Graph Neural Networks for Syntax Encoding in Cross-Lingual Semantic Role Labeling

Anonymous ACL submission

Abstract

Recent models in cross-lingual semantic role labeling (SRL) barely consider the applicability of their network selection. They rely on LSTMs as their encoders, even though LSTMs do not transfer effectively to distant languages. We evaluate the effectiveness of different graph neural networks (GNNs) enriched with universal dependency trees, i.e., transformer-based, graph convolutional network-based, and graph attention network (GAT)-based models, and compare them with a BiLSTM-based model. We investigate which dependency-aware GNNs transfer best as an alternative encoder to LSTMs in cross-lingual SRL. We focus our study on a zero-shot setting by training the models in English and evaluating the models in 23 target languages in the Universal Proposition Bank. We consistently show that syntax from universal dependency trees is essential for cross-lingual SRL models to achieve better transferability. Dependency-aware self-attention with relative position representations (SAN-RPRs) transfer best across languages, especially in the long-range dependency distance. Furthermore, our proposed dependency-aware two-attention relational GATs perform better than SAN-RPRs in languages where most arguments lie in the $1-2$ dependency distance.

1 Introduction

Semantic role labeling (SRL) is a task to assign semantic roles to words or phrases in a sentence concerning a specific predicate, as shown in Figure 1. SRL supports many natural language processing (NLP) tasks, e.g., information extraction (Christensen et al., 2010; Stanovsky and Dagan, 2016), abstractive summarization (Khan et al., 2015), and machine translation (Rapp, 2022). However, SRL resource availability is still low, hindering the performance of other NLP tasks in diverse languages. Cross-lingual SRL models try to solve the problem by training the model in resource-rich languages and applying the model to resource-poor languages.

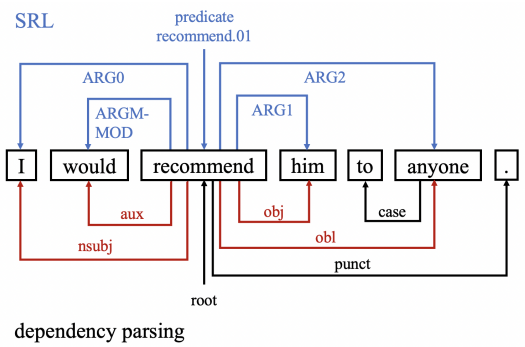


Figure 1: An example of an SRL task (top) and dependency parsing task (bottom) applied to a sentence taken from UPB. The red color indicates path intersections in both tasks.

Recent studies in cross-lingual SRL try to improve the model performance by separating language-universal and language-specific components (Fei et al., 2020; Conia et al., 2021) or minimizing dependence on external tools (Cai and Lapata, 2020). Nevertheless, these models barely consider the applicability of their network selection. They rely on LSTMs as their encoders, even though a study in cross-lingual dependency parsing (Ahmad et al., 2019) has stated that LSTMs do not transfer effectively to distant languages. LSTMs encode sentences sequentially, making them sensitive to word orders that vary across languages.

In this work, we investigate the effectiveness of various dependency-aware graph neural networks (GNNs) to find an alternative encoder for building cross-lingual SRL models. We encode universal dependency trees in dependency-aware GNNs, for the following reasons: (1) Many predicate-argument paths and argument roles in SRL intersect with dependency paths and dependency relations in dependency parsing (Marcheggiani and Titov, 2017), as shown in Figure 1. (2) Universal dependency tree representing a sentence's grammatical structure in a language-universal scheme provides a general representation across languages. (3) Universal de-

pendency trees help cross-lingual models achieve better transferability (Ahmad et al., 2021a; Ahmad et al., 2021b; Zhang et al., 2021).

We conduct comprehensive experiments on various networks as encoders, including transformer-based, graph convolutional network (GCN)-based, graph attention network (GAT)-based, and BiLSTM-based encoders. We choose to investigate transformer-based models because they have been proven effective in performing cross-lingually in dependency parsing (Ahmad et al., 2019) and event argument role labeling (EARL) (Ahmad et al., 2021b). Furthermore, we also investigate GCN-based and GAT-based models because different NLP tasks, e.g., monolingual SRL (Marcheggiani and Titov, 2017), aspect-based sentiment analysis (ABSA) (Wang et al., 2020; Jiang et al., 2021), and relation prediction (Nathani et al., 2019), have shown the effectiveness of exploiting the networks to encode dependency trees in their models.

Following previous work (Fei et al., 2020), we limit our exploration to argument detection and argument labeling in the dependency-based SRL. We conduct experiments in a zero-shot setting to find the most transferable network across languages. We train and evaluate the models in 23 languages provided by Universal Proposition Bank (UPB) v2. We show that: (1) Universal dependency trees are essential for cross-lingual SRL models to achieve better transferability. (2) Transformer-based model with dependency relation embedding (DRE) in the node representation and relative position representation (RPR) in the edge representation, i.e., SAN-RPRs, outperforms other models, especially as the dependency distance increases. (3) Two-attention relational GATs (TAGATs) with structural absolute position embedding (SAPE) in the node representation, and also dependency relation representation (DR) and RPR in the edge representation, perform best in languages where most arguments lie in the $1-2$ dependency distance.

2 Background

2.1 Universal Proposition Bank

Universal Proposition Bank (UPB) is a corpus containing SRL annotations for diverse languages. UPB v2 (Jindal et al., 2022) provides SRL annotations for 43 treebanks consisting of 23 languages, shown in Table 1. UPB is annotated semi-automatically through filtered annotation projection and bootstrap training (Akbik et al., 2015).

Target Languages in UPB v2		
Chinese (ZH)	Czech (CS)	Dutch (NL)
Finnish (FI)	Greek (EL)	Polish (PL)
Italian (IT)	Korean (KO)	Telugu (TE)
Spanish (ES)	Romanian (RO)	Indonesian (ID)
French (FR)	Hindi (HI)	Japanese (JA)
German (DE)	Marathi (MR)	Russian (RU)
Portuguese (PT)	Tamil (TA)	Ukrainian (UK)
	Hungarian (HU)	Vietnamese (VI)

Table 1: List of target languages available in UPB v2.

UPB v2 has significantly improved over UPB v1 regarding SRL annotation quality, language scope, and availability of span-based SRL annotations (Jindal et al., 2022). We use dependency-based SRL annotations in UPB v2 that are annotated according to UD v2.9 throughout our experiments.

2.2 Universal Dependencies

Universal Dependencies (UD) is a corpus containing consistent syntactic annotations for diverse languages, i.e., part-of-speech (POS) tags, morphological features, and dependency tree annotations. UD v1 (Nivre et al., 2016) and UD v2 (Nivre et al., 2020) have different annotation schemes[1] in terms of word segmentation, POS tags, morphological features, and syntactic relations. UD v1 and UD v2 have 40[2] and 37[3] universal dependency relations, respectively. UD v2.9 (Zeman et al., 2021a) contains dependency tree annotations for 217 treebanks of 122 languages.

2.3 Dependency-based Semantic Role Labeling

Instead of labeling the whole argument span with a semantic role, dependency-based SRL only labels the argument head, i.e., the head of the argument span according to the dependency tree. For example, in Figure 1, the phrase "to anyone" is the "ARG2" argument of the predicate "recommend". Based on the dependency tree at the bottom of the figure, "anyone" is the head of the phrase "to anyone". Therefore, dependency-based SRL annotates the edge that connects "recommend" to "anyone" with the "ARG2" argument.

2.4 Related Work

Recent cross-lingual SRL models encode sentences sequentially using BiLSTM-based models, even though LSTMs do not transfer effectively across

[1] https://universaldependencies.org/v2/summary.html
[2] https://universaldependencies.org/docsv1/u/dep/
[3] https://universaldependencies.org/u/dep/

languages. Fei et al. (2020) introduce a parameter generation network to vanilla BiLSTMs that isolates language-specific parameters from the universal parameters in BiLSTMs. Meanwhile, Cai and Lapata (2020) exploit BiLSTM-based semantic role labeler and compressor to build a cross-lingual SRL model and utilize parallel sentences to help the semantic role compressor achieve generalization. Finally, Conia et al. (2021) introduce a BiLSTM-based universal sentence encoder and predicate-argument encoder, and also language-specific decoders to build a universal SRL model.

GNNs have been used to encode dependency trees in building models for different NLP tasks, e.g., monolingual SRL, ABSA, EARL, and relation prediction. In monolingual SRL, Marcheggiani and Titov (2017) employ syntactic GCNs (SGCNs) on top of BiLSTMs to incorporate dependency trees as graphs. In ABSA, Wang et al. (2020) and Jiang et al. (2021) apply relational GATs (R-GATs) and attention-based relational GCNs (ARGCNs), respectively, over modified dependency trees to establish direct connections between aspects and their corresponding words. In EARL, Ahmad et al. (2021b) modify vanilla Transformers to encode syntactic structures from dependency trees, i.e., graph attention transformer encoders (GATEs). Finally, Nathani et al. (2019) propose KBGATs as a modification to original GATs (Veličković et al., 2018) to encode nodes and edge relations for relation prediction in knowledge graphs.

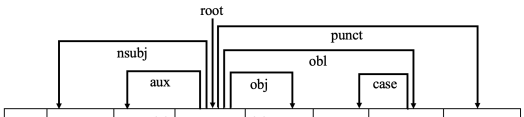
3 Model

3.1 Architecture

We apply an encoder-decoder architecture to compare transformer-based, GCN-based, GAT-based, and BiLSTM-based cross-lingual SRL models, consisting of an input layer, an encoder, and a decoder.

3.1.1 Input Layer

To produce the final word representation, h_i, for each word in a sentence, we optionally concatenate: (1) Predicate indicator embedding (PIE), p_i, represents whether a word is a predicate or not (Fei et al., 2020). (2) POS tag embedding (POSE), o_i, the POS tag of each word. (3) Absolute position embedding (APE), a_i, the position of each word in the sentence (Vaswani et al., 2017). (4) Structural absolute position embedding (SAPE), s_i, the dependency depth of each word relative to the root of the dependency tree (Wang et al., 2019b). (5)



	I	would	recom-mend	him	to	anyone	.
p_i	0	0	1	0	0	0	0
o_i	PRON	AUX	VERB	PRON	ADP	PRON	PUNCT
a_i	0	1	2	3	4	5	6
s_i	1	1	0	1	2	1	1
d_i	nsubj	aux	root	obj	case	obl	punct

Figure 2: The illustration of predicate indicator, POS tag, absolute position, structural absolute position, and dependency relation of each word in a sentence.

