input representation.

We conduct experiments to verify the effectiveness of the proposed model on a news-domain constituency treebank PTB (Marcus, Santorini, & Marcinkiewicz, 1993) and a multi-domain constituency treebank MCTB (Yang, Cui, Ning, Wu, & Zhang, 2022) consisting of five domains: dialogue, forum, law, literature and review. Experimental results show that both domain knowledge transfer and task knowledge transfer are effective for cross-domain constituency parsing. Our cross-domain constituency parser gives the best reported performance on all of the five domains, outperforming various baselines, including chart-based parsers (Kitaev & Klein, 2018), transition-based parsers (Liu & Zhang, 2017) and ChatGPT (Ouyang, Wu, Jiang, Almeida, Wainwright, Mishkin, Zhang, Agarwal, Slama, Ray, et al., 2022).

To our knowledge, we are the first to allow a single constituent parser model to benefit from both cross-task and cross-domain knowledge from a wide range of heterogeneously labeled data, achieving the best reported results on standard benchmarks across different domains.[2]

2. Related Work

Cross-domain Constituency Parsing. Constituency parsing is an important and fundamental task in computational linguistics, which has not been completely solved. The main challenge is stable cross-domain parsing performance. Early work of constituency

[2] Our code is available at `https://github.com/guopeiming/CD_ConsParing_HeterData`.

parsing focuses on the news domain (Collins, 1997; Stern et al., 2017) and short sentences (McClosky, Charniak, & Johnson, 2006, 2008). In recent years, the natural language processing community has begun to pay attention to constituency parsing on different domains. So there has been limited work investigating cross-domain constituency parsing. McClosky et al. (2010) propose multiple source parser adaptation, which trains constituency parsers on multiple domain treebanks and combines these models by linear regression. Joshi, Peters, and Hopkins (2018) study single source domain adaptation based on the contextualized word representations, where they train the parsers on PTB only for similar target domains. For syntactically-distant target domains, they employ a dozen partial annotations to improve cross-domain constituency parsing performance. Fried et al. (2019) and Yang et al. (2022) perform a systematic analysis on various constituency parsers. Yang et al. (2022) annotate a constituency treebank MCTB, which contains five target domains. Our work is this line since we also focus on cross-domain constituency parsing. However, we investigate cross-domain and cross-task knowledge transfer for this problem to improve the utility of heterogeneously labeled data sources.

Parser architectures. Researchers have developed various constituency parsers for in-domain settings in the past years. Broadly speaking, there are four types of constituency parsers: (1) chart-based parser (Stern et al., 2017; Kitaev & Klein, 2018; Kitaev, Cao, & Klein, 2019), (2) transition-based parser (Zhu, Zhang, Chen, Zhang, & Zhu, 2013; Watanabe & Sumita, 2015; Liu & Zhang, 2017; Fernández-González & Gómez-Rodríguez, 2019; Yang & Deng, 2020), (3) sequence labeling-based parser (Gómez-Rodríguez & Vilares, 2018; Kitaev & Klein, 2020; Amini & Cotterell, 2022), and (4) sequence-to-sequence-based parser (Vinyals, Kaiser, Koo, Petrov, Sutskever, & Hinton, 2015; Liu, Zhu, & Shi, 2018; Yang & Tu, 2022). Based on the pretrained language models, researchers exploit various methods to improve the chart-based parser, which achieves state-of-the-art performance compared with the other three types of parsers (Zhou & Zhao, 2019; Zhang, Zhou, & Li, 2020; Tian et al., 2020; Cui et al., 2022; Shi, Wang, Xiao, & Liu, 2022). Concretely, one class of approaches adds regular terms to the loss objective to inject extra syntactic information into the chart-based parser, such as grammar rules (Shi et al., 2022), non-local features (Cui et al., 2022) or other parsing formalizations (Zhou & Zhao, 2019; Gu, Hou, Wang, Duan, & Li, 2024). The other line of work does not change the loss function, but encodes syntactic information (e.g., high-order features (Zhang et al., 2020) or n-grams (Tian et al., 2020; Kim, Cho, Kim, & Choi, 2023)) into the encoder to improve in-domain constituency parsing performance. Our parser combines the strengths of these two methods, extending the chart-based parser to the cross-domain scenario and integrating multi-dimensional heterogeneous information. We propose a novel multi-source prefix encoder and design a consistent loss objective for transferring cross-task and cross-domain knowledge to constituency parsing.

Knowledge Transfer for Constituency Parsing. Finkel and Manning (2009) transfer task knowledge to constituency parsing by a joint model of named entity recognition and constituency parsing, where entities are nested to the parse tree. Sun and Wan (2013) propose several strategies to acquire pseudo constituency treebanks only from dependency annotations. Zhou and Zhao (2019) exploit head-driven phrase structure grammar to encode dependency parsing and constituency parsing jointly. Their joint model absorbs dependency knowledge to improve constituency parsing performance. Yang and Tu (2022) formalize a nested named entity as a constituency tree and propose a pointer net for nested named entity recognition and constituency parsing to predict span boundaries in post-order. In contrast to these methods which focus on one joint task, our work aims to transfer auxiliary task knowledge to cross-domain constituency parsing, considering multiple tasks covering multiple domains. In addition, we are the first to investigate the task transfer between CCG supertagging and constituency parsing.

