Dynamic Syntactic Feature Filtering and Injecting Networks for Cross-lingual Dependency Parsing

Jianjian Liu, Zhengtao Yu, Ying Li*, Yuxin Huang, Shengxiang Gao

Yunnan Provincial Key Laboratory of Artificial Intelligence, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650000, China
jjliu_nj@foxmail.com, ztyu@hotmail.com, {yingli_hlt, gaoshengxiang.yn}@foxmail.com, huangyuxin2004@163.com

Abstract

Pre-trained language models enhanced parsers have achieved outstanding performance in rich-resource languages. Cross-lingual dependency parsing aims to learn useful knowledge from high-resource languages to alleviate data scarcity in low-resource languages. However, effectively reducing the syntactic structure distributional bias and excavating the commonalities among languages is the key challenge for cross-lingual dependency parsing. To address this issue, we propose novel dynamic syntactic feature filtering and injecting networks based on the typical shared-private model that employs one shared and two private encoders to separate source and target language features. Concretely, a Language-Specific Filtering Network (LSFN) on private encoders emphasizes helpful information and ignores the irrelevant or harmful parts of it from the source language. Meanwhile, a Language-Invariant Injecting Network (LIIN) on the shared encoder integrates the advantages of BiLSTM and improved Transformer encoders to transcend language boundaries, thus amplifying syntactic commonalities across languages. Experiments on seven benchmark datasets show that our model achieves an average absolute gain of 1.84 UAS and 3.43 LAS compared with the shared-private model. Comparative experiments validate that both LSFN and LIIN components are complementary in transferring beneficial knowledge from source to target languages. Detailed analyses highlight that our model can effectively capture linguistic commonalities and mitigate the effect of distributional bias, showcasing its robustness and efficacy.

Code — https://github.com/Flamelunar/crosslingualDP

Introduction

The purpose of dependency parsing is to identify and describe the syntactic and grammatical relationships between input words via a dependency tree. Figure 1 depicts a dependency tree, where a dependency arc from the head word "hợp đồng (contract)" to the modifier word "vô hiệu (invalid)" with relation label "amod" indicates that "vô hiệu (invalid)" serves as an adjective to modify "hợp đồng (contract)". These dependency trees reveal the syntax information via a hierarchical structure which is easily injected into various artificial intelligence models, i.e., neural machine translation (Yamin, Sarno, and Tambunan 2024), grammatical error correction (Tang, Qu, and Wu 2024), and question answering (Hou et al. 2024).

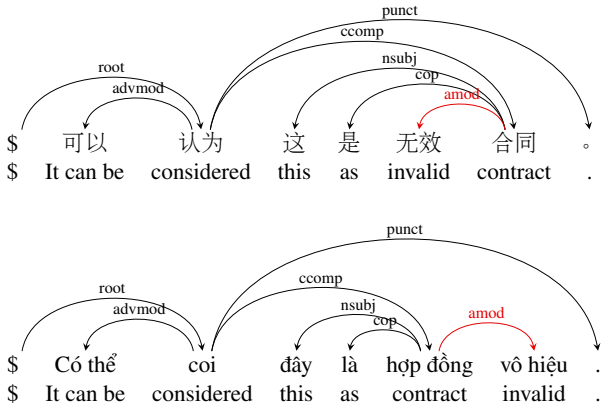

Figure 1: Examples of the dependency trees, with the Chinese on the top and the Vietnamese on the bottom.

Recent researchers have focused on integrating strong representations of pre-trained language models into dependency parsers, enhancing the parsing performance significantly (Ross, Cai, and Min 2020; Yao, Xue, and Min 2022; Nishida and Matsumoto 2022a; Gu et al. 2024). In its early stages, Dozat and Manning (2017) propose a BiAffine parser which employs GloVe word embeddings (Pennington, Socher, and Manning 2014) as its inputs and utilizes a multi-layer BiLSTM to encode contextual information, achieving excellent performances on multiple languages. Li et al. (2019a) improve the BiAffine parser by integrating the representations of ELMo (Peters et al. 2018) and BERT (Devlin et al. 2019). Nguyen (2020) utilize the fine-tuned BERT model and extra POS tagging task to improve the Vietnamese parsing performance. However, the effectiveness of these models is still limited for low-resource languages due to insufficient training data (Rotman and Reichart 2019; Wang et al. 2020; Effland and Collins 2023).

To mitigate the scarce corpus of low-resource languages, cross-lingual dependency parsing has gained more attention,

*Corresponding author.

which mainly transfers useful knowledge from rich-resource languages to enhance parsing accuracy in low-resource languages (Schuster et al. 2019; Lauscher et al. 2020; Ansell et al. 2021). Sun, Li, and Zhao (2023) employ a self-training strategy to construct pseudo corpus and a multi-task framework to capture invariant linguistic features, thus improving the parsing performance of low-resource languages. Choudhary and O'riordan (2023) incorporate linguistic typological knowledge into a multi-task learning framework to enhance cross-lingual knowledge transfer. However, these approaches often focus on capturing linguistic invariant features, ignoring language distributional bias and in-depth commonalities. As illustrated in Figure 1, both the target language Vietnamese and the source language Chinese share a "subject-predicate-object" main syntactic structure. But as shown by the red arrow, the Vietnamese usually adopt a post-modifiers pattern while the Chinese use a pre-modifiers pattern. Therefore, it is extremely important and challenging to excavate in-depth linguistic commonalities and filter beneficial knowledge from the source language (Yuan, Jiang, and Tu 2019; Sun, Li, and Zhao 2023; Huang et al. 2024; Sherborne, Hosking, and Lapata 2023).