Dependency relation embedding (DRE), d_i, the dependency relation of each word as a dependent. (6) Contextualized multilingual word embedding, c_i, obtained from the concatenation of the last four hidden layers (Conia et al., 2021) in multilingual BERT (mBERT) (Devlin et al., 2019). Figure 2 shows the example to obtain each embedding. As shown in Equation 1, we apply a dropout to the concatenation result, and then a linear transformation following GATEs (Ahmad et al., 2021b).

$$h_i = W(\text{dropout}([p_i, o_i, a_i, s_i, d_i, c_i])) + b \quad (1)$$

3.1.2 Encoder

We experiment with transformer-based, GCN-based, GAT-based, and BiLSTM-based encoders. For GCN-based and GAT-based encoders, we apply an activation function, dropout, and residual connection in consecutive order after each layer. As proposed by GATs (Veličković et al., 2018), we can either apply or drop the activation function at the final layer. For the BiLSTM-based encoder, we only apply the consecutive operations in the last layer. Through our comprehensive experiments, we find the best combination of node and edge representation in each encoder, as shown in Table 2.

Transformer-Based Encoders In transformer-based encoders, we experiment with different modifications to vanilla Transformers, i.e., self-attention with APE in the node representation (**Trans**) (Vaswani et al., 2017), **GATEs** (Ahmad et al., 2021b), self-attention with RPR in the edge representation (**SAN-RPRs**) (Shaw et al., 2018), self-attention with SAPE in the node representation (**SAN-SAPRs**) (Wang et al., 2019b),

Model	Node Representation			Edge Representation		
	APE	SAPE	DRE	RPR	DR	SRPR
GATEs	✗	✓	✓	✗	✗	✗
Trans	✓	✗	✓	✗	✗	✗
SAN-RPRs	✗	✗	✓	✓	✗	✗
SAN-SAPRs	✗	✓	✗	✗	✗	✗
SAN-SRPRs	✗	✗	✗	✗	✗	✓
Trans-SRPRs	✓	✗	✓	✗	✗	✓
SAPR-RPRs	✗	✓	✗	✓	✗	✗
Trans-SRPR-DRs	✓	✗	✓	✗	B	✓
SAPR-RPR-DRs	✗	✓	✗	✗	B	✗
SGCNs	✗	✗	✓	✗	A	✗
R-GCNs	✗	✓	✗	✗	A	✗
ARGCNs	✗	✗	✓	✓	A	✗
GATs	✗	✓	✓	✗	✗	✗
S-HGNs	✗	✗	✗	✗	A	✗
TAGATs	✗	✓	✗	✓	A	✗
KBGATs	✗	✗	✓	✓	B	✗
BiLSTMs	✗	✓	✓	✗	✗	✗

Table 2: Best combination of embeddings in node and representations in edge for each model in transformer-based, GCN-based, and GAT-based models.

and self-attention with structural relative position representation (SRPR) in the edge representation (**SAN-SRPRs**) (Wang et al., 2019b). We also combine the node representation with the edge representation, i.e., SAN-SAPRs with SAN-RPRs (**SAPR-RPRs**) and Transformers with SAN-SRPRs (**Trans-SRPRs**). Following Ahmad et al. (2019), we take the absolute value when calculating RPR and SRPR to make the models more robust to word order differences. Furthermore, we experiment with incorporating dependency relation representation (DR) into the edge representation. We explain how to construct DR in Section 3.2.

GCN-Based Encoders In GCN-based encoders, we experiment with different types of GCNs, i.e., syntactic GCNs (**SGCNs**) (Marcheggiani and Titov, 2017), relational GCNs (**RGCNs**) (Schlichtkrull et al., 2018), and attention-based relational GCNs (**ARGCNs**) (Jiang et al., 2021). Since SGCNs and RGCNs encode the edge representation by differentiating their weight matrices based on the edge relations, they can only incorporate A-DR (refer to Section 3.2).

GAT-Based Encoders In GAT-based encoders, we experiment with different types of GATs, i.e., **GATs** (Veličković et al., 2018), simple heterogeneous GNNs (**SHGNs**) (Lv et al., 2021), relational GATs (RGATs) (Wang et al., 2020), and knowledge-based GATs (**KBGATs**) (Nathani et al., 2019). RGATs calculate the second attention weight, β, using position-wise feed-forward network (FFN) (Vaswani et al., 2017). However, we find that the original dot-product equation used to calculate the attention weight proposed by Veličković et al. (2018) works better for this task. Therefore, we modify RGATs into **TAGATs**. To calculate the second attention weight, β, TAGATs slightly modify Equation 3 in the original GAT paper (Veličković et al., 2018) to incorporate the DR, r_{ij}, as shown in Equation 2, where $\mathcal{N}_i$ is neighbor nodes of node i, W_r is a weight matrix to linearly transform the DR, r_{ij}, and LR is a Leaky ReLU. We provide a detailed explanation of TAGATs in Appendix C.

$$\beta_{ij} = \text{softmax}_{j \in \mathcal{N}_i}(\text{LR}(a^T[W_r r_{ij}])) \quad (2)$$

3.1.3 Decoder

We apply a linear scorer as the decoder. For each word, we concatenate sentence representation, h_s, predicate node representation, h_p, and node representation, h_i. h_s is obtained by applying max-pooling over node representations in the sentence (Ahmad et al., 2021b). Meanwhile, h_p is taken from the node representation of the sentence's predicate. After that, following GATEs (Ahmad et al., 2021b), we apply two feed-forward neural networks (FFNNs), each followed by a ReLU (RL), to produce the final node representation, h_f, with the dimension equal to the number of arguments, c, as shown in Equation 3.

$$h_f = \text{RL}(W_2(\text{RL}(W_1[h_s, h_p, h_i]+b_1))+b_2) \quad (3)$$

Finally, we apply a softmax function to produce the probability for each argument, z, as shown in Equation 4. We train the model to minimize the cross-entropy loss.

$$P(z) = \text{softmax}_z(h_f), \; z \in [1, c] \quad (4)$$

3.2 Graph Construction

First, we explain how to construct dependency relation representation (DR) to be encoded in the edges of the graphs. DR encodes the dependency direction, i.e., self-connection, head-to-dependent, and dependent-to-head, and the dependency relation between a pair of nodes. There are two ways of generating DR to be encoded in the edge representation of the graphs, i.e., A-DR and B-DR. In A-DR, following Marcheggiani and Titov (2017), we assign two completely different representations for edges with the same dependency relation but have different dependency directions (Appendix B). In B-DR, we first generate the representations

separately for each edge's dependency relation and direction with d_r and d_d dimensions, respectively. Then, we concatenate both representations to produce the DR with $(d_r + d_d)$ dimension.

In transformer-based models, we encode a sentence as a **fully-connected graph**. Shaw et al. (2018) modify the vanilla Transformers to incorporate RPR among the edges. We use the same approach to encode the DR by adding the DR to the RPR. Edges in the fully-connected graph that do not have the corresponding edges in the dependency tree do not have dependency relations. Therefore, we label these edges as "norel" (short for no relation) when constructing DRs.

In GCN-based and GAT-based models, we encode a sentence by forming a **dependency graph** based on its dependency tree. We follow the method proposed by Marcheggiani and Titov (2017) (Appendix B). They convert a dependency tree to a graph by adding edges that flow in the opposite direction of the original dependency direction and edges that flow from nodes to themselves. Furthermore, we encode either A-DR or B-DR in the edge representation.

4 Experiments

4.1 Corpus

We conduct experiments using corpus from UPB v2[4] (Jindal et al., 2022). UPB v2 contains SRL annotations based on dependency tree annotations in UD v2.9 (Zeman et al., 2021a). Some treebanks in UPB v2 have enhanced dependency tree annotations that cause new tokens (i.e., enhanced tokens) to be added to the sentences. The enhanced tokens cause some SRL annotations in UPB v2 to be shifted when merged with UD v2.9, resulting in the shifted predicate or semantic role annotations. Therefore, we run some preprocessing steps to fix the shifted annotation problem (Appendix A.2). When running our experiments, we merge all treebanks that belong to the same language.

4.2 Settings

We focus on conducting experiments in a zero-shot setting. We train the model in English and evaluate the model in 23 target languages provided by UPB v2, as shown in Table 1. In each experiment, we choose the final model from the epoch whose model performs best in the English validation set. However, we take the average F1 scores from the validation sets of 23 languages when choosing the best hyperparameter setting. We run exhaustive experiments to find the best hyperparameter setting to obtain the most transferable cross-lingual SRL model (Appendix D.2).

We use predicted dependency trees and POS tags for model evaluation (Appendix D.1). To obtain the predicted dependency trees and POS tags, we use pre-trained models trained on UD 2.8[5] (Zeman et al., 2021b) provided by Stanza (Qi et al., 2020).

We train the models for 100 epochs with 32 batch size. We use SGD optimizer (Kiefer and Wolfowitz, 1952) with a 0.1 learning rate. Following GATEs (Ahmad et al., 2021b), we apply early stopping if there is no improvement after 20 consecutive epochs. If the validation performance decreases, we decrease the learning rate by 10%. The training process will be stopped if the learning rate falls below 0.00001 after the decrement. We freeze mBERT as our contextualized word embedding to observe each model's effectiveness solely. Our preliminary experiments show that taking the average of subword embeddings from mBERT works best for our task. Therefore, we are going to apply this setting to our experiments. We report the average F1 scores from five runs with the standard deviation for model comparison.

4.3 Comparison Among Transformer-Based Models

Table 3 compares transformer-based models with the best hyperparameter setting. Trans-SRPR-DRs and SAPR-RPR-DRs perform worst among the transformer-based models. To obtain DR in the fully-connected graph, we label the edges that do not reflect the edges in the corresponding dependency tree as "norel". However, there are a lot more edges labeled as "norel" than edges that are labeled as the actual dependency relations. The imbalanced proportion of dependency relations might cause the model to overfit the limited actual dependency relations.