Prefix-tuning for Knowledge Transfer Prefix-tuning appends soft prefix vectors (Li & Liang, 2021) to the input sentence for task-specific representation learning. There has been work employing prefix-tuning for knowledge transfer. Yuan, Wang, Cao, and Li (2022) propose prefix-merging to transfer knowledge of text summarization and question answering to assist few-shot learning in query-focused summarization. Chen, Li, Deng, Tan, Xu, Huang, Si, Chen, and Zhang (2022), Chen, Li, Qiao, Zhang, Tan, Jiang, Huang, and Chen (2023) employ prefix-tuning to transfer domain knowledge for named entity recognition. All these works insert prefix into the generative pretrained language model such as BART (Lewis, Liu, Goyal, Ghazvininejad, Mohamed, Levy, Stoyanov, & Zettlemoyer, 2020) and GPT-2 (Radford, Wu, Child, Luan, Amodei, Sutskever, et al., 2019), in contrast we attempt to inject prefix vectors into an encoder-based pretrained language model, BERT (Devlin et al., 2019). Besides, they consider domain knowledge transfer or task knowledge transfer separately, while we consider task and domain knowledge transfer simultaneously and design multiple prefixes to fuse cross-domain and cross-task data for cross-domain constituency parsing. To the best of our knowledge, we are the first to empirically investigate prefix-tuning for integrating multi-dimensional heterogeneous information.

3. Method

We build our model based on a chart-based neural constituency parser (§ 3.1), selecting four auxiliary tasks (§ 3.2) as potential sources to maximize the utility of heterogeneous labeled datasets. To bridge knowledge across multiple auxiliary tasks and domains, we develop a framework (§ 3.3) that employs heterogeneous prefixes to integrate multi-dimensional heterogeneous knowledge for cross-domain constituency parsing.

3.1 Baseline Constituency Parser

We employ a chart-based neural constituency parser (Stern et al., 2017; Kitaev & Klein, 2018; Teng & Zhang, 2018) as the backbone model, which formalizes constituency parsing as a span-based classification task. The parser builds the hierarchical constituency syntax tree based on the scores of the labeled spans (i, j, l), where i, j and l are the start and end span endpoint and constituency label, respectively.

Given an input sentence $X = x_1 \cdots x_n$ (n is the length), our parser first computes the word representation $\boldsymbol{x}_i$ based on the pretrained language model (PLM). Following Kitaev and Klein (2018), word-level hidden representation $\boldsymbol{h}_i$ is generated by the stacked partitioned transformers, which extract contextual features with content and position attention. Then, a span encoder is adopted to obtain each span representation $\boldsymbol{s}_{i,j}$ in the sentence by subtracting of the word representation of the span endpoints, which is detailed in the Eq 5. Subsequently, based on the multi-layer perceptron (MLP), the parser assigns a score

$s^c(i, j, l)$ to each labeled span, which represents the score of the span as a constituent with the syntactic label l. The scoring process of the labeled span triplet is as follows:

$$\begin{aligned}\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n &= \text{PLM}(X) \\ \boldsymbol{h}_1, \cdots, \boldsymbol{h}_n &= \text{WordEncoder}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \\ \boldsymbol{s}_{i,j} &= \text{SpanEncoder}(\boldsymbol{h}_i, \boldsymbol{h}_j) \\ \boldsymbol{s}^c(i, j, \cdot) &= \text{MLP}^c(\boldsymbol{s}_{i,j}),\end{aligned} \tag{1}$$

where PLM denotes pretrained language model. Finally, the score of the constituency syntax tree T is computed by summing the scores of all the labeled spans within it. The chart-based parser exploits the CKY algorithm (Cocke, 1969; Kasami, 1966; Younger, 1967) to efficiently search for the parse tree with the highest score, which is then used as the predicted output $\hat{T}$:

$$\begin{aligned}s(T) &= \sum_{(i,j,l)\in T} s^c(i, j, l) \\ \widehat{T} &= \arg\max_{T} \; s(T).\end{aligned} \tag{2}$$

For training, we minimize the tree-based max-margin loss $\mathcal{L}^c$ following Stern et al. (2017) and Kitaev and Klein (2018):

$$\mathcal{L}^c = s(\widehat{T}) - s(T^*) + \Delta(\widehat{T}, T^*), \tag{3}$$

where Δ represents the Hamming difference between the predicted parse tree $\widehat{T}$ and the gold-standard parse tree T^*.

3.2 Auxiliary Tasks

We make use of heterogeneous data over four auxiliary tasks, conducting special data processing that converts cross-domain and cross-task data to a format closer to constituency parsing, which narrows the gap between auxiliary tasks and constituency parsing, thereby making knowledge transfer more efficient. Besides, we formulate the loss objectives of these four auxiliary tasks as a consistent format, which is described in §3.3.

Language Modeling. Language modeling can make PLM acquire domain knowledge on available raw corpora from the target domain (Gururangan, Marasović, Swayamdipta, Lo, Beltagy, Downey, & Smith, 2020). Therefore, we consider it as a simple and effective method to transfer knowledge for cross-domain constituency parsing. Following Devlin et al. (2019) and Gururangan et al. (2020), we perform masked language modeling on an encoder-based PLM (Devlin et al., 2019) and formulate it as a partial sequence labeling task as shown in Figure 2(a).

Instead of random sampling for token masking, we design a strategy of masking tokens based on their probability of acting as a constituent span boundary. Specifically, we first calculate the number of times each word in the sentence acts as a constituent span boundary and then normalize the frequency distribution to obtain the masking probability. In total, 30% of the tokens in the input sentence are masked, where masked tokens are substituted by the `[mask]` token, random token and themselves of the probability of 0.8, 0.1 and 0.1, respectively.

(a) Language Modeling
Input: Such an [mask] might be too [mask] to [mask] Rail .
Output: Such an approach might be too favourable to Queensland Rail .

(b) Named Entity Recognition
Input: Are there any 24 hour breakfast places nearby ?
Output: (3, 4, MISC), (5, 5, MISC), (7, 7, LOC)

(c) CCG Supertagging
Input: It has no bearing on our work force today .
Output: NP, (S[dcl]\NP)/NP, NP[nb]/N, N, (NP\NP)/NP, NP[nb]/N, N/N, N, NP\NP, .

(d) Dependency Parsing
Input: In Ramadi , there was a big demonstration .
Output: (1, 2, ←), (2, 5, ←), (3, 5, ←), (4, 5, ←), (0, 5, →), (6, 8, ←), (7, 8, ←), (5, 8, →), (5, 9, →)

Figure 2: Format of input and output for four auxiliary tasks. Language modeling and CCG supertagging take sequence labeling forms, while named entity recognition and dependency parsing are turned into span classification tasks.