To address these challenges, we propose novel dynamic syntactic feature filtering and injecting networks for cross-lingual dependency parsing. We first utilize the traditional shared-private model to yield original language-specific and language-invariant features by separated BiLSTM encoders. Then, a Language-Specific Filtering Network (LSFN) is applied in private encoders to emphasize useful source linguistic features and ignore detrimental ones. Simultaneously, we exploit a Language-Invariant Injecting Network (LIIN) on the shared encoder to enhance syntactic commonalities across languages by integrating the advantages of BiLSTM and improved Transformer encoders. Finally, we substitute the Multi-Layer Perception (MLP) with the Kolmogorov-Arnold Network (KAN) (Liu et al. 2024) to enrich syntax features more flexibly and dynamically. Experiments on seven benchmark datasets show that our model achieves average improvements of $1.84/3.43$ points in the UAS/LAS scores compared with the strong shared-private model, leading to new state-of-the-art results on all datasets. Comparison experiments demonstrate that LSFN can filter beneficial information from the source language to decrease language distributional bias and LIIN is helpful in excavating language-invariant features. Detailed analyses further prove these components complement each other with light parameters, greatly improving the cross-lingual dependency parsing performance.

Related Works

Cross-lingual dependency parsing leverages syntactic information from rich-resource languages to enhance parsing accuracy in low-resource languages (Zhang 2020; Xu et al. 2022; Zhao et al. 2024). Early approaches predominantly utilize projection-based techniques and annotation transfer (Xiao and Guo 2015). Xiao and Guo (2014) and Guo et al. (2015) employ distributed representations to map lexical features across languages, facilitating the capture of linguistic structures. Similarly, Tiedemann and Agić (2016) and Lacroix et al. (2016) introduce several parsers from partially annotated data through annotation projection. Despite their innovation, these methods are limited by the quality and availability of parallel data and the syntactic divergences between languages.

The emergence of Transformer architectures (Vaswani et al. 2017) and pre-trained language models (PLMs) like BERT (Devlin et al. 2019), XLM-RoBERTa (Conneau et al. 2020), and BART (Lewis et al. 2020) revolutionize cross-lingual dependency parsing. Researchers usually extract robust contextual representations from these PLMs to enhance traditional parsers or fine-tune PLMs for adapting parsing tasks. Kumar et al. (2022) use word-to-word dependency tagging features from BERT to enhance the malt parser, tackling data imbalance and consequently improving parsing results. Choenni, Garrette, and Shutova (2023) fine-tune mBERT assisted with language-specific subnetworks for controlling cross-lingual parameter sharing, improving low-resource languages parsing accuracy. However, the improvement in syntactic analysis for low-resource languages is still limited, since these PLMs are initially trained on a small-scale low-resource context (Haque, Liu, and Way 2021; Min et al. 2023).

Recent advancements in cross-lingual dependency parsing emphasize the transfer of explicit linguistic typology knowledge to reduce the interference of language differences and enhance generalization for low-resource languages (Choudhary and O'riordan 2023; Danilova and Stymne 2024; Kunz and Holmström 2024). Choudhary and O'riordan (2023) utilize multi-task Learning to transfer linguistic typology knowledge from the source language to enhance the target language parsing performance. Danilova and Stymne (2024) use topic modeling to assist cross-genre transfer and gain promising parsing performance. Despite prior researches achieving good results, excavating the implicit commonalities across different languages and mitigating distributional deviation still hinder current cross-lingual dependency parsing. Motivated by Wu et al. (2021) and Li et al. (2022), we propose dynamic syntactic feature filtering and injecting networks to filter beneficial information from the source language and extract deeper syntactic commonalities across different languages.

Our Approach

The typical shared-private model for cross-lingual dependency parsing employs a shared encoder to extract language-invariant features and multiple private encoders to capture language-specific features (Nishida and Matsumoto 2022b). However, this model treats all features equally and fails to construct linguistic differences and comprehensive linguistic links. Inspired by Li et al. (2022) and Gu et al. (2024), we propose dynamic syntactic feature filtering and injecting networks based on the shared-private model. On the one hand, a Language-Specific Filtering Network (LSFN) is applied to private encoders to emphasize useful source linguistic features and ignore irrelevant or detrimental ones. On the other hand, a Language-Invariant Injecting Network (LIIN) is exploited on the shared encoder to enhance syntactic commonalities across languages. Figure 2 presents the overall