Furthermore, the table shows the superiority of SAN-RPRs over Trans, indicating that encoding the position of each word relative to another word in the edge representation produces a more general model than encoding the absolute position of

[4] https://github.com/UniversalPropositions

[5] We train the models from scratch for `Japanese-GSDLUW` and `French-Rhapsodie` treebanks because the pre-trained models are unavailable, and for `English-EWT` treebank because the SRL annotations are annotated based on an older version of UD, i.e., UD v2.5.

Model	PR	EN	AVG
GATEs	12.2M	78.96±0.31	52.57±0.23
Trans	12.2M	76.16±0.51	52.07±0.21
SAN-RPRs	12.2M	78.26±0.40	**52.73±0.40**
SAN-SAPRs	12.2M	75.93±0.47	51.98±0.10
SAN-SRPRs	12.1M	78.27±0.50	51.51±0.27
Trans-SRPRs	12.2M	79.03±0.32	52.21±0.30
SAPR-RPRs	12.2M	78.11±0.42	52.64±0.38
Trans-SRPR-DRs	12.2M	79.85±0.21	50.60±0.14
SAPR-RPR-DRs	12.2M	79.83±0.19	50.69±0.21
SGCNs	5.99M	79.94±0.27	**52.52±0.38**
R-GCNs	3.23M	78.28±0.29	51.48±0.35
ARGCNs	3.52M	77.84±0.44	52.13±0.32
GATs	5.09M	79.81±0.19	52.66±0.14
S-HGNs	5.06M	78.84±0.35	52.61±0.26
TAGATs	6.31M	79.07±0.19	**52.78±0.14**
KBGATs	7.73M	79.53±0.31	52.31±0.32

Table 3: F1 scores (%) of transformer-based, GCN-based, GAT-based models evaluated on UPB v2 test set with predicted parsers. AVG indicates the average F1 scores of a specific model evaluated in target languages. The bold score and underlined score indicate the highest and second-highest scores in each group.

each word in the sentence in the node representation. This finding aligns with the results in Ahmad et al. (2019). On the other hand, SAN-SAPRs outperform SAN-SRPRs, indicating that encoding the dependency distance of each word relative to the root of the dependency tree in the node representation produces a more general model than encoding each word's dependency distance relative to another word in the edge representation. We further combine the information regarding the position of each word according to the sentence and the sentence's dependency tree, i.e., SAPR-RPRs and Trans-SRPRs. Consistent with the previous results, SAPR-RPRs, consisting of features from SAN-SAPRs and SAN-RPRs, outperform Trans-SRPRs, consisting of features from Trans and SAN-SRPRs.

According to the average F1 score, SAN-RPRs and SAPR-RPRs are strong models with 52.73% and 52.64% average F1 scores, respectively, outperforming GATEs (Ahmad et al., 2021b), i.e., the model proposed for cross-lingual EARL. Furthermore, SAN-RPRs which employ DRE in the node representation, perform better than SAPR-RPRs. Nevertheless, SAPR-RPRs perform best among transformer-based models in more languages than SAN-RPRs (Appendix E.1.1). We take both models for comparison with other best models in Section 4.6.

4.4 Comparison Among GCN-Based Models

Table 3 compares GCN-based models with the best hyperparameter setting. SGCNs significantly outperform the other GCN-based models, i.e., RGCNs and ARGCNs, with a 52.52% average F1 score. RGCNs perform the worst among GCN-based models because RGCNs are the only network not applying the attention mechanism. ARGCNs apply a self-attention mechanism, while SGCNs realize the attention mechanism in the form of a gating mechanism. Self-attention or gating mechanism measures how much attention each node should pay to other nodes when the network updates each node's representation. Those mechanisms help emphasize the edges in the dependency graph that intersect with the predicate-argument paths.

4.5 Comparison Among GAT-Based Models

Table 3 compares GAT-based models with the best hyperparameter setting. TAGATs perform better than SHGNs indicating that the GAT-based model learns better when we separate the attention weight calculation based on node representations and edge representations. Moreover, GATs also perform better than SHGNs indicating that encoding the SAPE and DRE in the node representation is a better way to encode dependency features than combining node representations with edge representations when calculating the attention weight. According to the average F1 score, TAGATs and GATs are strong models with 52.78% and 52.66% average F1 scores, respectively. Therefore, we compare both models with other best models in Section 4.6.

On the other hand, according to the average F1 score, KBGATs perform worst among GAT-based models. The significant difference between KBGATs and the others is that KBGATs update each node representation with (1) neighbor node representations and (2) surrounding edges' representations that contain information about DR and RPR. Meanwhile, the other models update each node representation only with (1). We conjecture that the approach of KBGATs might overpopulate each node in every update with information too specific to a particular language the network learns from.

4.6 Comparison Among Best Models

We compare the best models from each group, i.e., SAN-RPRs, SAPR-RPRs, SGCNs, GATs, and TAGATs, with the BiLSTM-based model, i.e., BiLSTMs, in Table 4. We calculate each model's supe-

	$\mathbf{d \geq 3}$	**SAN-RPRs**	**SAPR-RPRs**	**SGCNs**	**GATs**	**TAGATs**	**BiLSTMs**
EN	2.68	78.26±0.40	78.11±0.42	79.94±0.27	79.81±0.19	79.07±0.19	76.86±0.34
AVG	-	52.73±0.40	52.64±0.38	52.52±0.38	52.66±0.14	**52.78±0.14**	51.85±0.09
TA	17.18	37.96±1.75	**39.57±1.18**	34.32±1.12	35.08±0.58	35.68±1.28	34.19±1.22
HI	8.46	47.51±0.62	45.04±0.35	48.24±0.61	**48.25±0.33**	47.65±0.33	46.63±0.38
ZH	8.41	50.37±1.17	**50.96±0.88**	45.77±0.67	46.12±0.39	46.80±0.64	47.56±0.89
JA	8.11	37.69±0.99	34.78±1.29	37.43±0.28	37.99±0.52	**39.30±0.73**	37.40±0.61
VI	7.89	28.69±0.79	**29.10±0.45**	27.95±0.56	28.06±0.59	28.31±0.55	28.18±0.88
KO	6.88	42.61±1.84	**45.24±1.23**	42.92±0.64	43.22±0.56	44.57±0.24	41.77±1.61
ID	5.52	58.78±1.09	**59.97±0.53**	58.54±0.82	58.33±0.69	59.11±0.87	56.11±0.84
HU	5.38	49.76±0.35	49.08±0.34	50.76±0.41	**51.10±0.51**	50.90±0.37	50.64±0.39
RO	5.32	54.23±0.67	**54.46±0.52**	53.57±0.47	54.12±0.49	53.60±0.45	53.26±0.34
FR	4.55	**62.19±0.41**	62.11±0.47	60.93±0.38	61.64±0.44	61.13±0.22	61.22±0.27
MR	4.08	**41.06±2.89**	40.36±2.20	40.97±3.40	38.06±0.13	39.26±1.20	37.18±2.28
UK	4.06	58.92±0.26	59.36±0.72	59.66±0.76	59.49±0.56	**59.72±0.31**	58.96±0.07
PT	3.75	66.05±0.21	**66.49±0.33**	65.62±0.43	65.99±0.32	65.61±0.15	64.40±0.33
IT	3.73	**58.11±0.33**	57.80±0.39	57.43±0.42	58.00±0.42	57.34±0.34	58.02±0.27
ES	3.67	63.71±0.33	63.62±0.25	63.87±0.61	**64.29±0.36**	63.91±0.27	62.48±0.29
CS	3.66	56.87±0.27	55.80±0.51	57.95±0.52	**58.02±0.21**	57.62±0.28	56.59±0.36
EL	3.59	60.59±0.23	60.23±0.40	60.56±0.69	60.74±0.48	**60.86±0.34**	59.76±0.45
FI	3.35	**55.58±0.42**	55.29±0.54	54.87±0.40	54.62±0.32	54.88±0.20	54.62±0.20
RU	3.07	60.18±0.44	**61.13±0.50**	59.98±0.34	60.14±0.16	60.30±0.22	59.73±0.25
NL	3.05	62.84±0.37	62.22±0.58	62.94±0.21	**63.53±0.64**	62.97±0.37	62.47±0.36
TE	2.49	44.66±2.00	43.88±1.57	46.08±1.07	**46.96±1.49**	**46.96±1.82**	45.95±0.51
DE	2.46	56.86±0.32	56.98±1.13	**58.61±0.30**	58.52±0.26	58.52±0.18	57.72±0.23
PL	1.71	57.67±0.36	57.28±0.46	**59.08±0.31**	59.00±0.44	58.92±0.52	57.73±0.23
SC	-	13	18	9	16	15	1
PR	-	12.2M	12.2M	5.99M	5.09M	6.31M	9.03M

Table 4: F1 scores (%) of best models evaluated on UPB v2 test set with predicted parsers. The bold score and underlined score indicate the highest and second-highest scores. AVG indicates the average F1 scores of a specific model evaluated in target languages. PR and SC are the number of parameters and the superiority score of each model. $d \geq 3$ column indicates the proportion of gold arguments (%) that fall in ≥ 3 dependency distance. We use predicted dependency trees to measure the dependency distance.

riority score (SC) based on the model performance in target languages. We allocate 2 points if the model achieves the highest F1 score or 1 point if the model achieves the second-highest F1 score for a specific language.

TAGATs have the best average F1 score among the models with 52.78% average F1 score. However, SAPR-RPRs and GATs perform best among the models in slightly more languages, indicated by the higher SCs. Despite having the second-best average F1 score of 52.73%, SAN-RPRs have a lower SC than SAPR-RPRs, GATs, and TAGATs. Overall, transformer-based and GAT-based models outperform SGCNs, indicating that the self-attention mechanism is better at emphasizing the essential dependency paths than the gating mechanism. Finally, BiLSTMs perform the worst as the only network that encodes sentences sequentially.

According to hyperparameter search, stacking two layers of GAT-based encoders performs best. However, this approach has a drawback, as the information can only travel as far as two hops from the origin node. On the other hand, stacking three layers of transformer-based encoders does not affect the traveling distance of the information. In transformer-based models, we construct a fully-connected graph of a sentence allowing the information to travel from one node to every other node despite how many layers are stacked together.