Named Entity Recognition. We consider the NER task as it has been proven beneficial to constituency parsing (Finkel & Manning, 2009, 2010), where named entities usually correspond to the NP tag in the constituency syntax tree, which is the most frequent constituent category. As shown in Figure 2(b), NER also can be treated as a span classification task like consistency parsing.

For NER datasets, following Finkel and Manning (2009), we only retain the entity that matches a constituent span exactly or aligns to multiple continuous children nodes of a shared parent node. We omit the entities that cross non-sibling constituents to avoid introducing unnecessary ambiguity to constituency parsing. Additionally, we normalize entity types into four common categories: Person, Location, Organization, and Misc.

CCG Supertagging. Combinatory Categorial Grammar (Steedman, 2001) is a lexicalized grammatical formalism in which the lexical categories of words in a sentence are known as super tags. Such CCG super tags represent rich lexical syntactic knowledge, which can be treated as a form of shallow parsing. Therefore, CCG supertagging can provide relevant knowledge for hierarchical phrase structure syntax. We convert CCG treebank into token-level CCG supertags, which can be treated as a sequence labeling task as shown in Figure 2(c).

Dependency Parsing. Dependency parsing (Kübler et al., 2009) and constituency parsing are the two most popular sentence-level grammars in computational linguistics. Dependency trees adopt labeled dependency arcs to represent syntactic information, while constituency trees use hierarchical nested constituents to organize sentences. These two grammar formalisms can be converted into each other (Magerman, 1994; Nivre, Hall, & Nilsson, 2006; Johansson & Nugues, 2007; Xia & Palmer, 2001) and syntactic task knowledge can be transferred between dependency and constituency parsing (Sun & Wan, 2013; Zhou & Zhao, 2019; Gu et al., 2024).

We formulate dependency parsing as a span classification task and integrate dependency structure into chart-based constituency parser for better syntax task knowledge transfer.

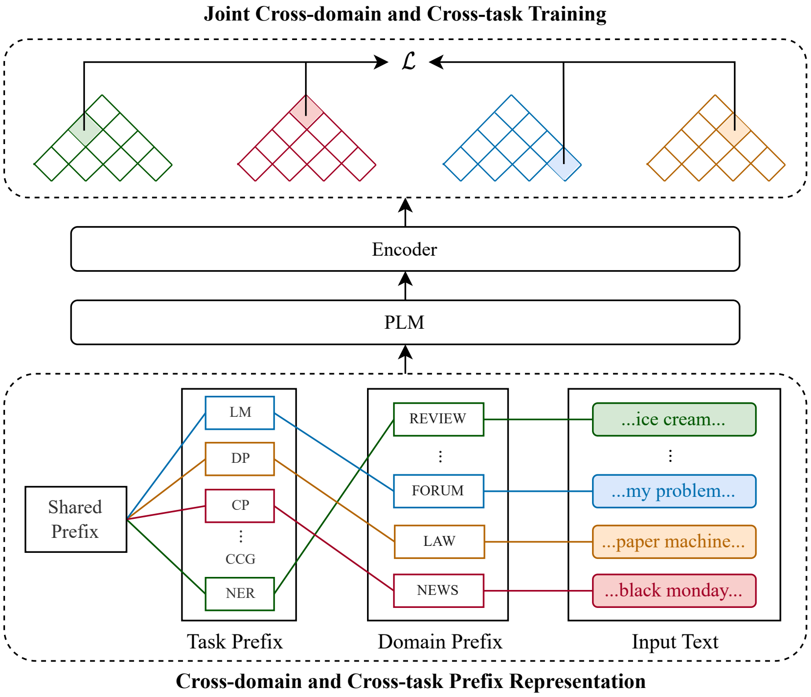


Figure 3: The architecture of prefix-based cross-domain and cross-task knowledge transfer model.

Specifically, a dependency tree is composed of directed dependency arcs with dependency relation labels, which link head words and dependent words. We treat the dependency arc as a span, where the head and dependent word are the span boundaries. Compared with dependency relation labels, dependency directions imply the hierarchical relationship of constituent spans, which is essential for constituency parsing. As a result, we use dependency arc direction as the label of the span as shown in Figure 2(d).

3.3 Prefix-based Knowledge Transfer

The overall prefix-based cross-domain and cross-task knowledge transfer model is illustrated in Figure 3. Specifically, the model enhances the basic constituency parser for cross-domain constituency parsing from two perspectives: first, cross-domain and cross-task prefix representation adopts different prefixes to decompose the general, task and domain representations in the PLM and fuse multi-dimensional heterogeneous data. Second, cross-domain and cross-task joint training transfers task and domain knowledge to cross-domain constituency parsing based on the supervised signals from various tasks.

Cross-domain and Cross-task Prefix Representation. As a representative method of soft prompt, prefix tuning (Li & Liang, 2021) inserts an additional key-value prefix vector pair into each transformer layer in the PLM. As shown in Figure 3, we define three types of prefixes to decompose the different aspects of knowledge and representation. Specifically, the model exploits the task and domain prefix to extract the feature related to task formalization and domain distribution, respectively. Besides, the shared prefix is responsible to activate the general knowledge in PLM, which is agnostic to task and domain.