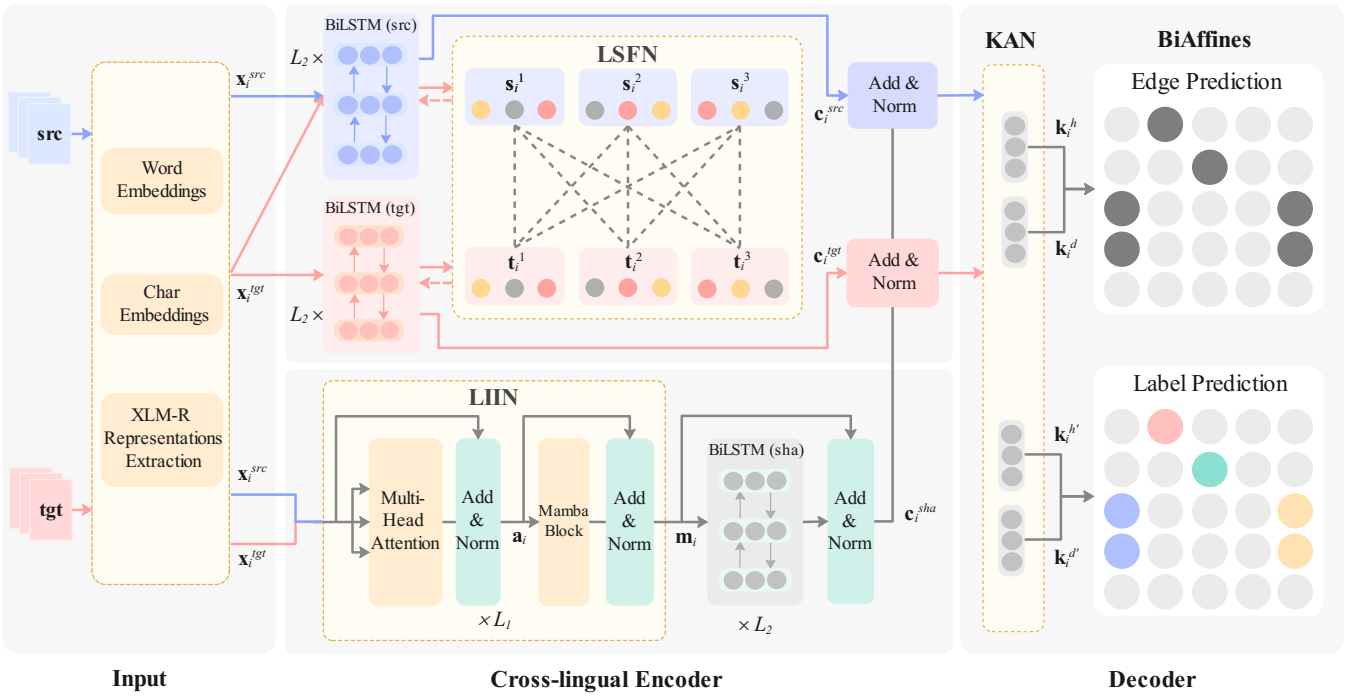

Figure 2: The architecture of our model, where "LSFN" and "LIIN" are the Language-Specific Filtering Network and the Language-Invariant Injecting Network. "KAN" stands for the Kolmogorov-Arnold Network. "L_1" and "L_2" means layer numbers. Dashed arrows indicate that the LSFN optimizes the private BiLSTM parameters. Solid lines in blue and red represent data flows from the source and target languages, while black solid lines represent shared data flows.

architecture of our proposed model, which is organized into three components, i.e., *Input*, *Cross-Lingual Encoder*, and *Decoder*.

Input Component

Given a sentence $w_1, w_2, \ldots, w_n$ either in source or target language, the input layer converts them into high-dimensional vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$. Different from the traditional shared-private model, we leverage the multi-language pre-trained language model (XLM-RoBERTa)[1] to enhance the representation capability of word vectors. As illustrated in Equation 1, each word vector $\boldsymbol{x}_i$ comprises its word representation and corresponding character representation $\mathbf{word}_i^{\texttt{char}}$. The word representation is the combination of the XLM-RoBERTa output $\mathbf{rep}_i^{\texttt{XLM-R}}$ and a randomly initialized word embedding $\mathbf{emb}_i^{\texttt{word}}$. The character representation $\mathbf{word}_i^{\texttt{char}}$ is produced by a Char-BiLSTM network, which utilizes a one-layer BiLSTM to encode the characters of each word w_i and merges the hidden vectors from two directions (Lample et al. 2016).

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

Cross-Lingual Encoder

The original shared-private model utilizes one shared and two private three-layer BiLSTMs to encode language-invariant and language-specific features. However, these BiLSTMs treat all syntactic features equally, thus limiting their ability to excavate the generality and discrepancy between source and target languages. To construct an in-depth relationship with the source language, we design a language-specific filtering network on private BiLSTMs to dynamically emphasize helpful source features and ignore the harmful ones. Meanwhile, we exploit a language-invariant injecting network on the shared BiLSTM to extract in-depth commonalities.

Language-Specific Filtering Network (LSFN). In the shared-private model, the input sentence from the source or target language is fed into its private BiLSTM to obtain language-specific contextual representation $\boldsymbol{c}_i^{src}$ or $\boldsymbol{c}_i^{tgt}$.

$$\begin{aligned} \boldsymbol{c}_i^{src} &= \text{BiLSTM}^{src}(\boldsymbol{x}_i^{src}, \theta_{\text{BiLSTM}^{src}}) \\ \boldsymbol{c}_i^{tgt} &= \text{BiLSTM}^{tgt}(\boldsymbol{x}_i^{tgt}, \theta_{\text{BiLSTM}^{tgt}}) \end{aligned} \quad (2)$$

Considering some source language-specific syntactic features may benefit the target language dependency parsing, we design a language-specific filtering network on private BiLSTMs to construct the in-depth relationship between source and target representations. Motivated by Li, Li, and Zhang (2022), we minimize the $\mathcal{L}_2$ distance between outputs of source and target private BiLSTMs to transfer useful information from source to target language, thus enriching the target language feature space.

$$\mathcal{L}_2 = \| f_\theta(\boldsymbol{t}_i^m) - \boldsymbol{s}_i^n \|_2^2 \quad (3)$$

where $f_\theta(\cdot)$ denotes a linear transformation, $\boldsymbol{s}_i^n$ is the

[1]https://huggingface.co/FacebookAI/xlm-roberta-base

n^{th}-layer BiLSTM^{src} output, and $\boldsymbol{t}_i^m$ is the m^{th}-layer BiLSTM^{tgt} output.

Concretely, each input sentence of the target language is first fed into both source and target private BiLSTMs to acquire target language-specific representation $\boldsymbol{t}_i$ and source representation $\boldsymbol{s}_i$. Then, our filtering network is applied to the private BiLSTMs to screen out harmful source information and emphasize useful ones by mimicking the well-trained source representations. For each filtering pair (n, m) between their private BiLSTMs are assigned learnable layer filtering weights $\mathbf{W}^{n,m}$ to determine the migration impact from BiLSTM^{src} to BiLSTM^{tgt}. Moreover, our model learns element filtering weight $\mathbf{E}_d^{n,m}$ to filter useful elements from source representations.

$$\begin{aligned} \mathbf{W}^{n,m} &= q_\phi^{n,m}(\boldsymbol{s}_i^n) \\ \mathbf{E}_d^{n,m} &= \text{softmax}\left(p_\phi^{n,m}(\boldsymbol{s}_i^n)\right)_d \end{aligned} \tag{4}$$

where $q_\phi^{n,m}$ and $p_\phi^{n,m}$ are a nonlinear and a linear transformation respectively. $\mathbf{E}_d^{n,m}$ is the non-negative weight of the d-th element in (n, m). $n, m \in \{1, 2, 3\}$ are the layer numbers of source and target BiLSTMs. Once the optimal layer matching weights are learned, element filtering weights are optimized. The filtering loss for the pair (n, m) is defined as follows,

$$\mathcal{L}^{fil} = \frac{1}{KD} \sum_{n,m} \mathbf{W}^{n,m} \sum_{d=1}^{D} \mathbf{E}_d^{n,m} \left(f_\theta(\boldsymbol{t}_i^m)_d - (\boldsymbol{s}_i^n)_d\right)^2 \tag{5}$$

where D is the output dimension of BiLSTMs, and $K = 3 \times 3 = 9$ denotes the total filtering pairs.