The second column in Table 4 shows the percentage of arguments in ≥ 3 dependency distance, i.e., the number of hops from the predicate node to a specific node according to the sentence's dependency tree. Some languages like TA, HI, ZH, JA, and VI have a relatively high number of arguments ($> 7\%$) in ≥ 3 dependency distance. Transformer-based models perform significantly better in TA ($> 3\%$) and ZH ($> 4\%$), slightly better in VI ($< 1\%$), and slightly worse in HI ($< 1\%$) and JA ($< 2\%$). This evidence proves that transformer-based models are generally better in long-range dependency distance than GAT-based models. On the other hand, some languages have a relatively low number of arguments ($< 3\%$) in ≥ 3 dependency distance, i.e., EN, TE, DE, and PL. For these languages, where most of the arguments lie in the

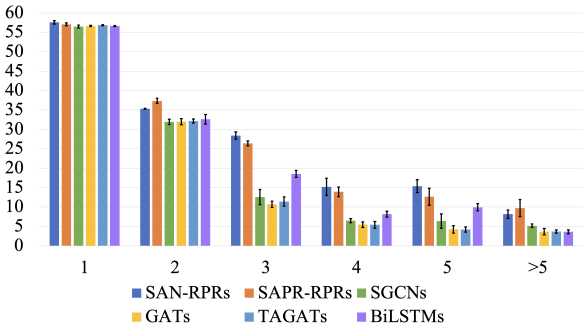


Figure 3: Average F1 scores (%) of best models grouped by the dependency distance evaluated on UPB v2 test set with predicted parsers.

Model	SAN-RPRs	SAPR-RPRs
base	**52.73**$\pm$**0.40**	**52.64**$\pm$**0.38**
w/o POSE	**52.73**$\pm$**0.32**	51.91$\pm$0.46
w/o PIE	51.54$\pm$0.36	51.14$\pm$0.43
w/o SAPE	-	51.65$\pm$0.28
w/o DRE	51.65$\pm$0.28	-

Table 5: Average F1 scores (%) of SAN-RPRs and SAPR-RPRs with certain embedding removed from the node representation evaluated on UPB v2 test set with predicted parsers.

$1-2$ dependency distance, GAT-based models perform better than transformer-based models.

Figure 3 shows the average F1 scores of each model grouped by the dependency distance. Although BiLSTMs perform worst among the models, BiLSTMs are better in long-range dependency distance than GCN-based and GAT-based models. The figure also shows the superiority of transformer-based models as the dependency distance increases, even when compared to BiLSTMs.

4.7 Ablation Study

We conduct ablation studies for the best transformer-based models, i.e., SAN-RPRs and SAPR-RPRs, as shown in Table 5. We experiment with removing DRE, POSE, or PIE from the node representation in SAN-RPRs. Removing either DRE or PIE from the node representation reduces the performance of SAN-RPRs. However, SAN-RPRs without POSE perform better in most languages (Appendix E.1.2).

Furthermore, we also experiment with removing SAPE, POSE, or PIE from the node representation in SAPR-RPRs. Removing SAPE, POSE, or PIE from the node representation reduces the performance of SAPR-RPRs. Unlike SAN-RPRs, SAPR-RPRs do not have DRE in their node representation. We conjecture that the combination of POSE and SAPE in SAPR-RPRs is necessary to replace the role of DRE.

In Table 4, according to the SC, we can see that SAPR-RPRs perform best in more languages than SAN-RPRs, even though the average F1 score of SAN-RPRs is better than SAPR-RPRs. However, as discussed, SAN-RPRs without POSE perform better in most languages than SAN-RPRs with POSE, which we use for comparison in Table 4. Therefore, we re-compare SAN-RPRs without POSE with the other best models in Table 4. After removing POSE from SAN-RPRs, we find that the model performs best in more languages than SAPR-RPRs (Appendix E.2.1).

5 Conclusions and Future Work

Through comprehensive experiments, we consistently show that incorporating syntax from dependency trees can improve the transferability of cross-lingual SRL models across languages. Overall, we show that the transformer-based model, i.e., SAN-RPRs that encode DRE without POSE in the node representation and RPR in the edge representation, stacked in three layers, performs the best among all models, especially as the dependency distance increases. However, TAGATs that encode SAPE in the node representation, and also DR and RPR in the edge representation, stacked in two layers, perform better than SAN-RPRs in languages where most of the arguments lie in the $1-2$ dependency distance.

In the future, we can extend our model to incorporate language-specific components and modify the objective function to maximize the learning of universal features without ignoring the specific features that appear in each language. This can be useful if we want to extend the model training to a few-shot setting where we include a certain proportion of target sentences in the training set.

Limitations

The limitation of this work is that we focus on argument detection and argument labeling in cross-lingual SRL, assuming that the sentences' gold predicates are easy to obtain. Furthermore, we focus on conducting experiments in a zero-shot setting. The availability of target sentences in the training set might affect the models' behavior, which should be investigated further.

Ethical Statement

We believe there is no ethical issue raised in this work. SRL is a low-level task to support other advanced NLP applications. Therefore, increasing the coverage of SRL models in various languages is beneficial for developing NLP tools to help solve the problems in this diverse society.

References

Wasi Ahmad, Haoran Li, Kai-Wei Chang, and Yashar Mehdad. 2021a. Syntax-augmented multilingual BERT for cross-lingual transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4538–4554, Online. Association for Computational Linguistics.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021b. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12462–12470.

Alan Akbik, Laura Chiticariu, Marina Danilevsky, Yunyao Li, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2015. Generating high quality proposition Banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 397–407, Beijing, China. Association for Computational Linguistics.

Rui Cai and Mirella Lapata. 2020. Alignment-free cross-lingual semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3883–3894, Online. Association for Computational Linguistics.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2010. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 52–60, Los Angeles, California. Association for Computational Linguistics.

Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026, Online. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1243–1252. JMLR.org.

Aleksa Gordić. 2020. pytorch-gat. `https://github.com/gordicaleksa/pytorch-GAT`.

Junfeng Jiang, An Wang, and Akiko Aizawa. 2021. Attention-based relational graph convolutional network for target-oriented opinion words extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1986–1997, Online. Association for Computational Linguistics.

Ishan Jindal, Alexandre Rademaker, Michał Ulewicz, Ha Linh, Huyen Nguyen, Khoi-Nguyen Tran, Huaiyu Zhu, and Yunyao Li. 2022. Universal Proposition Bank 2.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1700–1711, Marseille, France. European Language Resources Association.

Atif Khan, Naomie Salim, and Yogan Jaya Kumar. 2015. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, 30:737–747.

Jack Kiefer and Jacob Wolfowitz. 1952. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, page 1150–1160, New York, NY, USA. Association for Computing Machinery.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Reinhard Rapp. 2022. Using semantic role labeling to improve neural machine translation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3079–3083, Marseille, France. European Language Resources Association.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019a. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019b. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409, Hong Kong, China. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkadur Barkarson, Rodolfo Basile, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Juyeon Chung, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Jannatul Ferdaousi, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Siratun Jannat, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Elena Klyachko, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Mehmet Köse, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Sandra Kübler, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phûông Lê Hông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Stefano Lusito, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Menel Mahamdi, Jean Maillard, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Lorena Martín-Rodríguez, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Tatiana Merzhevich, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lûông Nguyên Thị, Huyên Nguyên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayò Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Mizanur Rahoman, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Mathilde Regnault, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Putri Rizqiyah, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf

Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Syeda Shahzadi, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurð'sson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Shafi Sourov, Carolyn Spadine, Rachele Sprugnoli, Steinþór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Chihiro Taguchi, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Dipta Tanaya, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Sri Hartati Wijono, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Arlisa Yuliawati, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, He Zhou, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021a. Universal dependencies 2.9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, Hórunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkadur Barkarson, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon. Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phùòng Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher

Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lùòng Nguyên Thị, Huyên Nguyên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayó Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurd'sson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Rachele Sprugnoli, Steinþór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Taksum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021b. Universal dependencies 2.8. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2021. On the benefit of syntactic supervision for cross-lingual transfer in semantic role labeling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6229–6246, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lang	Train	Dev	Test
English (EN)	12,542	1,974	2,062
Chinese (ZH)	3,997	500	500
Czech (CS)	102,993	11,311	12,203
Dutch (NL)	18,078	1,394	1,472
Finnish (FI)	27,198	3,239	3,422
French (FR)	17,968	2,970	1,712
German (DE)	166,849	19,233	19,436
Greek (EL)	1,662	403	456
Hindi (HI)	13,304	1,659	1,684
Hungarian (HU)	910	441	449
Indonesian (ID)	4,482	559	557
Italian (IT)	29,685	2,277	2,518
Japanese (JA)	14,100	1,014	1,086
Korean (KO)	27,410	3,016	3,276
Marathi (MR)	373	46	47
Polish (PL)	31,496	3,960	3,942
Portuguese (PT)	16,633	2,376	2,367
Romanian (RO)	35,911	2,247	2,272
Russian (RU)	19,894	1,525	1,482
Spanish (ES)	28,474	3,054	2,147
Tamil (TA)	400	80	120
Telugu (TE)	1,051	131	146
Ukrainian (UK)	5,496	672	892
Vietnamese (VI)	1,400	800	800

Table 6: Number of sentences available in each language in UPB v2.

A Artifacts

A.1 Corpus Distribution

Table 6 shows the corpus distribution in UPB v2. Since we run our experiments in a zero-shot setting, we only use the dev set and test set for languages other than English.