For the input sentence from different tasks and domains, X^t and X^d denote the task type and domain type, respectively. The cross-domain and cross-task prefix representation finds the corresponding task soft prompt vector $\boldsymbol{E}^t$ and domain soft prompt vector $\boldsymbol{E}^d$ and prepends the shared soft prompt vector $\boldsymbol{E}^s$ to them. In particular, domain prefix activates domain-specific features in PLM, while task prefix focuses on extracting task-specific representations. The shared prefix can control domain- and task-agnostic knowledge of PLM. Following Li and Liang (2021) and Yuan et al. (2022), we then exploit reparametrization to generate the shared prefix $\boldsymbol{P}^s$, task prefix $\boldsymbol{P}^t$ and domain prefix $\boldsymbol{P}^d$, which leads to stable optimization and better performance: Finally, we inject the prefix vectors into the transformer layers and generate word representations based on the PLM:

$$\begin{aligned}\boldsymbol{E}^s, \boldsymbol{E}^t, \boldsymbol{E}^d &= \mathrm{SoftPrompt}(X^t, X^d)\\ \boldsymbol{P}^s, \boldsymbol{P}^t, \boldsymbol{P}^d &= \mathrm{MLP}([\boldsymbol{E}^s; \boldsymbol{E}^t; \boldsymbol{E}^d])\\ \boldsymbol{x}_1, \cdots, \boldsymbol{x}_n &= [\boldsymbol{P}^s; \boldsymbol{P}^t; \boldsymbol{P}^d] \diamond \mathrm{PLM}(X),\end{aligned} \tag{4}$$

where $\boldsymbol{E}^s$, $\boldsymbol{E}^t$ and $\boldsymbol{E}^d \in \mathbb{R}^{l \times s}$ are learnable parameters, l and s are the length and hidden size of the prefix prompt, respectively, $\diamond$ is the prefix vector injection operation.

Joint Training. Cross-domain and cross-task training integrates the supervised signals of different tasks into the cross-domain constituency parser and achieves task and domain knowledge transfer. To mitigate negative transfer in multi-task learning and narrow the task formalization gap between constituency parsing and auxiliary tasks, joint cross-domain and cross-task training in Figure 3 converts NER and DP into span-based classification tasks and applies the final classification layers of language model and CCG supertagging to span-level representations, rather than token-level representations. As a result, the span representation $\boldsymbol{s}_{i,j}$ for the three aforementioned auxiliary tasks is computed by the shared encoder and PLM. Then we feed the span representations into different MLPs for scoring task-dependent label sets:

$$\begin{aligned}\boldsymbol{s}_{i,j} &= (\overrightarrow{\boldsymbol{h}_j} - \overrightarrow{\boldsymbol{h}_{i-1}}) \oplus (\overleftarrow{\boldsymbol{h}_{j+1}} - \overleftarrow{\boldsymbol{h}_i})\\ \boldsymbol{s}^{\mathcal{A}}(i,j,\cdot) &= \mathrm{MLP}^{\mathcal{A}}(\boldsymbol{s}_{i,j}),\end{aligned} \tag{5}$$

where $\mathcal{A} \in \{\mathrm{LM}, \mathrm{NER}, \mathrm{CCG}, \mathrm{DP}\}$.

For each auxiliary tasks, joint training minimizes the cross-entropy loss function between the predicted probability distribution $\hat{\boldsymbol{p}}^{\mathcal{A}}_{i,j}$ and the gold-standard labels $\boldsymbol{p}^{*\mathcal{A}}_{i,j}$ based on softmax:

$$\begin{aligned}\hat{\boldsymbol{p}}^{\mathcal{A}}_{i,j} &= \mathrm{SoftMax}(\boldsymbol{s}^{\mathcal{A}}(i,j,\cdot))\\ \mathcal{L}^{\mathcal{A}} &= -\sum_{1 \le i \le j \le N} \boldsymbol{p}^{*\mathcal{A}}_{i,j} \; log \; \hat{\boldsymbol{p}}^{\mathcal{A}}_{i,j}\end{aligned} \tag{6}$$

In particular, language modeling and CCG supertagging are token classification tasks, while NER and DP are span classification tasks. Therefore, for language modeling and CCG supertagging, we only compute the loss on the spans of length 1 as illustrated in the blue leaf node in Figure 3. In other words, j always equals to i in Eq 6 for these two tasks.

Dataset	**Task**	**Domain**	**#Sentence**
Multi-task Training Data			
PTB	Constituency Parsing	news	39,832
CoNLL03	Named Entity Recognition	news	10,000
restaurant	Named Entity Recognition	restaurant	8,662
ccgbank	CCG Supertagging	news	10,000
EWT	Dependency Parsing	web	10,000
Wizard	Language Modeling	dialogue	10,000
Reddit	Language Modeling	forum	10,000
ECtHR	Language Modeling	law	10,000
Gutenberg	Language Modeling	literature	10,000
Amazon	Language Modeling	review	10,000
Low Resource Evaluation Data			
MCTB	Constituency Parsing	dialogue	1,000
	Constituency Parsing	forum	1,000
	Constituency Parsing	law	1,000
	Constituency Parsing	literature	1,000
	Constituency Parsing	review	1,000

Table 1: Summary of Datasets used in our paper. We train our model on source constituency parsing (PTB) and auxilary tasks in a multi-task learning manner and evaluate the resulting model on the MCTB benchmark in a low resource setting.

Finally, we jointly optimize the multi-task loss functions:

$$\mathcal{L} = \mathcal{L}^c + \alpha \mathcal{L}^{\mathcal{A}}, \tag{7}$$

where α is a factor to weight the auxiliary tasks.

Test Scenarios For the zero-shot scenario, we add the domain prefix that is tuned on the LM corpora and the constituency parsing task prefix to the input sentence. For the few-shot scenario, we first pre-train the whole parser on the multi-dimensional heterogeneous datasets and then fine-tune the domain prefix again on the limited number of examples.

4. Experiments

We conduct experiments to verify the effectiveness of our proposed knowledge transfer method and analyse to gain a deeper understanding of knowledge transfer for cross-domain constituency parsing.

4.1 Experimental Setup

Datasets. We use PTB (Marcus et al., 1993) and MCTB (Yang et al., 2022) as the source and target constituency parsing datasets, respectively. For domain knowledge transfer, we collect 5 domain raw corpora with sources matching the target treebank in MCTB for the language modeling task, including Wizard (Dinan, Roller, Shuster, Fan, Auli, & Weston, 2019), Reddit (Völske, Potthast, Syed, & Stein, 2017), ECtHR (Stiansen & Voeten, 2019), Gutenberg[3], and Amazon (He & McAuley, 2016).