Language-Invariant Injecting Network (LIIN). In the shared-private model, sentences from both source or target languages are encoded by shared BiLSTM to obtain language-invariant contextual representation $\boldsymbol{c}_i^{sha}$. Considering self-attention is more suitable for capturing long-distance dependencies due to its capability of directly building connections between distant word pairs, while BiLSTM may fade the long-distance information in the encoding process, we design a language-invariant injecting network below the shared BiLSTM to compensate for the above shortcomings. Concretely, each LIIN layer is composed of a multi-head self-attention and a Mamba sub-layer. First, the multi-head attention layer computes pairwise correlations between word vectors, thus generating attention scores that weight contributions based on their relevance. Meanwhile, self-attention enables parallel processing across multiple subspaces, thus capturing syntactic knowledge of diverse aspects and intricate patterns within the input data. The formulas are defined as follows,

$$\begin{aligned} \textit{head}_i &= \text{Softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}_i \\ \boldsymbol{a}_i &= (\textit{head}_1, \ldots, \textit{head}_n)\mathbf{W}^0 \end{aligned} \tag{6}$$

where $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ are transformations of each word vector $\boldsymbol{x}_i$ by learnable matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$. $\mathbf{W}^0$ is a model parameter. Second, the Mamba sub-layer addresses limitations of static self-attention mechanisms by dynamically adjusting positional weights $\mathbf{A}$ and $\mathbf{B}$, which are generated based on the relevance of words $\boldsymbol{a}_i$ and their corresponding state vector $\boldsymbol{h}_i$. This adaptability enhances computational efficiency and improves the parsing of complex syntactic structures. The formulas are as follows,

$$\begin{aligned} \hat{\boldsymbol{h}}_i &= \mathbf{A}(\mathbf{B}\boldsymbol{a}_{i-1}) + \mathbf{B}\boldsymbol{a}_i \\ \boldsymbol{h}_i &= \mathbf{C}\hat{\boldsymbol{h}}_i \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{h}}_1 = \mathbf{B}\boldsymbol{a}_1$. $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ are adaptive weight matrices tailored to syntactic positions. Next, vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n$ are fed into the shared BiLSTM to learn contextual word information.

$$\boldsymbol{c}_i^{sha} = \text{BiLSTM}^{sha}(\boldsymbol{h}_i, \theta_{\text{BiLSTM}^{sha}}) \tag{8}$$

where θ_{BiLSTM} is the shared BiLSTM parameters. Finally, we obtain the cross-lingual encoder output $\boldsymbol{c}_i$ by adding $\boldsymbol{c}_i^{src}$ and $\boldsymbol{c}_i^{sha}$ for source language data, or $\boldsymbol{c}_i^{tgt}$ and $\boldsymbol{c}_i^{sha}$ for target language data.

$$\boldsymbol{c}_i = \begin{cases} \boldsymbol{c}_i^{sha} + \boldsymbol{c}_i^{src}, l = src \\ \boldsymbol{c}_i^{sha} + \boldsymbol{c}_i^{tgt}, l = tgt \end{cases} \tag{9}$$

where l is the language type of current data.

Decoder Component

Kolmogorov-Arnold Network (KAN). Unlike the shared-private model, we substitute the traditional Multi-Layer Perception (MLP) with KAN (Liu et al. 2024). The MLP aims to downscale contextualized word representations and capture syntactic features. However, its fixed activation function may limit the expressiveness and interpretability of syntax information extraction. To address this issue, we replace it with the KAN, which uses learnable nonlinear activation functions to replace MLP weights, thus enriching syntactic features flexibly. In general, the KAN takes contextualized word representation $\boldsymbol{c}_i$ as input and obtain its low-dimension head representations ($\boldsymbol{k}_i^h$ & $\boldsymbol{k}_i^{h'}$) and modifier representations ($\boldsymbol{k}_i^d$ & $\boldsymbol{k}_i^{d'}$).

$$\begin{aligned} \boldsymbol{k}_i^h, \boldsymbol{k}_i^d, \boldsymbol{k}_i^{h'}, \boldsymbol{k}_i^{d'} =& \text{KAN}_h(\boldsymbol{c}_i), \text{KAN}_d(\boldsymbol{c}_i), \\ & \text{KAN}_{h'}(\boldsymbol{c}_i), \text{KAN}_{d'}(\boldsymbol{c}_i) \end{aligned} \tag{10}$$

The formula of one-layer KAN is defined as follows,

$$\begin{aligned} \boldsymbol{k}_{i,b} &= \sum_{a=1}^{n_{\text{in}}} \sum_{i=1}^{n} \psi_{i,a,b}(\boldsymbol{c}_{i,a}) \\ \boldsymbol{k}_i &= (\boldsymbol{k}_{i,1}, \cdots, \boldsymbol{k}_{i,b}, \cdots \boldsymbol{k}_{i,n_{\text{out}}}) \end{aligned} \tag{11}$$

where n is the word number of the entered sentences. n_{in} and n_{out} are the input and output dimensions of KAN, respectively. $\boldsymbol{c}_{i,a}$ means the a-th dimension of input representation $\boldsymbol{c}_i$. $\boldsymbol{k}_{i,b}$ means the b-th dimension of output representation $\boldsymbol{k}_i$. $\psi_{i,a,b}$ represents a parameterized learnable nonlinear activation function that establishes the syntactic association between the a-th dimension of all input word vectors and the b-th dimension of each corresponding output word vector. Finally, each output vector $\boldsymbol{k}_i$ is obtained by concatenating all dimension representations $\boldsymbol{k}_{i,b}$ where $b \in \{1, 2, \ldots, n_{\text{out}}\}$.

Algorithm 1: Cross-lingual Training Procedure.