A.2 Corpus Preprocessing

Some treebanks have enhanced dependency tree annotations that cause new tokens to be added to the sentences. These tokens are called enhanced tokens. The enhanced tokens cause some SRL annotations in UPB v2 to be shifted when merged with UD v2.9, resulting in the wrong predicate or semantic role annotations. For example, look at the example of wrong predicate annotation taken from the dev set in `Finnish-TDT` (UPB v2). Token 10.1 is an enhanced token.

```
# sent_id = w063.9
# text = Osasto.........
1 _ _ _
2 _ _ _
3 be.01 A1:2|AM-LOC:7
A1:1-2|AM-LOC:6-7
4 _ _ _
5 _ _ _
6 _ _ _
7 _ _ _
8 _ _ _
9 _ _ _
10 _ _ _
10.1 _ _ _
11 be.01 A1:9 A1:9-10
12 _ _ _
13 _ _ _
```

The corresponding annotation in UD v2.9 is as follows. Note that we only present the tokenized words, lemmas, and POS tags here since we only present the UD annotation to highlight the shifted annotation problem.

```
# sent_id = w063.9
# text = Osasto N7 sijaitsee
  samassa korttelissa
  Naistenklinikan rakennuksessa
  ja osasto LV37 Kätilöopiston
  sairaalassa.
1 Osasto osasto NOUN
2 N7 N7 SYM
3 sijaitsee sijaita VERB
4 samassa sama PRON
5 korttelissa kortteli NOUN
6 Naistenklinikan nais#klinikka
  NOUN
7 rakennuksessa rakennus NOUN
8 ja ja CCONJ
9 osasto osasto NOUN
10 LV37 LV37 SYM
10.1 sijaitsee sijaita VERB
11 Kätilöopiston kätilö#opisto
   NOUN
12 sairaalassa sairaala NOUN
13 . . PUNCT
```

If we compare the two annotations from UPB v2 and UD v2.9, we can see that the first predicate annotated on token 3, i.e., “sijaitsee”, is correct. However, the second predicate annotated on the token 11, i.e., “Kätilöopiston”, is wrong. The correct second predicate is token 10.1, i.e., “sijaitsee”. The annotation is somehow shifted because of the enhanced token added, token 10.1. Therefore, we fix the annotation in UPB v2 to be as follows.

```
# sent_id = w063.9
# text = Osasto.........
1 _ _ _
2 _ _ _
3 be.01 A1:2|AM-LOC:7
A1:1-2|AM-LOC:6-7
4 _ _ _
5 _ _ _
```

```
6 _ _ _
7 _ _ _
8 _ _ _
9 _ _ _
10 _ _ _
10.1 be.01 A1:9 A1:9-10
11 _ _ _
12 _ _ _
13 _ _ _
```

In some cases, not only the predicate annotations indicated in the third column but also the semantic role annotations indicated in the fourth column are shifted. We run a script to fix the annotation problems in all treebanks with enhanced dependency tree annotations. The treebanks with enhanced dependency tree annotations are `Czech-CAC`, `Czech-FicTree`, `Czech-PDT`, `Dutch-Alpino`, `Dutch-LassySmall`, `Finnish-TDT`, `Italian-ISDT`, `Spanish-AnCora`, and `Ukrainian-IU`.

After we fix the shifted predicate and semantic role annotations, we notice that some predicates and their semantic roles are annotated in the enhanced tokens that do not appear in the original sentence. The treebanks that contain this phenomenon are `Dutch-Alpino`, `Dutch-LassySmall`, `Finnish-TDT`, `Ukrainian-IU`, and `Spanish-AnCora`. We cannot accommodate these annotations since we build the models based on the sentence's original tokens. Therefore, we omit the predicates and their corresponding semantic roles annotated on enhanced tokens in our experiments.

A.3 License

Complete UPB v2 contains annotations from UD v2.9. Table 7 shows the license for each treebank in UD v2.9. Despite licenses inherited from UD v2.9, UPB v2 also has a CDLA-Sharing-1.0 license.

We refer to publicly available codes to build the corpus and models for experiments. We provide the list of GitHub repositories with their corresponding licenses, as follows.

1. UniversalPropositions/tools: Apache-2.0
2. diegma/neural-dep-srl (Marcheggiani and Titov, 2017): Apache-2.0
3. AnWang-AI/towe-eacl (Jiang et al., 2021): No License
4. dmlc/dgl (Wang et al., 2019a): Apache-2.0
5. gordicaleksa/pytorch-GAT (Gordić, 2020): MIT
6. deepakn97/relationPrediction (Nathani et al., 2019): No License
7. thudm/hgb (Lv et al., 2021): No License
8. shenwzh3/RGAT-ABSA (Wang et al., 2020): MIT
9. wasiahmad/GATE (Ahmad et al., 2021b): MIT

We access all the resources we mentioned above solely for academic research. We make sure that we obey the intended usage of each artifact.

Treebank	License
English-EWT	CC BY-SA 4.0
Chinese-GSD	CC BY-SA 4.0
Czech-CAC	CC BY-SA 4.0
Czech-CLTT	CC BY-SA 4.0
Czech-FicTree	CC BY-NC-SA 4.0
Czech-PDT	CC BY-NC-SA 3.0
Dutch-Alpino	CC BY-SA 4.0
Dutch-LassySmall	CC BY-SA 4.0
Finnish-FTB	CC BY 4.0
Finnish-TDT	CC BY-SA 4.0
French-GSD	CC BY-SA 4.0
French-Rhapsodie	CC BY-SA 4.0
French-Sequoia	LGPL-LR
German-GSD	CC BY-SA 4.0
German-HDT	CC BY-SA 4.0
Greek-GDT	CC BY-NC-SA 3.0
Hindi-HDTB	CC BY-NC-SA 4.0
Hungarian-Szeged	CC BY-NC-SA 3.0
Indonesian-GSD	CC BY-SA 4.0
Italian-ISDT	CC BY-NC-SA 3.0
Italian-ParTUT	CC BY-NC-SA 4.0
Italian-PoSTWITA	CC BY-NC-SA 4.0
Italian-TWITTIRO	CC BY-SA 4.0
Italian-VIT	CC BY-NC-SA 3.0
Japanese-GSD	CC BY-SA 4.0
Japanese-GSDLUW	CC BY-SA 4.0
Korean-GSD	CC BY-SA 4.0
Korean-Kaist	CC BY-SA 4.0
Marathi-UFAL	CC BY-SA 4.0
Polish-LFG	GNU GPL 3.0
Polish-PDB	CC BY-NC-SA 4.0
Portuguese-Bosque	CC BY-SA 4.0
Portuguese-GSD	CC BY-SA 4.0
Romanian-Nonstandard	CC BY-SA 4.0
Romanian-RRT	CC BY-SA 4.0
Romanian-SiMoNERo	CC BY-SA 4.0
Russian-GSD	CC BY-SA 4.0
Russian-Taiga	CC BY-SA 4.0
Spanish-AnCora	CC BY 4.0
Spanish-GSD	CC BY-SA 4.0
Tamil-TTB	CC BY-NC-SA 3.0
Telugu-MTG	CC BY-SA 4.0
Ukrainian-IU	CC BY-NC-SA 4.0
Vietnamese-VTB	CC BY-SA 4.0

Table 7: License for each treebank in UD v2.9.

Treebank	Dev			Test		
	$\mathbf{F1_{POS}}$	UAS	LAS	$\mathbf{F1_{POS}}$	UAS	LAS
English-EWT	96.79	92.46	90.86	96.80	91.42	89.82
Chinese-GSD	95.35	85.11	83.19	95.52	87.06	85.13
Czech-CAC	99.26	92.97	91.62	98.70	93.43	91.68
Czech-CLTT	99.44	89.13	86.98	98.98	88.32	86.09
Czech-FicTree	98.43	94.68	93.11	98.34	94.61	92.76
Czech-PDT	98.77	93.74	92.24	98.63	93.50	91.87
Dutch-Alpino	98.36	94.53	92.24	97.33	92.87	90.42
Dutch-LassySmall	97.03	90.77	87.62	96.31	92.12	89.11
Finnish-FTB	96.90	93.77	92.29	96.87	94.03	92.41
Finnish-TDT	98.08	91.97	90.41	97.78	92.24	90.74
French-GSD	98.45	95.66	94.45	98.20	93.47	91.87
French-Rhapsodie	98.12	87.75	83.25	97.64	86.42	81.88
French-Sequoia	99.03	93.54	92.23	99.12	93.10	91.70
German-GSD	96.19	91.78	88.61	95.37	89.65	85.62
German-HDT	98.08	95.18	93.64	98.30	95.30	93.72
Greek-GDT	97.74	91.77	90.43	97.71	92.93	91.19
Hindi-HDTB	97.89	96.62	94.49	97.93	96.68	94.43
Hungarian-Szeged	96.66	87.64	84.10	96.06	86.72	83.25
Indonesian-GSD	94.64	86.49	76.25	94.73	87.31	77.33
Italian-ISDT	98.54	94.41	92.84	98.62	94.37	93.16
Italian-ParTUT	97.86	92.76	90.52	98.54	93.10	91.40
Italian-PoSTWITA	97.35	87.21	83.20	96.96	88.33	84.41
Italian-TWITTIRO	96.79	87.25	81.64	96.20	84.85	79.77
Italian-VIT	98.12	90.63	88.82	98.16	91.54	89.05
Japanese-GSD	98.34	96.09	95.47	98.10	95.11	94.21
Japanese-GSDLUW	98.54	96.12	95.82	98.58	95.35	95.12
Korean-GSD	95.79	88.22	85.41	96.27	89.65	87.07
Korean-Kaist	96.19	91.35	90.39	95.58	90.41	89.45
Marathi-UFAL	89.32	74.55	64.32	90.53	79.85	70.63
Polish-LFG	98.94	97.56	96.73	99.05	97.80	96.92
Polish-PDB	98.75	94.17	92.69	98.74	94.58	93.16
Portuguese-Bosque	97.92	94.25	92.51	98.10	94.85	93.54
Portuguese-GSD	98.36	94.44	93.34	98.28	94.21	93.23
Romanian-Nonstandard	96.77	93.18	90.04	96.40	91.43	87.75
Romanian-RRT	98.06	91.96	88.60	97.92	91.93	88.45
Romanian-SiMoNERo	98.19	93.38	91.21	98.23	93.78	91.86
Russian-GSD	98.38	90.55	87.80	98.09	90.44	87.21
Russian-Taiga	95.80	83.94	79.32	97.06	84.42	81.41
Spanish-AnCora	98.99	93.83	92.16	98.96	93.82	92.00
Spanish-GSD	97.13	91.91	89.79	97.26	91.93	89.58
Tamil-TTB	87.17	81.24	73.48	86.93	80.89	72.30
Telugu-MTG	94.41	92.90	86.25	94.45	93.07	85.58
Ukrainian-IU	98.08	91.14	89.34	97.67	90.10	88.24
Vietnamese-VTB	92.84	78.92	74.99	92.81	77.58	74.16

Table 8: $\mathrm{F1_{POS}}$, UAS, and LAS of each treebank's POS tagger and dependency parser in UPB v2. $\mathrm{F1_{POS}}$ indicates the F1 score of the POS tagger.