For task knowledge transfer, we select CoNLL03 (Tjong Kim Sang & De Meulder, 2003) and restaurant (Liu, Meng, Zhang, Xu, Chen, & Zhou, 2019b) for NER, ccgbank (Hockenmaier & Steedman, 2007) for CCG supertagging and EWT treebank in universal dependencies v2.2 (Nivre, de Marneffe, Ginter, Hajič, Manning, Pyysalo, Schuster, Tyers, & Zeman, 2020) for dependency parsing. We sample 10,000 sentences with lengths ranging from 8 to 256 for the corpora of auxiliary tasks. If the number of filtered sentences is less than 10,000, we include the entire dataset. For each batch, we sample examples of constituency parsing and auxiliary tasks by the 1:3 proportion. Specific tasks, domains and number of sentences are listed in Table 1. Additionally, we obtain pseudo constituency parse trees for data processing of auxiliary tasks using the basic constituency parser. Specifically, we sample 10/20/50 examples from MCTB for the few-shot setting. To avoid sample bias, we sample three times to generate different few-shot training sets by different seeds and report the average results.

Evaluation. We use precision (P), recall (R), and F1 score (F1) of labeled bracketed spans to evaluate the performance of constituency parsing. In particular, we compute metrics of parsing via the standard toolkit evalb[4]. We conduct the experiments on three different random seeds and report the average results, ignoring punctuation following (Kitaev & Klein, 2018).

In particular, we evaluate our cross-domain and cross-task knowledge transfer model on the constituency parsing task only and do not report the performance of the auxiliary tasks. This is because both the training objective and the task formulation of auxiliary tasks are designed for constituency parsing optimization, which makes evaluating the auxiliary tasks difficult. Take NER for example, we delete entities crossing constituency spans and unify entity types into four categories. Therefore, it is difficult to recognize some entities, especially cross-domain entities with unseen types. As for dependency parsing, our auxiliary loss function is different from the conventional optimization objective, and it only involves dependency directions not dependency relations.

Hyperparameters. We use BERT-large-uncased as pretrained language model backbone (Devlin et al., 2019). The lengths l and hidden sizes d of shared, task and domain prefix are 25 and 1024, respectively. Weight factor of auxiliary tasks α is 0.1 for multi-task learning. Following Kitaev and Klein (2018), we set partition transformer layers to 2 for all chat-based parsers. For model training, we use the AdamW algorithm with learning rate 3e-5, batch size 60, weight decay 0.01, linear learning rate warmup over the first 400 steps to optimize parameters. We stop early training when the F1 score does not increase on the PTB development set for 4 epochs.

Baselines. We compare the proposed method with the following baseline models: (1) a strong *Transition*-based model (Liu & Zhang, 2017), whose results are reported by Yang et al. (2022), (2) a strong *chart*-based model (Kitaev & Klein, 2018), which is re-implemented by us as the basic constituency parser and (3) *DAPT* (Gururangan et al.,

[3] `https://www.gutenberg.org/`

[4] `https://nlp.cs.nyu.edu/evalb/`

Method		Dialogue			Forum			Law			Literature			Review		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ChatGPT	full	32.23	28.79	30.54	20.18	17.70	18.86	34.54	22.90	24.93	12.78	11.24	11.96	31.33	26.42	28.67
	valid only	74.97	66.33	70.38	75.30	66.03	70.36	88.57	74.11	80.70	79.87	70.23	74.74	75.50	63.66	69.08
Transition		–	–	85.56	–	–	86.33	–	–	91.50	–	–	84.96	–	–	83.89
DAPT		88.28	84.31	86.25	87.63	86.46	87.04	94.35	89.77	92.00	86.26	86.59	86.42	86.13	81.71	83.86
Chart		88.12	84.18	86.10	87.59	86.27	86.92	94.36	89.89	92.07	86.13	86.43	86.28	86.71	82.05	84.32
CL^S (Ours)		88.11	84.53	86.28	87.67	86.30	86.98	94.00	90.03	91.97	86.25	87.08	86.66	86.86	82.13	84.43
CL (Ours)		88.42	84.46	86.39	87.85	86.28	87.06	94.03	90.05	92.00	86.29	87.16	86.72	87.16	82.26	84.64
CLD (Ours)		88.50	84.52	86.46	87.95	86.44	87.19	94.11	90.20	92.11	86.37	87.22	86.79	87.26	82.74	84.94
CLN (Ours)		88.57	84.65	86.57	87.99	86.39	87.18	94.20	90.28	92.20	86.28	87.19	86.73	87.21	82.60	84.84
CLT (Ours)		88.56	84.44	86.45	87.81	86.47	87.14	94.21	90.41	92.27	86.14	87.50	86.82	87.24	82.64	84.88
CLDNT (Ours)		**88.63**	**85.49**	**87.03**	**88.42**	**86.82**	**87.61**	**94.37**	**90.55**	**92.42**	**86.37**	**87.36**	**86.86**	**87.24**	**82.90**	**85.01**

Table 2: Zero-shot results on MCTB benchmark. C, L, N, T and D indicate constituency parsing, language modeling, named entity recognition, CCG supertagging and dependency parsing, respectively. Specially, for the language modeling task, we perform experiments on the single target domain corpus L^S and multiple domain corpora L to show the influence of cross-domain texts. The best results and the second-best results of each group are noted by **bold** and underline, respectively.

2020), which continues pretraining on the target domain texts and then fine-tunes the chart-based constituency parser on the source treebank.

We also report the constituency parsing performance of *ChatGPT* (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, et al., 2020; Ouyang et al., 2022) [5], which is one of the most popular large-scale language models at present. We use gpt-3.5-turbo to generate bracketed parse tree with in-context-learning (ICL) (Brown et al., 2020), where 10 constituency tree examples are pre-pended before the testing instance as demonstrations. For zero-shot cross-domain constituency parsing, we select demonstrations from the source treebank PTB, to which we refer as *ChatGPT* in Table 2. For few-shot learning, we employ examples from the target domain treebank (MCTB), following the same settings for the other models in Table 3.