Input: Source language data S, target language data T
Parameter: Loss weight α, training iterations k.
Output: Result of parsing

```
Initialize iter = 0
while iter = k or convergence do
    Select mini-batch x alternately from S or T
    if x ∈ S then
        Compute parser loss L = L^par
        Update parser, LIIN parameters by minimizing L
    else if x ∈ T then
        Compute final loss L = L^par + αL^fil
        Update all parameters by minimizing L
    end if
    iter+ = 1
end while
```

BiAffine Layer. The dependency arc score between the modifier word w_j and its head word w_i is score$(i \leftarrow j)$ which is computed by a BiAffine operation. Simultaneously, the score of dependency label score$(i \overset{l}{\leftarrow} j)$ is calculated by another separated BiAffine operation as equation 12.

$$\begin{aligned} \text{score}(i \leftarrow j) &= \begin{bmatrix} \boldsymbol{k}_i^d \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{U}_1 \boldsymbol{k}_j^h \\ \text{score}(i \overset{l}{\leftarrow} j) &= \boldsymbol{k}_j^{h'} \mathbf{U}_2 \boldsymbol{k}_i^{d'} + (\boldsymbol{k}_j^{h'} \oplus \boldsymbol{k}_i^{d'}) \mathbf{U}_3 + b \end{aligned} \tag{12}$$

where $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_3$, and b are parameters. l denotes the relation label. The Maximum Spanning Tree (MST) algorithm is used to find the highest-score tree as the final parsing result.

Parser Loss. For each position i, if the gold-standard head word w_i modifies word w_j with relation label l, the parsing loss is computed as follows,

$$\begin{aligned} \mathcal{L}^{par} = &- \log \frac{e^{\text{score}(i \leftarrow j)}}{\sum\limits_{0 \le k \le n, k \ne i} e^{\text{score}(i \leftarrow k)}} \\ &- \log \frac{e^{\text{score}(i \overset{l}{\leftarrow} j)}}{\sum_{l' \in L} e^{\text{score}(i \overset{l'}{\leftarrow} j)}} \end{aligned} \tag{13}$$

where L refers to the collection of all dependency labels.

Cross-lingual Training

In this work, we propose a cross-lingual training strategy to take advantage of both source and target languages as Algorithm 1. We sample mini-batch x from source or target language data alternately. If x belongs to the source language S, we only update the parameters of the shared-private parser and LIIN by minimizing parsing loss. While x comes from the target language T, we update all parameters by minimizing parsing and filtering loss. Finally, we iteratively train all data until it converges or stops prematurely.

Experiments

Experimental Setups

Datasets. We conduct experiments on seven low-resource languages, i.e., Vietnamese (vi), Wolof (wo), Coptic (cop), Maltese (mt), Tamil (ta), Uyghur (ug), and Thai (th) where six languages are sourced from the Universal Dependencies (UD) v2.13 treebank[2], while the Thai dataset comes from the TUD corpus[3]. According to the similar language family, we select Chinese as the source language for Vietnamese, Tamil, Uyghur, and Thai, while English is the source language for Wolof, Coptic, and Maltese. Detailed dataset information is presented in Table 1.

Dataset	Train	Dev	Test	All
English (EWT)	12,544	2,001	2,077	16,622
Chinese (GSDSimp)	3,997	500	500	4,997
Vietnamese (VTB)	1,400	1,123	800	3,323
Wolof (WTB)	1,188	449	470	2,107
Coptic (Scriptorium)	1,419	381	403	2,203
Maltese (MUDT)	1,123	433	518	2,074
Tamil (TTB)	400	80	120	600
Uyghur (UDT)	1,656	900	900	3,456
Thai (TUD)	2,902	362	363	3,627

Table 1: Dataset statistics in sentence number.

Evaluation. We utilize Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS) as evaluation metrics (Hajic et al. 2009). All models are trained with no more than $1,000$ iterations, and their performances are evaluated on the development dataset after each iteration to guide the model selection. Model training is stopped if the peak performance does not increase for 20 consecutive iterations.

Hyper-parameter choices. We follow the most hyper-parameter settings of Li et al. (2019a), including MLP and BiAffine dimensions and learning rates. In addition, attention and Mamba components have 3 layers with a 0.5 dropout rate. The KAN uses a 0.33 dropout rate and its basic activation is initialized with the SiLU function. The loss weight α for the LSFN is set as 1.

Baselines. We reproduce the following baseline models for our comparative experiments.

- **Fully shared method (FulSha).** Peng, Thomson, and Smith (2017) utilize the fully shared encoder parameters to construct the commonality cross three dependency graph formalisms, thus enhancing heterogeneous dependency parsing performance. Here, we directly train the BiAffine parser with both source and target language data, which treats training data equally and shares all parameters between different languages.
- **Language embedding method (LanEmb).** Li et al. (2019b) use domain embeddings as an extra input to indicate the domain type of each word, which is proved effective for cross-domain dependency parsing. Motivated by this work, we also leverage 8 dimension language embeddings to guide the model to identify language types.
- **Shared-private method (ShaPri).** Wu et al. (2021) introduce a text-centred shared-private framework to capture shared semantic features and distinguish private ones

[2]https://universaldependencies.org/

[3]https://github.com/nlp-chula/TUD/tree/main/TUD

Model	vi		wo		cop		mt		ta		ug		th		avg.	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Results of previous works																
UDPipe(2019)	70.38	62.56	-	-	85.58	80.97	-	-	74.11	66.37	78.46	67.09	82.34	75.28	78.17	70.45
UDify(2019)	74.11	66.00	-	-	27.58	10.82	83.07	75.56	-	-	65.89	48.80	-	-	62.66	50.45
MBERT(2022)	-	-	-	72.94	-	82.11	-	72.69	-	54.94	-	42.97	-	-	-	65.13
ESR(2023)	-	60.80	-	73.30	-	-	-	74.20	-	66.40	-	39.20	-	-	-	62.78
Compare with baseline models																
FulSha	78.94	62.53	81.99	74.25	85.60	79.28	81.61	72.79	77.23	63.15	78.40	63.45	83.12	70.99	80.98	69.49
LanEmb	79.28	63.52	82.47	75.18	85.52	79.14	81.74	73.01	78.18	64.25	78.56	63.77	83.08	70.96	81.26	69.98
ShaPri	78.61	63.36	82.18	74.95	88.29	83.83	81.31	73.46	76.77	63.70	77.82	63.50	83.12	71.18	81.16	70.57
Our	**80.03**	**66.75**	**83.20**	**76.34**	**89.95**	**86.32**	**83.28**	**76.19**	**79.09**	**69.18**	**79.98**	**67.67**	**85.45**	**75.52**	**83.00**	**74.00**

Table 2: Main results of seven languages on the test dataset.

across modalities. Inspired by this work, we exploit a shared-private framework to capture language-invariant and language-specific features via separated BiLSTMs.