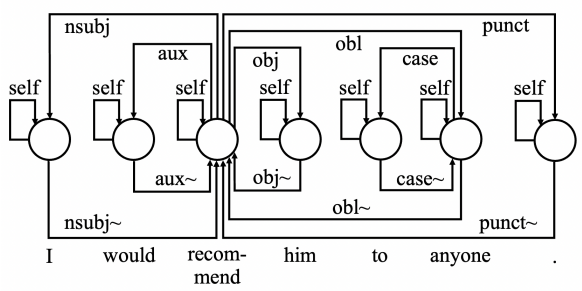


Figure 4: Dependency graph of a sentence converted from its dependency tree.

B Dependency Tree to Dependency Graph

Marcheggiani and Titov (2017) convert a dependency tree to a dependency graph in the following steps:

1. Add the edges from the dependency tree to the graph. These edges flow from the heads to the dependents. Label each edge with the corresponding dependency relation from the dependency tree.

2. Add new edges that flow in the opposite directions of the original dependency directions. These new edges flow from the dependents to the heads.

3. Assign a unique dependency relation, derived from the original dependency relation, to each new edge added in step 2. For example, if the original dependency relation is "nsubj", the edge that flows in the opposite direction is labeled as "nsubj∼".

4. Add new edges that flow from each node to itself (self-connection) and label them as "self".

Figure 4 displays the dependency graph derived from the dependency tree at the bottom of Figure 1.

C Two-Attention Relational Graph Attention Networks

Similar to RGATs (Wang et al., 2020), two-attention relational GATs (TAGATs) also apply two types of attention weights to measure the influence of neighbor nodes when updating the corresponding node representation. However, instead of using position-wise FFN to calculate the second attention weight as in RGATs (Wang et al., 2020), two-attention relational GATs (TAGATs) apply dot-product equation proposed by Veličković et al. (2018) to calculate both attention weights. We explain the modification that we made to RGATs in this section.

TAGATs calculate the first attention weight, α, using Equation 3 in the original GAT paper (Veličković et al., 2018). To calculate the second attention weight, β, TAGATs slightly modify the equation to incorporate the DR, r_{ij}, as shown in Equation 5, where k is the current attention head, l is the current layer, $\mathcal{N}_i$ is the neighbor nodes of node i, W_r is a weight matrix to linearly transform the DR, r_{ij}, and LR is a Leaky ReLU.

$$\beta_{ij}^{l,k} = \text{softmax}_{j \in \mathcal{N}_i}(\text{LR}(a^{l,k^T}[W_r^{l,k} r_{ij}^l])) \quad (5)$$

Furthermore, TAGATs obtain node representation from attention weight α using Equation 5 and Equation 6 in the original GAT paper (Veličković et al., 2018). To obtain node representation from attention weight β, TAGATs employ Equation 6 and Equation 7. TAGATs concatenate node representations from K heads in the intermediate layers, as shown in Equation 6. Meanwhile, in the final layer, TAGATs take the average of node representations from K heads, as shown in Equation 7, where L is the number of layers.

$$h_{i,\beta}^{l+1} = \sigma(||_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \beta_{ij}^{l,k} W^{l,k} h_j^l), \; l < L \quad (6)$$

$$h_{i,\beta}^{l+1} = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \beta_{ij}^{l,k} W^{l,k} h_j^l), \; l = L \quad (7)$$

Finally, TAGATs calculate the node representation for layer $(l+1)$, h_i^{l+1}, by applying a linear transformation to the concatenation of node representation $h_{i,\alpha}^{l+1}$ and node representation $h_{i,\beta}^{l+1}$ and optionally apply an activation function, σ, on top of the linear transformation, as shown in Equation 8.

$$\begin{aligned} x_i^{l+1} &= [h_{i,\alpha}^{l+1}, h_{i,\beta}^{l+1}] \\ h_i^{l+1} &= \sigma(W_{l+1} x_i^{l+1} + b_{l+1}) \end{aligned} \quad (8)$$

D Experiments

D.1 Dependency Parsers and POS Taggers

Table 8 shows the POS tagger and dependency parser evaluation results on each treebank in UPB v2. For `Japanese-GSDLUW`,

Hyperparameter	Value
num_epochs	100
batch_size	32
optimizer	SGD
learning_rate	0.1
num_early_stop	20
num_decay_epoch	5
lr_decay	0.9
min_lr	0.00001
pos_dim	30
pred_ind_dim	30
emb_dropout	0.5
hid_dim	512
num_heads	8
d_k	64
d_v	64
d_ff	2048

Table 9: Basic hyperparameters applied in the experiments.

French-Rhapsodie, and English-EWT treebanks, we train the POS taggers and dependency parsers from scratch using Stanza (Qi et al., 2020) with a 0.0005 learning rate, 70,000 max steps, and 10,000 max steps before stopping. We measure the performance of POS taggers with the F1 score. Meanwhile, we measure the performance of dependency parsers with the unlabeled attachment score (UAS) and the labeled attachment score (LAS).

D.2 Hyperparameter Search

We take the representative models from transformer-based models (i.e., SAN-RPRs), GCN-based models (i.e., ARGCNs and SGCNs), GAT-based models (i.e., GATs), and BiLSTM-based model (i.e., BiLSTMs), to experiment with the optimizers. We experiment with SGD (Kiefer and Wolfowitz, 1952), Adam (Kingma and Ba, 2015), and AdamW (Loshchilov and Hutter, 2019) as the optimizer. We also try different learning rates for each optimizer, i.e., 0.1, 0.01, and 0.001. The SGD optimizer with a 0.1 learning rate works the best in all the representative models. Therefore, we apply this setting to the rest of our experiments. Table 9 summarizes the fixed hyperparameters we use throughout our experiments. Algorithm 1 shows the logic for model training. Below, we explain each hyperparameter:

1. num_epochs: Number of epochs for model training.
2. batch_size: Batch size for model training.
3. optimizer: Type of optimizer for model training.
4. learning_rate: Initial learning rate for model training.
5. num_early_stop: Stop the training if there is no improvement after a certain number of consecutive epochs.
6. num_decay_epoch: The upper limit of epoch before we start decaying the learning rate.
7. lr_decay: The ratio to decay the learning rate.
8. min_lr: Minimum learning rate allowed. We stop the model training if the learning rate falls below this threshold.
9. pos_dim: Dimension of POSE, o (Equation 1). If the dimension is 0, we will not concatenate o in the input layer.
10. pred_ind_dim: Dimension of PIE, p (Equation 1). If the dimension is 0, we will not concatenate p in the input layer.
11. emb_dropout: Dropout applied in the input layer (Equation 1).
12. hid_dim: The dimension of the node representation that the encoder accepts.
13. num_heads: Number of heads applied in the multi-head self-attention mechanism present in transformer-based models, GAT-based models, and ARGCNs.
14. d_k: The dimension of keys applied in transformer-based models.
15. d_v: The dimension of values applied in transformer-based models.
16. d_ff: The output dimension of the first linear transformation in transformer-based model's position-wise FFN.

The following sections will explain the hyperparameter search and hyperparameter values that work best in each model. Table 11, Table 12, Table 13, and Table 10 describe the hyperparameter search for transformer-based models, GCN-based models, GAT-based models, and BiLSTM-based

Algorithm 1 Pseudocode of the model training.

```
Require: num_early_stop, num_decay_epoch, min_lr, lr_decay, num_epochs, learning_rate
  best_f1 ← 0
  no_improvement ← 0
  for curr_epoch ← 1, num_epochs do
     curr_f1 ← train(learning_rate)
     if curr_f1 > best_f1 then
        best_f1 ← curr_f1
        no_improvement ← 0
     else
        no_improvement ← no_improvement + 1
        if no_improvement ≥ num_early_stop then
           break
        end if
        if curr_epoch > num_decay_epoch then
           learning_rate ← lr_decay * learning_rate
           if learning_rate < min_lr then
              break
           end if
        end if
     end if
  end for
```

models, respectively. Due to the number of hyperparameters, we divide the hyperparameter search into groups indicated by the leftmost column, i.e., the column with a "No" header. We will search for the best combination between the hyperparameters in the same group. For example, in Table 11, `num_enc_layers` and `enc_dropout` belong to group 1, which means we experiment with different dropouts, i.e., 0.1, 0.2, 0.3, 0.4, and 0.5, for each number of layers, i.e., 1, 2, 3, and 4. Below, we explain each hyperparameter involved in the hyperparameter search.

1. `num_enc_layers`: Number of layers stacked together.

2. `enc_dropout`: Dropout applied in the models, including dropout applied in the encoder (Section 3.1.2).

3. `lstm_num_layers`: Number of BiLSTM layers stacked together.

4. `lstm_dropout_net`: Dropout applied in BiLSTMs, including dropout applied in the encoder (Section 3.1.2).

5. `gnn_activation`: Activation function used in GCN-based and GAT-based models, applied in the encoder (Section 3.1.2).

6. `gnn_activation_at_final_layer`: Whether to apply the activation function in the last layer, especially if we stack more than one layer.

7. `lstm_activation`: Activation function used in BiLSTMs (Section 3.1.2).

8. `lstm_hidden_size`: Hidden size of BiLSTMs. Since the network is bidirectional, the dimension of the final hidden representation is $2 \times$ `lstm_hidden_size`.

9. `deprel_dim`: Dimension of DRE, d (Equation 1). If the dimension is 0, we will not concatenate d in the input layer.

10. `abs_position_dim`: Dimension of APE, a, and SAPE, s (Equation 1).

11. `use_dep_abs_position`: Boolean value which indicates whether to concatenate SAPE, s, in the input layer (Equation 1).

12. `use_word_abs_position`: Boolean value which indicates whether to concatenate APE, a, in the input layer (Equation 1).

13. `att_dim`: Dimension of a trainable vector a in ARGCNs (Equation 7 in Jiang et al. (2021)).

14. `base_size`: Base size, B, in RGCNs (Equation 3 in Schlichtkrull et al. (2018)).

15. `rel_pos_dim`: Dimension of RPR in GCN-based and GAT-based models. We do not use this parameter for transformer-based models because the RPR in transformer-based models must have the same dimension as d_k and d_v (Vaswani et al., 2017).

16. `num_embed_graph_heads`: Number of heads where we modify M matrix according to distance matrix, D, in GATEs (Equation 3 in Ahmad et al. (2021b)). We apply a zero matrix for M for the other heads to connect all the nodes in the graph.

17. `max_tree_dists`: The δ parameter applied to each head in GATEs (Equation 3 in Ahmad et al. (2021b)). The length of this parameter must equal `num_embed_graph_heads`.