Notably, the outputs can contain numerous errors, including unmatched brackets, omitted words from input sentences, and responses lacking bracketed parse trees, because ChatGPT predicts the next token auto-regressively and does not ensure the generation of valid constituency parse trees. We report results both considering and not considering invalid trees in Table 2 and Table 3.

4.2 Main Results

In this subsection, we report results on zero-shot and few-shot settings to verify the effectiveness of our proposed approach.

Zero-shot Results. Table 2 lists zero-shot results on the 5 target domains in MCTB. Based on the basic constituency parser, *Chart*, we incrementally append auxiliary tasks (C: constituency parsing, L: language modeling, D: dependency parsing, N: NER and T:

[5] `https://chat.openai.com/`

Method		**Dialogue**			**Forum**			**Law**			**Literature**			**Review**		
		10	**20**	**50**	**10**	**20**	**50**	**10**	**20**	**50**	**10**	**20**	**50**	**10**	**20**	**50**
ChatGPT	full	40.02	–	–	27.31	–	–	23.89	–	–	19.82	–	–	41.25	–	–
	valid only	73.16	–	–	73.41	–	–	79.64	–	–	76.82	–	–	71.24	–	–
Chart		86.82	87.37	87.62	87.69	87.98	88.13	92.35	92.57	92.70	86.90	87.34	87.55	84.96	85.28	85.50
CLDNT		**87.65**	**87.82**	**88.21**	**88.15**	**88.33**	**88.51**	**92.54**	**92.78**	**93.00**	**87.89**	**88.38**	**88.32**	**85.87**	**86.06**	**86.44**

Table 3: Few-shot results (F1 score) for the target domains on 10, 20 and 50 shots. Please note that due to the length limit of OpenAI API, we only present the 10-shot ICL results for ChatGPT.

CCG supertagging) to transfer domain and task knowledge for cross-domain constituency parsing.

First, our *CLDNT* method, which incorporates all auxiliary tasks from diverse domains, outperforms all baselines by a large margin. In particular, *CLDNT* (avg. 87.78 F1) shows better performance than previous best reported *Chart* (avg. 87.14 F1), on which *CLDNT* is built, across domains, with an averaged improvement of 0.64 F1 score. In addition, we see that model performance varies across domains, and all models show the best performance in *Law*. One main reason can be that text in *Law* is most similar to general news text, with formal language written in a monologue style.

Second, with labeled data for more tasks being integrated (e.g., from *CL*, *CLN* to *CLDNT*), our framework can benefit from diverse tasks of multiple domains. Compared with CL (avg. 87.36 F1), our three auxiliary structure prediction tasks, dependency parsing (*D*), NER (*N*) and CCG supertagging (*T*), advance the results on all domains, with an improvement of 0.42 F1 score. Such results show that our framework successfully allows model to transfer the task knowledge to constituency parsing of a target domain. Also, compared with only using vanilla *DAPT* for domain adaptation, our *CL* also shows higher performance across all tasks. This verifies that our designs of span boundary masking (§ 3.2) is more efficient than vanilla *DAPT* and can be a more useful.

Third, *Transition* underperforms *Chart* across five target domains. This observation applies to in-domain settings as well (Yang et al., 2022). In fact, the chart-based parser also is superior to sequence-labeling-based parsers and sequence-to-sequence-based parsers (Amini & Cotterell, 2022). Based on the pretrained language models, the chart-based parser achieves competitive parsing performance.

Finally, *ChatGPT* shows poor performance on all domains, which suggests that such generative LLMs can be less capable of solving structure prediction problems (Roy, Thomson, Chen, Shin, Pauls, Eisner, & Van Durme, 2024). We find that ChatGPT tends to generate invalid parse trees. Take the input sentence "*He is right .*" for example, ChatGPT might generate unmatched brackets (e.g., *"[S [NP [PRP He]] [VP [VBD was] [ADJP [JJ right] [. .]"*) or drop sentential words (e.g., *"[S [VP [VBD was] [ADJP [JJ right]]] [. .]]"*). Therefore, when taking all outputs of ChatGPT (the second line *full*) into evaluation, its performance decreases severely.

Few-shot Results. Table 3 reports few-shot results (F1 scores) for the 5 target domains on 10, 20 and 50 shots. First, our *CLDNT* model outperforms *Chart* baseline and *ChatGPT* by a large margin in all domains and all few-shot settings, which demonstrates the effec-

Method	**R**	**P**	**F**
CLDNT	88.63	85.49	87.03
-Prefix	-1.06	-0.97	-1.01
-EM	-0.35	-0.27	-0.30
-DD	-0.33	-0.24	-0.28
-LC	-0.22	-0.19	-0.20
-BM	-0.16	-0.15	-0.15

Table 4: Ablation experiments. Prefix: vanilla-prefix, which removes our novel multiple prefix strategy. EM: entity match, DD: dependency direction as span label, LC: label conversion. BM: boundary masking.

tiveness of our method. Furthermore, compared with zero-shot performance, our *CLDNT* model and *Chart* show better performance, and when provided with more training instances (from 10 to 50), model performance consistently improves, which can be attributed to the fact that models are presented with training instances of target domains.

It is worth noting that our 10-shot *CLDNT* can outperform 50-shot *Chart* on most tasks (e.g., for the dialogue domain, 10-shot *CLDNT*: 87.65, 50-shot *Chart*: 87.62), which shows that our method can efficiently adapt knowledge from limited training instances, thereby having its advantages in a low-resource setting.

4.3 Analysis

To better understand the effectiveness of cross-domain and cross-task knowledge transfer of our method, we conduct in-depth analyses on the model output in the *dialogue* domain unless when otherwise specified.