Experimental Results

Table 2 presents the final results of seven languages on the test dataset. First, we can see that the "LanEmb" model outperforms the "FulSha" model, indicating that extra language embeddings can help the parser effectively distinguish the source and target languages. Second, compared with "FulSha" and "LanEmb", the "ShaPri" model further improves the parsing accuracy, demonstrating that separated features can initially construct the commonalities and differences between different languages. Finally, our model achieves the best performance among all strong baselines, illustrating its effectiveness in meticulously capturing invariant syntactic features and distinguishing linguistic structural differences.

We also compare our model with several previous works. Straka, Straková, and Hajic (2019) present a UDpipe framework, which is jointly trained with tokenization, POS tagging, and dependency parsing sub-tasks on multiple languages to enhance parsing accuracy. Then, Kondratyuk and Straka (2019) propose a UDify framework which fine-tunes a multilingual BERT model in 104 languages to enhance parsing accuracy. Moreover, Gessler and Zeldes (2022) employ a vocabulary expansion method and fine-tune the BERT for parsing. Lastly, Effland and Collins (2023) adopt an expected statistic regularization that utilizes low-order multi-task structural statistics to shape model distributions to improve dependency parsing performance. Compared with these models, our approach achieves superior performance with only a single source language, highlighting its efficiency and robustness.

Model	Parameter (M)	Time (s)	UAS	LAS
Our model	310.4	81	**76.53**	**61.65**
w/o LSFN	304.9	69	75.58	60.90
w/o LIIN	306.5	79	75.81	60.30
w/o Two	301.0	67	75.04	59.52

Table 3: Ablation study on Vietnamese dev dataset.

Ablation Study

Table 3 shows the ablation study results on dev data. First, although our model introduces 9.4 million additional parameters, it only increases training time by 14 seconds and achieves a 2.13 LAS improvement, proving the efficiency and lightweight of our model. Second, we can see that removing the LSFN ("w/o LSFN") or LIIN ("w/o LIIN") component leads to an obvious decrease in parsing performance, demonstrating that each module is crucial for mitigating conflicts from direct language transfer and effectively helps our model to learn language commonalities and differences. Then, removing the LIIN and LSFN components simultaneously reduces dependency parsing accuracy significantly, indicating they are complementary and can benefit from each other. Finally, these observations emphasise the importance of retaining language-invariant features and reinforce the need for strategic filtering of source language-specific features.

Error Analysis

Sentence lengths. Figure 3 illustrates the LAS across various sentence lengths on Vietnamese dev data. All models perform better on shorter sentences. For sentences under 10

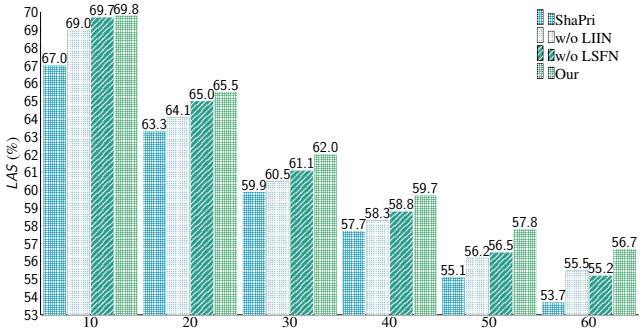

Figure 3: LAS regarding diverse sentence lengths.

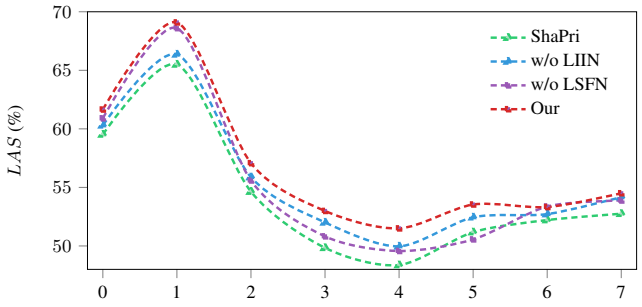

Figure 4: LAS curves regarding dependency distances.

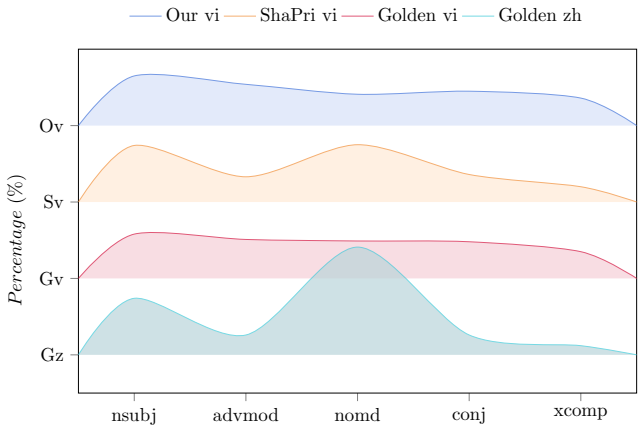

Figure 5: Dependency label distributional biases analysis.

words, the LAS ranges from 67 to 70. However, there is a significant decline of over 13 points for sentences around 60 words, highlighting the increased parsing difficulty with longer sentences. The "ShaPri" model consistently records the lowest scores across all length categories. In contrast, the "ShaPri" model with LSFN and LIIN achieves higher scores across all lengths, indicating their effectiveness in learning cross-lingual commonalities and filtering out irrelevant or harmful source language information. Finally, our model achieves the highest scores in all sentence lengths, demonstrating that our model greatly enhances short- and long-range syntactic dependency parsing capabilities.

Dependency Distances. Figure 4 presents LAS based on absolute dependency distances between head and modifier words on Vietnamese dev data. The "ShaPri" model consistently performs the lowest across most distances. On the contrary, "w/o LSFN" and "w/o LIIN" models acquire higher accuracies across most distances, suggesting their superior capability to capture language-invariant features and filter out language-specific noise. Our model shows significant improvements across all distances, highlighting the parsing ability of our model on both short- and long-range dependency distances.