18. `max_relative_positions`: The maximum absolute value for relative position (k in Shaw et al. (2018)) or structural relative position (r in Section Wang et al. (2019b)).

19. `use_dep_rel_pos`: The boolean value indicates whether to incorporate SRPR in the edge representation.

20. `use_word_rel_pos`: The boolean value indicates whether to incorporate RPR in the edge representation.

21. `deprel_edge_dim`: Dimension of representation for dependency relation, d_r, when we use B-DR.

22. `deparc_edge_dim`: Dimension of representation for direction, d_d, when we use B-DR.

23. `deprel_ext_edge_dim`: Dimension of dependency relation representation when we use A-DR.

There are two ways of generating APE, SAPE, RPR, and SRPR, i.e., using learned positional embedding (Gehring et al., 2017) and using sine and cosine functions (Vaswani et al., 2017). We conduct preliminary experiments and find that sine and cosine functions work better than learned positional embedding. Therefore, we generate the representation for each type of position in the experiments using sine and cosine functions.

D.2.1 Transformer-Based Models

Table 11 shows the hyperparameter search in transformer-based models.

D.2.2 GCN-Based Models

Table 12 shows the hyperparameter search in GCN-based models.

No	Hyperparameter	Value
1	`lstm_num_layers` `lstm_dropout_net`	1, 2, 3, **4** 0.1, 0.2, **0.3**, 0.4, 0.5
2	`lstm_activation`	**ReLU**, Leaky ReLU, ELU
3	`lstm_hidden_size`	**256**, 512

4	`deprel_dim`
	`abs_position_dim`
	`use_dep_abs_position`
		0, **30**
		0, **30**
		T

Table 10: Hyperparameter search in BiLSTM-based models. The search is divided into groups shown in the "No" header. We search for the best combination of hyperparameters in the same group. The bold values indicate the results of the hyperparameter search.

D.2.3 GAT-Based Models

Table 13 shows the hyperparameter search in GAT-based models.

D.2.4 BiLSTM-Based Models

Table 10 shows the hyperparameter search in BiLSTM-based models.

D.3 Computational Resource

We use Tesla P100 to train the models. Training time for GCN-based and GAT-based models takes around 5 hours. Meanwhile, training time for BiLSTM-based and transformer-based models takes around 10 hours. Hyperparameter search in GCN-based models costs around 775 GPU hours. Hyperparameter search in GAT-based models costs around 910 GPU hours. Hyperparameter search in transformer-based models costs around 1,720 GPU hours. Hyperparameter search in BiLSTM-based models costs around 290 GPU hours. After searching for the best hyperparameter setting for each model, we run the training five times for each model, spending around 1,075 GPU hours. Therefore, in total, we spend around 4,770 hours.

E Supporting Results

E.1 Transformer-Based Models

E.1.1 Comparison

Table 14 shows the detailed comparison of transformer-based models in each language. We calculate the superiority score (SC) of each model based on the model performance in target languages. We allocate 2 points if the model achieves the highest F1 score or 1 point if the model achieves the second-highest F1 score for a specific language.

No	Hyperparameter	Value
General		
1	num_enc_layers enc_dropout	1, 2, **3**, 4 0.1, **0.2**, 0.3, 0.4, 0.5
Transformers		
1	deprel_dim	0, **30**
GATEs		
1	num_embed_graph_heads max_tree_dists	4 <1, 1, 2, 2>, <2, 2, 4, 4>, <**4, 4, 8, 8**>, <1, 2, 4, 8>
2	deprel_dim abs_position_dim <use_dep_abs_position, use_word_abs_position>	0, **30** 0, **30** <**T, F**>, <F, T>
SAN-RPRs		
1	max_relative_positions	1, 2, 4, 8, **16**
2	deprel_dim	0, **30**
SAN-SAPRs		
1	deprel_dim	**0**, 30
SAN-SRPRs		
1	max_relative_positions	1, **2**, 4, 8, 16
2	deprel_dim	**0**, 30
Trans-SRPRs		
1	deprel_dim	0, **30**
Trans-SRPR-DRs		
1	deprel_dim abs_position_dim use_word_abs_position use_dep_rel_pos <deprel_edge_dim, deparc_edge_dim, deprel_ext_edge_dim>	0, **30** 0, **30** T **T**, F <32, 32, 0>, <48, 16, 0>, <56, 8, 0>, <**60, 4, 0**>, <62, 2, 0>, <63, 1, 0> <0, 0, 64>
SAPR-RPRs		
1	deprel_dim	**0**, 30
SAPR-RPR-DRs		
1	deprel_dim abs_position_dim use_dep_abs_position use_word_rel_pos <deprel_edge_dim, deparc_edge_dim, deprel_ext_edge_dim>	**0**, 30 0, **30** T T, **F** <32, 32, 0>, <48, 16, 0>, <56, 8, 0>, <**60, 4, 0**>, <62, 2, 0>, <63, 1, 0> <0, 0, 64>

Table 11: Hyperparameter search in transformer-based models. The search is divided into groups shown in the "No" header. We search for the best combination of hyperparameters in the same group. The bold values indicate the results of the hyperparameter search.

No	Hyperparameter	Value
SGCNs		
1	num_enc_layers enc_dropout	1, 2, **3**, 4 0.1, **0.2**, 0.3, 0.4, 0.5
2	gnn_activation gnn_activation_at_final_layer	**ReLU**, Leaky ReLU, ELU T, **F**
3	deprel_dim abs_position_dim <use_dep_abs_position, use_word_abs_position>	0, **30** **0**, 30 <T, F>, <F, T>
RGCNs		
1	num_enc_layers enc_dropout	**1**, 2, 3, 4 0.1, 0.2, 0.3, 0.4, **0.5**
2	gnn_activation gnn_activation_at_final_layer	**ReLU**, Leaky ReLU, ELU **T**, F
3	base_size	1, **2**, 4, 8, 16, 32, 80
4	deprel_dim abs_position_dim <use_dep_abs_position, use_word_abs_position>	**0**, 30 0, **30** <**T, F**>, <F, T>
ARGCNs		
1	num_enc_layers enc_dropout	**1**, 2, 3, 4 0.1, 0.2, 0.3, 0.4, **0.5**
2	gnn_activation gnn_activation_at_final_layer	**ReLU**, Leaky ReLU, ELU **T**, F
3	deprel_ext_edge_dim att_dim rel_pos_dim use_word_rel_pos	1, 2, 4, **8**, 16 1, 2, 4, 8, **16** **64**, 128 T
4	deprel_dim abs_position_dim use_dep_abs_position	0, **30** **0**, 30 T

Table 12: Hyperparameter search in GCN-based models. The search is divided into groups shown in the "No" header. We search for the best combination of hyperparameters in the same group. The bold values indicate the results of the hyperparameter search.

E.1.2 Ablation Study of SAN-RPRs

We experiment with removing either DRE, d, POSE, o, or PIE, p, from the node representation in SAN-RPRs. Table 15 shows the results of the ablation study. Removing either DRE or PIE from the node representation reduces the performance of SAN-RPRs. However, SAN-RPRs without POSE perform better in most languages.

E.2 Best Models

E.2.1 Comparison with SAN-RPRs w/o POSE

In Table 4, according to the superiority score, we can see that SAPR-RPRs perform best in more languages than SAN-RPRs, even though the average F1 score of SAN-RPRs is better than SAPR-RPRs. However, as discussed before, SAN-RPRs without POSE perform better in most languages than SAN-RPRs with POSE, which we use for comparison in Table 4. Therefore, in Table 16, we re-compare SAN-RPRs without POSE with the other best models. After removing POSE from SAN-RPRs, we find that the model performs best in more languages than SAPR-RPRs with 18 and 16 superiority scores, respectively.

E.2.2 Fine-Tuned Models

We fine-tune the contextualized word embedding in SAN-RPRs and SAPR-RPRs, i.e., multilingual BERT (mBERT). Table 17 compares the average F1 scores of models with frozen and fine-tuned mBERT. Overall, fine-tuning increases the performance of both models. However, in the fine-tuned models, the variability of the average F1 score in each run increases, indicated by the higher standard deviation in fine-tuned mBERT. This is expected as when we fine-tune the mBERT, many parameters from mBERT are involved in the training process, increasing the randomness variable in model training. The behavior of the models is similar before and after the fine-tuning. The fine-tuned SAN-RPRs perform better than SAPR-RPRs with 54.01% and 53.82% average F1 scores, respectively.

No	Hyperparameter	Value
General		
1	gnn_activation gnn_activation_at_final_layer	ReLU, **Leaky ReLU**, ELU T, **F**
GATs		
1	num_enc_layers enc_dropout	1, **2**, 3, 4 **0.1**, 0.2, 0.3, 0.4, 0.5
2	deprel_dim abs_position_dim <use_dep_abs_position, use_word_abs_position>	0, **30** 0, **30** <**T, F**>, <F, T>
SHGNs		
1	num_enc_layers enc_dropout	1, **2**, 3, 4 0.1, 0.2, **0.3**, 0.4, 0.5
2	deprel_ext_edge_dim rel_pos_dim use_word_rel_pos	**16**, 32, 64, 128 **16**, 32, 64, 128 T
3	deprel_dim abs_position_dim use_word_rel_pos use_dep_abs_position	**0**, 30 **0**, 30 T, **F** T
4	<deprel_edge_dim, deparc_edge_dim, deprel_ext_edge_dim>	<8, 8, 0>, <12, 4, 0>, <14, 2, 0>, <15, 1, 0>, <**0, 0, 16**>
TAGATs		
1	num_enc_layers enc_dropout	1, **2**, 3, 4 0.1, 0.2, **0.3**, 0.4, 0.5
2	deprel_ext_edge_dim rel_pos_dim use_word_rel_pos	**16**, 32, 64, 128 16, 32, 64, **128** T
3	deprel_dim abs_position_dim use_word_rel_pos use_dep_abs_position	**0**, 30 0, **30** **T**, F T
4	<deprel_edge_dim, deparc_edge_dim, deprel_ext_edge_dim>	<8, 8, 0>, <12, 4, 0>, <14, 2, 0>, <15, 1, 0>, <**0, 0, 16**>
KBGATs		
1	num_enc_layers enc_dropout	1, **2**, 3, 4 0.1, **0.2**, 0.3, 0.4, 0.5
2	deprel_ext_edge_dim rel_pos_dim use_word_rel_pos	16, **32**, 64, 128 16, **32**, 64, 128 T
3	deprel_dim abs_position_dim use_word_rel_pos use_dep_abs_position	0, **30** **0**, 30 **T**, F T
4	<deprel_edge_dim, deparc_edge_dim, deprel_ext_edge_dim>	<16, 16, 0>, <**24, 8, 0**>, <28, 4, 0>, <30, 2, 0>, <31, 1, 0>, <0, 0, 32>

Table 13: Hyperparameter search in GAT-based models. The search is divided into groups shown in the "No" header. We search for the best combination of hyperparameters in the same group. The bold values indicate the results of the hyperparameter search.