Ablation Experiment. To verify the effectiveness of our multiple prefix strategy, we use vanilla-prefix to denote conventional prefix-tuning, which exploits single shared prefix with the same length as our proposed model. Intuitively, vanilla-prefix can not handle merging multi-source heterogeneous corpora to transfer domain knowledge and task knowledge because it appends the same prefix to all heterogeneous input data. We also perform ablation experiment to verify the effectiveness of the data processing in § 3.2, which transforms the diverse data from various auxiliary tasks into the format closer to constituency parsing. Specifically, EM and LC represent entity match and label conversion for the named entity recognition task, while constituent boundary mask for the language model task and dependency direction as span label for dependency parsing task are denoted as BM and DD, respectively. -DD means that the dependency relation is exploited as the span label not the dependency direction.

Table 4 reports the results of ablation experiments. The results of vanilla-prefix are even lower than the basic chart-based constituency parser. In addition, for the three data processing operations, EM has the greatest impact. This is reasonable because unmatched entities introduce substantial noise for cross-domain constituency parsing.

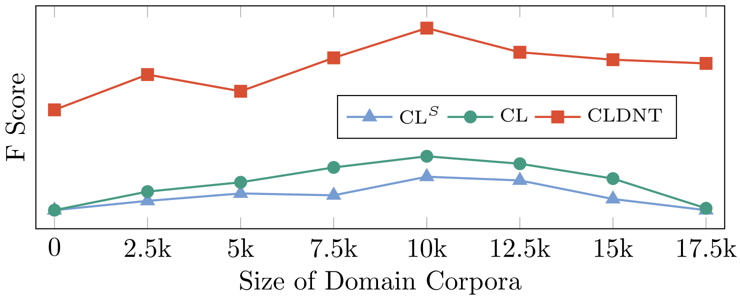
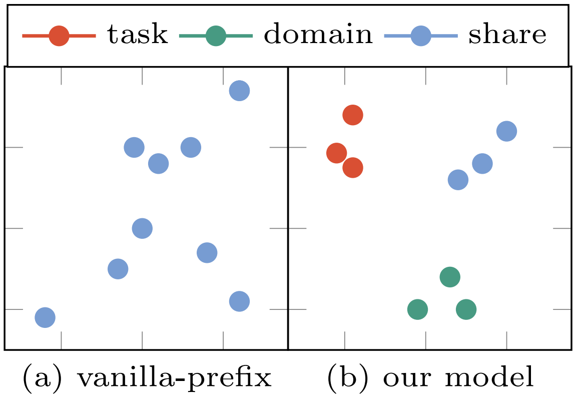


Figure 4: F score of different models with respect to the size of domain corpora.



(a) vanilla-prefix (b) our model

Figure 5: Prefix embedding visualization of vanilla-prefix and our model.

Corpora Size. Intuitively, the performance of our cross-domain and cross-task knowledge transfer model should be related to the size of domain corpora. Here we conduct an experimental study to examine the relation between F1 score and domain corpus size for three models: *CL*^{S}, *CL* and *CLDNT*. The results are shown in Figure 4, where x-axis denotes the size of domain corpora and y-axis denotes F1 score for the target domain constituency parsing.

When the size of domain corpora is zero, auxiliary tasks of our model do not contain language model, thus *CL*^{S} and *CL* are equivalent to the basic chart-based constituency parser. As the size grows larger in the initial phase, the F1 score of all the models increase significantly, which demonstrates the effectiveness of domain corpora for language model task for cross-domain constituency parsing. The performance stops increasing after 10k sentences for each domain are utilized, which could be noise for our cross-domain and cross-task knowledge transfer model. The larger corpora of auxiliary tasks are noise for our primary task, constituency parsing. In other words, this phenomenon can also be understood from the angle that the larger corpora implicitly increased the weight of the relevant task and domain, and hurt the performance of cross-domain constituency parsing. Besides, there is a large gap between *CLDNT* and the other models, which shows the effectiveness of more auxiliary tasks, including dependency parsing, named entity recognition and CCG supertagging.

Prefix Visualization. The learned prefix embeddings can be visualized to observe the relationship between tasks and domains. Here, we show the visualization results in Figure 5, where Principal Component Analysis (PCA) is applied to map the high-dimension prefix embedding representation to two-dimentional space. For the cross-domain and cross-task knowledge transfer model, the first three embeddings in shared, task and domain prefix are selected for visualization. For vanilla-prefix, we select the prefix embedding with the same position as our model. The embeddings from the same prefix are closer to each other in Figure 5b, where our multiple prefix strategy can distinguish general, task and domain representations. However, there is no pattern in the visualization of vanilla-prefix.

5. Conclusion

We leveraged heterogeneous data to transfer cross-domain and cross-task knowledge to constituency parsing, selecting language model, named entity recognition, CCG supertagging and dependency parsing as auxiliary tasks, proposing a novel multiple prefixes strategy to make use of heterogeneous source of labeled and unlabeled data. Experimental results showed that our cross-domain constituency parser gains the state-of-the-art performance on a range of test domains compared with various baselines, including basic chart-based parser, transition-based parser and ChatGPT. To our knowledge, this is the first attempt to make use of the most available multi-source heterogeneous data to improve constituency parsing.

Acknowledgments

We sincerely thank the reviewers for their invaluable feedback. This work was done when Peiming Guo was a research assistant at Westlake University. The corresponding author of this paper is Yue Zhang. This work was supported by the National Natural Science Foundation of China under Grant 61976180 and 62176180.

References

Amini, A., & Cotterell, R. (2022). On parsing as tagging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Bingel, J., & Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. In *Advances in neural information processing systems*.

Chen, X., Li, L., Deng, S., Tan, C., Xu, C., Huang, F., Si, L., Chen, H., & Zhang, N. (2022). Lightner: A lightweight tuning paradigm for low-resource ner via pluggable prompting. In *Proceedings of the 29th International Conference on Computational Linguistics*.

Chen, X., Li, L., Qiao, S., Zhang, N., Tan, C., Jiang, Y., Huang, F., & Chen, H. (2023). One model for all domains: Collaborative domain-prefix tuning for cross-domain ner. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*.