Distributional biases. Figure 5 illustrates the dependency label distributions across various data sources, where the percentage is a certain label number divided by the total label number. First, the Chinese and Vietnamese golden UD

Model	Models with MLP		Models with KAN	
	UAS	LAS	UAS	LAS
FulSha	75.04	57.63	74.83	57.89
ShaPri	75.48	59.24	75.04	59.52
Our	**76.02**	**61.23**	**76.53**	**61.65**

Table 4: Comparative experiments between MLP and KAN on Vietnamese dev dataset.

test data, denoted by the solid blue and red curves respectively, exhibit notable distributional differences. For example, the "nmod" label percentage is 14.1 in Chinese versus 4.9 in Vietnamese. Then, the "ShaPri" model predicted label distribution for Vietnamese deviates substantially from the Vietnamese golden data distribution, instead approximating the Chinese golden data distribution. This is attributable to excessive transfer stemming from the resource imbalance between Chinese and Vietnamese data. Finally, our proposed method effectively mitigates this excessive transfer, resulting in a predicted Vietnamese label distribution that more closely aligns with the true Vietnamese distribution, thus demonstrating the validity and robustness of our model.

Effect on Different Syntax Extraction Strategies

Table 4 compares two different syntax extraction strategies on three models. First, KAN consistently achieves higher LAS scores than MLP across all models, indicating that KAN can more effectively capture complex syntactic features and patterns. Then, "FulSha" and "ShaPri" with KAN slightly improve the UAS scores, possibly since KAN associates each output word with all input words which may yield a small amount of noise or interference. Finally, our model with KAN achieves the best performance, further demonstrating that our model can filter out irrelevant and harmful information to help KAN extract syntax features more accurately. In a word, compared with MLP, KAN can extract syntax features more flexibly and comprehensively.

Conclusion

In this work, we propose dynamic syntactic feature filtering and injecting networks for cross-lingual dependency parsing, where the LSFN emphasizes helpful information from the source language and LIIN excavates commonalities across different languages simultaneously. Experiments on seven benchmark datasets exhibit that our efficient yet lightweight model consistently outperforms all strong baselines, leading to new state-of-the-art results on all datasets. Detailed comparative experiments confirm that both LSFN and LIIN can effectively transfer valuable knowledge from the source language to the target language and benefit from each other. Further analysis demonstrates that our model has outstanding capability to capture long sentences and remote dependencies. In addition, the comparison between MLP and KAN verifies that KAN can extract syntax features more flexibly and comprehensively.

Acknowledgments

This work is financially supported by the National Natural Science Foundation of China (62306129, U21B2027, 62366027, 62266028), Yunnan Fundamental Research Projects (202401CF070121, 202401BC070021, 202301AS070047), Yunnan Provincial Major Science and Technology Special Plan Projects (202103AA080015, 202202AD080003, 202203AA080004), Kunming University of Science and Technology "Double First-rate" Construction Joint Project (202301BE070001-027, 202201BE070001-021), Yunnan High and New Technology Industry Project (201606).

References

Ansell, A.; Ponti, E. M.; Pfeiffer, J.; Ruder, S.; Glavaš, G.; Vulić, I.; and Korhonen, A. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of EMNLP*, 4762–4781.

Choenni, R.; Garrette, D.; and Shutova, E. 2023. Cross-Lingual Transfer with Language-Specific Subnetworks for Low-Resource Dependency Parsing. *Computational Linguistics*, 49(3): 613–641.

Choudhary, C.; and O'riordan, C. 2023. Multilingual End-to-end Dependency Parsing with Linguistic Typology knowledge. In *In proceedings of SIGTYP*, 12–21.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, É.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of ACL*, 8440–8451.

Danilova, V.; and Stymne, S. 2024. Relation between Cross-Genre and Cross-Topic Transfer in Dependency Parsing. In Calzolari, N.; Kan, M.-Y.; Hoste, V.; Lenci, A.; Sakti, S.; and Xue, N., eds., *Proceedings of LREC-COLING 2024*, 13879–13884. Torino, Italia: ELRA and ICCL.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Dozat, T.; and Manning, C. D. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of ICLR*.

Effland, T.; and Collins, M. 2023. Improving Low-Resource Cross-lingual Parsing with Expected Statistic Regularization. *TACL*, 122–138.

Gessler, L.; and Zeldes, A. 2022. MicroBERT: Effective Training of Low-resource Monolingual BERTs through Parameter Reduction and Multitask Learning. In *Proceedings of ACL-MRL*, 86–99.

Gu, Y.; Hou, Y.; Wang, Z.; Duan, X.; and Li, Z. 2024. High-order Joint Constituency and Dependency Parsing. In *Proceedings of LREC-COLING*, 8144–8154.

Guo, J.; Che, W.; Yarowsky, D.; Wang, H.; and Liu, T. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of ACL-IJCNLP*, 1234–1244.

Hajic, J.; Ciaramita, M.; Johansson, R.; Kawahara, D.; Martí, M. A.; Màrquez, L.; Meyers, A.; Nivre, J.; Padó, S.; Štěpánek, J.; et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL*, 1–18.

Haque, R.; Liu, C.-H.; and Way, A. 2021. Recent advances of low-resource neural machine translation. *Machine Translation*, 35(4): 451–474.

Hou, Z.; Bi, S.; Qi, G.; Zheng, Y.; Ren, Z.; and Li, Y. 2024. Syntax-guided question generation using prompt learning. *Neural Computing and Applications*, 36(12): 6271–6282.

Huang, W.; Zhang, J.; Tian, T.; and Ji, D. 2024. A syntax-enhanced parameter generation network for multi-source cross-lingual event extraction. *Knowledge-Based Systems*, 292: 111585.

Kondratyuk, D.; and Straka, M. 2019. 75 Languages, 1 Model: Parsing Universal Dependencies Universally. In *Proceedings of EMNLP-IJCNLP*, 2779–2795.

Kumar, C. S. A.; Maharana, A.; Murali, S.; B, P.; and Kp, S. 2022. BERT-Based Sequence Labelling Approach for Dependency Parsing in Tamil. In *In proceedings of Dravidian-LangTech*, 1–8.

Kunz, J.; and Holmström, O. 2024. The Impact of Language Adapters in Cross-Lingual Transfer for NLU. In *Proceedings of MOOMIN*, 24–43.

Lacroix, O.; Aufrant, L.; Wisniewski, G.; and Yvon, F. 2016. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of NAACL-HLT*, 1058–1063.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT*, 260–270.

Lauscher, A.; Ravishankar, V.; Vulić, I.; and Glavaš, G. 2020. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of EMNLP*, 4483–4499.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of ACL*, 7871.

Li, Y.; Li, S.; and Zhang, M. 2022. Semi-supervised domain adaptation for dependency parsing with dynamic matching network. In *Proceedings of ACL*, 1035–1045.

Li, Y.; Li, Z.; Zhang, M.; Wang, R.; Li, S.; and Si, L. 2019a. Self-attentive Biaffine Dependency Parsing. In *Proceedings of IJCAI*, 5067–5073.

Li, Z.; Peng, X.; Zhang, M.; Wang, R.; and Si, L. 2019b. Semi-supervised Domain Adaptation for Dependency Parsing. In *Proceedings of ACL*, 2386–2395.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. KAN: Kolmogorov-Arnold Networks. arXiv:2404.19756.

Min, B.; Ross, H.; Sulem, E.; Veyseh, A. P. B.; Nguyen, T. H.; Sainz, O.; Agirre, E.; Heintz, I.; and Roth, D. 2023. Recent advances in natural language processing via large

pre-trained language models: A survey. *ACM Computing Surveys*, 56(2): 1–40.

Nguyen, L. 2020. Implementing Bi-LSTM-based deep biaffine neural dependency parser for Vietnamese Universal Dependency parsing. In *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*, 60–63.

Nishida, N.; and Matsumoto, Y. 2022a. Out-of-Domain Discourse Dependency Parsing via Bootstrapping: An Empirical Analysis on Its Effectiveness and Limitation. *Transactions of the Association for Computational Linguistics*, 10: 127–144.

Nishida, N.; and Matsumoto, Y. 2022b. Out-of-domain discourse dependency parsing via bootstrapping: An empirical analysis on its effectiveness and limitation. *TACL*, 10: 127–144.

Peng, H.; Thomson, S.; and Smith, N. A. 2017. Deep Multitask Learning for Semantic Dependency Parsing. In *Proceedings of ACL*, 2037–2048.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, 1532–1543.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of NAACL-HLT*, 2227–2237.

Ross, H.; Cai, J.; and Min, B. 2020. Exploring Contextualized Neural Language Models for Temporal Dependency Parsing. In *Proceedings of EMNLP*, 8548–8553.

Rotman, G.; and Reichart, R. 2019. Deep Contextualized Self-training for Low Resource Dependency Parsing. *TACL*, 7: 695–713.

Schuster, S.; Gupta, S.; Shah, R.; and Lewis, M. 2019. Cross-lingual Transfer Learning for Multilingual Task Oriented Dialog. In *Proceedings of NAACL-HLT*, 3795–3805.

Sherborne, T.; Hosking, T.; and Lapata, M. 2023. Optimal Transport Posterior Alignment for Cross-lingual Semantic Parsing. *TACL*, 11: 1432–1450.

Straka, M.; Straková, J.; and Hajic, J. 2019. UDPipe at SIGMORPHON 2019: Contextualized Embeddings, Regularization with Morphological Categories, Corpora Merging. In *Proceedings of SIGMORPHON*, 95–103.

Sun, K.; Li, Z.; and Zhao, H. 2023. Cross-lingual universal dependency parsing only from one monolingual treebank. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tang, C.; Qu, F.; and Wu, Y. 2024. Ungrammatical-syntax-based In-context Example Selection for Grammatical Error Correction. In *Proceedings of NAACL-HLT*, 1758–1770.

Tiedemann, J.; and Agić, Z. 2016. Synthetic treebanking for cross-lingual dependency parsing. *Journal of Artificial Intelligence Research*, 55: 209–248.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; K, K.; Mayhew, S.; and Roth, D. 2020. Extending Multilingual BERT to Low-Resource Languages. In *Findings of EMNLP*, 2649–2656.

Wu, Y.; Lin, Z.; Zhao, Y.; Qin, B.; and Zhu, L.-N. 2021. A Text-Centered Shared-Private Framework via Cross-Modal Prediction for Multimodal Sentiment Analysis. In *Findings of ACL-IJCNLP*, 4730–4738.

Xiao, M.; and Guo, Y. 2014. Distributed Word Representation Learning for Cross-Lingual Dependency Parsing. In *Proceedings of CoNLL*, 119–129.

Xiao, M.; and Guo, Y. 2015. Annotation Projection-based Representation Learning for Cross-lingual Dependency Parsing. In *Proceedings of CoNLL*, 73–82.

Xu, Y.; Cao, H.; Du, W.; and Wang, W. 2022. A survey of cross-lingual sentiment analysis: Methodologies, models and evaluations. *Data Science and Engineering*, 7(3): 279–299.

Yamin, M.; Sarno, R.; and Tambunan, T. 2024. Enhancing machine translation: Syntax and semantics-based word type and function extraction through multi-task transfer learning in Indonesian, Tolaki, and English. *Periodicals of Engineering and Natural Sciences*, 12(1): 223–235.

Yao, J.; Xue, N.; and Min, B. 2022. Modal Dependency Parsing via Language Model Priming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2913–2919.

Yuan, Y.; Jiang, Y.; and Tu, K. 2019. Bidirectional Transition-Based Dependency Parsing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33: 7434–7441.

Zhang, M. 2020. A survey of syntactic-semantic parsing based on constituent and dependency structures. *Science China Technological Sciences*, 63(10): 1898–1920.

Zhao, C.; Wu, M.; Yang, X.; Zhang, W.; Zhang, S.; Wang, S.; and Li, D. 2024. A Systematic Review of Cross-Lingual Sentiment Analysis: Tasks, Strategies, and Prospects. *ACM Computing Surveys*, 56(7): 1–37.