Cocke, J. (1969). *Programming languages and their compilers: Preliminary notes*. New York University.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*.

Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. In *arXiv*.

Cui, L., Yang, S., & Zhang, Y. (2022). Investigating non-local features for neural constituency parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (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*.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., & Weston, J. (2019). Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Fernández-González, D., & Gómez-Rodríguez, C. (2019). Faster shift-reduce constituent parsing with a non-binary, bottom-up strategy. *Artificial Intelligence*, *275*, 559–574.

Finkel, J. R., & Manning, C. D. (2009). Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Finkel, J. R., & Manning, C. D. (2010). Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Fried, D., Kitaev, N., & Klein, D. (2019). Cross-domain generalization of neural constituency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR.

Gómez-Rodríguez, C., & Vilares, D. (2018). Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Gu, Y., Hou, Y., Wang, Z., Duan, X., & Li, Z. (2024). High-order joint constituency and dependency parsing. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. ELRA and ICCL.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

He, R., & McAuley, J. (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*.

Hockenmaier, J., & Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, *33*(3), 355–396.

Johansson, R., & Nugues, P. (2007). Extended constituent-to-dependency conversion for English. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*. University of Tartu, Estonia.

Joshi, V., Peters, M., & Hopkins, M. (2018). Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Kasami, T. (1966). An efficient recognition and syntax-analysis algorithm for context-free languages. In *Coordinated Science Laboratory Report no. R-257*. Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

Kim, S., Cho, W., Kim, M., & Choi, Y. (2023). Bidirectional masked self-attention and n-gram span attention for constituency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. Association for Computational Linguistics.

Kitaev, N., Cao, S., & Klein, D. (2019). Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Kitaev, N., & Klein, D. (2018). Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Kitaev, N., & Klein, D. (2020). Tetra-tagging: Word-synchronous parsing with linear-time inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Kübler, S., McDonald, R., & Nivre, J. (2009). Dependency parsing. In *Dependency parsing*, pp. 11–20. Springer.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Li, J., Zhang, M., Guo, P., Zhang, M., & Zhang, Y. (2023). Llm-enhanced self-training for cross-domain constituency parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Liu, J., & Zhang, Y. (2017). In-order transition-based constituent parsing. *Transactions of the Association for Computational Linguistics*, *5*.

Liu, L., Zhu, M., & Shi, S. (2018). Improving sequence-to-sequence constituency parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, P., Qiu, X., & Huang, X. (2017). Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Liu, S., Johns, E., & Davison, A. J. (2019a). End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y., Meng, F., Zhang, J., Xu, J., Chen, Y., & Zhou, J. (2019b). GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Magerman, D. M. (1994). *Natural language parsing as statistical pattern recognition.* stanford university.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

McClosky, D., Charniak, E., & Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*.

McClosky, D., Charniak, E., & Johnson, M. (2008). When is self-training effective for parsing?. In *Proceedings of the 22nd International Conference on Computational Linguistics*.

McClosky, D., Charniak, E., & Johnson, M. (2010). Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Nivre, J., Hall, J., & Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing.. In *LREC*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., & Yang, D. (2023). Is chatgpt a general-purpose natural language processing task solver?. In *arXiv*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. In *OpenAI blog*.

Roy, S., Thomson, S., Chen, T., Shin, R., Pauls, A., Eisner, J., & Van Durme, B. (2024). Benchclamp: a benchmark for evaluating language models on syntactic and semantic parsing. In *Proceedings of the 37th International Conference on Neural Information Processing Systems.*

Shi, T., Wang, Z., Xiao, L., & Liu, C. (2022). Fast rule-based decoding: Revisiting syntactic rules in neural constituency parsing. In *arXiv.*

Søgaard, A., & Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics.*

Steedman, M. (2001). *The syntactic process.* MIT press.

Stern, M., Andreas, J., & Klein, D. (2017). A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.*

Stiansen, Ø., & Voeten, E. (2019). ECtHR judgments..

Sun, W., & Wan, X. (2013). Data-driven, PCFG-based and Pseudo-PCFG-based Models for Chinese Dependency Parsing. *Transactions of the Association for Computational Linguistics*, *1*, 301–314.

Teng, Z., & Zhang, Y. (2018). Two local models for neural constituent parsing. In *Proceedings of the 27th International Conference on Computational Linguistics.*

Tian, Y., Song, Y., Xia, F., & Zhang, T. (2020). Improving constituency parsing with span attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020.*

Tjong Kim Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003.*

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., & Hinton, G. (2015). Grammar as a foreign language. In *Advances in neural information processing systems.*

Völske, M., Potthast, M., Syed, S., & Stein, B. (2017). TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization.*

Watanabe, T., & Sumita, E. (2015). Transition-based neural constituent parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing.*

Wu, H., Zhang, Y., Jin, X., Xue, Y., & Wang, Z. (2019). Shared-private lstm for multidomain text classification. In *Natural Language Processing and Chinese Computing.* Springer International Publishing.

Xia, F., & Palmer, M. (2001). Converting dependency structures to phrase structures. In *Proceedings of the First International Conference on Human Language Technology Research.*

Yang, K., & Deng, J. (2020). Strongly incremental constituency parsing with graph neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Yang, S., Cui, L., Ning, R., Wu, D., & Zhang, Y. (2022). Challenges to open-domain constituency parsing. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Yang, S., & Tu, K. (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*.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n3. *Information and control*, *10*(2), 189–208.

Yuan, R., Wang, Z., Cao, Z., & Li, W. (2022). Few-shot query-focused summarization with prefix-merging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Zhang, Y., & Yang, Q. (2022). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), 5586–5609.

Zhang, Y., Zhou, H., & Li, Z. (2020). Fast and accurate neural crf constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*.

Zhou, J., & Zhao, H. (2019). Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Zhu, M., Zhang, Y., Chen, W., Zhang, M., & Zhu, J. (2013). Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.