	GATEs	Trans	SAN-RPRs	SAN-SAPRs	SAN-SRPRs	Trans-SRPRs	SAPR-RPRs	Trans-SRPR-DRs	SAPR-RPR-DRs
EN	78.96±0.31	76.16±0.51	78.26±0.40	75.93±0.47	78.27±0.50	79.03±0.32	78.11±0.42	79.85±0.21	79.83±0.19
AVG	52.57±0.23	52.07±0.21	**52.73±0.40**	51.98±0.10	51.51±0.27	52.21±0.30	52.64±0.38	50.60±0.14	50.69±0.21
TA	36.72±0.89	35.32±1.56	37.96±1.75	38.07±1.47	35.93±2.16	35.74±1.47	**39.57±1.18**	32.23±1.14	32.91±1.22
HI	**48.56±0.37**	47.40±0.18	47.51±0.62	45.07±0.52	45.70±0.63	47.80±0.57	45.04±0.35	48.04±0.45	47.57±0.65
ZH	48.09±0.64	48.86±0.31	50.37±1.17	50.06±0.15	46.96±0.54	46.51±0.53	**50.96±0.88**	43.63±1.09	43.12±0.87
JA	37.45±0.86	**38.38±0.37**	37.69±0.99	37.91±0.88	36.29±1.60	36.46±1.20	34.78±1.29	33.86±0.61	33.56±1.47
VI	28.01±0.91	28.25±0.57	28.69±0.79	**29.70±0.50**	27.70±0.61	27.34±0.59	29.10±0.45	25.72±0.22	25.37±0.23
KO	43.44±0.90	43.13±2.26	42.61±1.84	**46.09±1.02**	43.38±0.84	43.62±0.57	45.24±1.23	39.34±1.19	40.76±0.51
ID	57.88±0.47	55.36±0.42	58.78±1.09	57.95±0.56	59.79±0.27	58.41±0.35	**59.97±0.53**	53.41±0.69	52.89±0.25
HU	**50.69±0.25**	50.13±0.85	49.76±0.35	49.68±0.56	49.28±0.20	50.60±0.49	49.08±0.34	49.51±0.47	49.31±0.72
RO	53.86±0.57	52.90±0.39	54.23±0.67	52.11±0.23	52.44±0.33	53.75±0.55	**54.46±0.52**	52.50±0.31	52.50±0.49
FR	61.67±0.30	60.23±0.39	**62.19±0.41**	60.52±0.27	60.34±0.37	61.30±0.50	62.11±0.47	60.82±0.20	60.91±0.28
MR	37.49±1.74	**42.85±2.67**	41.06±2.89	41.22±1.65	37.37±1.27	36.58±1.68	40.36±2.20	36.75±1.67	37.40±1.56
UK	58.89±0.53	58.83±0.20	58.92±0.26	58.87±0.48	58.23±0.69	58.93±0.72	**59.36±0.72**	57.94±0.42	58.23±0.39
PT	65.49±0.24	64.05±0.26	66.05±0.21	64.76±0.48	65.16±0.41	65.34±0.15	**66.49±0.33**	64.61±0.24	64.84±0.17
IT	57.52±0.38	57.42±0.43	**58.11±0.33**	56.35±0.45	56.09±0.33	57.42±0.32	57.80±0.39	56.60±0.31	56.72±0.42
ES	63.36±0.24	62.04±0.42	**63.71±0.33**	61.53±0.31	62.70±0.20	63.47±0.27	63.62±0.25	61.78±0.14	61.68±0.22
CS	**57.69±0.39**	56.19±0.48	56.87±0.27	55.27±0.57	56.58±0.27	57.61±0.34	55.80±0.51	56.44±0.26	56.35±0.49
EL	60.37±0.59	59.03±0.34	**60.59±0.23**	57.30±0.69	58.98±0.82	59.31±0.30	60.23±0.40	57.89±0.33	58.33±0.33
FI	55.00±0.34	54.72±0.33	**55.58±0.42**	54.74±0.47	53.91±0.37	54.86±0.31	55.29±0.54	54.11±0.17	54.15±0.25
RU	59.36±0.24	59.31±0.47	60.18±0.44	60.33±0.37	59.09±0.30	59.29±0.40	**61.13±0.50**	58.60±0.39	58.56±0.25
NL	**63.73±0.53**	62.48±0.76	62.84±0.37	61.75±0.26	61.73±0.41	63.07±0.49	62.22±0.58	62.23±0.20	62.43±0.39
TE	**46.32±0.98**	44.94±1.38	44.66±2.00	41.64±1.81	41.88±1.17	46.10±1.10	43.88±1.57	42.48±1.28	43.09±1.26
DE	**58.88±0.40**	58.29±0.70	56.86±0.32	58.39±0.34	58.14±0.30	58.84±0.30	56.98±1.13	58.21±0.26	58.30±0.34
PL	**58.73±0.64**	57.58±0.25	57.67±0.36	56.25±0.35	57.14±0.40	58.60±0.29	57.28±0.46	57.18±0.62	56.85±0.64
SC	15	4	13	8	1	7	20	1	0
PR	12.2M	12.2M	12.2M	12.2M	12.1M	12.2M	12.2M	12.2M	12.2M

Table 14: F1 scores (%) of transformer-based models evaluated on UPB v2 test set with predicted parsers. The bold score and underlined score indicate the highest and second-highest scores. AVG indicates the average F1 scores of a specific model evaluated in target languages. PR and SC are the number of parameters and the superiority score of each model.

	base	w/o DRE	w/o POSE	w/o PIE
EN	78.26±0.40	77.56±0.41	78.32±0.32	77.66±0.38
AVG	**52.73±0.40**	51.65±0.28	**52.73±0.32**	51.54±0.36
TA	**37.96±1.75**	37.59±1.95	36.53±1.32	32.03±1.68
HI	47.51±0.62	42.22±0.85	**47.69±1.10**	43.89±0.81
ZH	50.37±1.17	50.42±0.38	**50.81±0.51**	49.83±0.54
JA	**37.69±0.99**	33.25±2.01	36.07±1.20	30.67±2.90
VI	28.69±0.79	29.18±1.05	28.30±0.92	**29.34±0.96**
KO	42.61±1.84	**42.73±1.52**	42.06±1.50	40.13±0.65
ID	**58.78±1.09**	58.41±0.89	57.77±0.83	57.06±0.78
HU	49.76±0.35	48.25±0.62	**50.02±0.42**	49.27±0.86
RO	54.23±0.67	54.11±0.37	**54.94±0.50**	54.19±0.76
FR	62.19±0.41	61.74±0.32	**62.36±0.59**	61.85±0.60
MR	41.06±2.89	39.45±1.76	**41.83±3.50**	41.15±2.62
UK	58.92±0.26	58.75±0.94	58.84±0.59	**59.64±0.15**
PT	66.05±0.21	**66.67±0.53**	66.65±0.44	65.46±0.44
IT	58.11±0.33	57.43±0.24	**58.65±0.49**	58.34±0.32
ES	63.71±0.33	63.18±0.33	**64.07±0.51**	63.63±0.34
CS	56.87±0.27	54.94±0.40	**57.20±0.14**	56.40±0.28
EL	60.59±0.23	59.40±0.23	**60.86±0.56**	59.87±1.11
FI	**55.58±0.42**	54.78±0.25	55.40±0.16	55.18±0.24
RU	**60.18±0.44**	60.05±0.67	60.12±0.36	60.00±0.28
NL	62.84±0.37	60.97±0.84	**63.21±0.38**	60.76±0.68
TE	**44.66±2.00**	42.41±2.63	43.46±2.02	44.40±3.32
DE	56.86±0.32	55.34±1.44	**57.70±0.43**	54.36±0.52
PL	57.67±0.36	56.68±0.63	**58.16±0.35**	58.01±0.31

Table 15: F1 scores (%) of SAN-RPRs with certain embedding removed from the node representation evaluated on UPB v2 test set with predicted parsers. The bold score and underlined score indicate the highest and second-highest scores. AVG indicates the average F1 scores of a specific model evaluated in target languages.

	SAN-RPRs (w/o POSE)	SAPR-RPRs	SGCNs	GATs	TAGATs	BiLSTMs
EN	78.32±0.32	78.11±0.42	79.94±0.27	79.81±0.19	79.07±0.19	76.86±0.34
AVG	52.73±0.32	52.64±0.38	52.52±0.38	52.66±0.14	**52.78±0.14**	51.85±0.09
TA	36.53±1.32	**39.57±1.18**	34.32±1.12	35.08±0.58	35.68±1.28	34.19±1.22
HI	47.69±1.10	45.04±0.35	48.24±0.61	**48.25±0.33**	47.65±0.33	46.63±0.38
ZH	50.81±0.51	**50.96±0.88**	45.77±0.67	46.12±0.39	46.80±0.64	47.56±0.89
JA	36.07±1.20	34.78±1.29	37.43±0.28	37.99±0.52	**39.30±0.73**	37.40±0.61
VI	28.30±0.92	**29.10±0.45**	27.95±0.56	28.06±0.59	28.31±0.55	28.18±0.88
KO	42.06±1.50	**45.24±1.23**	42.92±0.64	43.22±0.56	44.57±0.24	41.77±1.61
ID	57.77±0.83	**59.97±0.53**	58.54±0.82	58.33±0.69	59.11±0.87	56.11±0.84
HU	50.02±0.42	49.08±0.34	50.76±0.41	**51.10±0.51**	50.90±0.37	50.64±0.39
RO	**54.94±0.50**	54.46±0.52	53.57±0.47	54.12±0.49	53.60±0.45	53.26±0.34
FR	**62.36±0.59**	62.11±0.47	60.93±0.38	61.64±0.44	61.13±0.22	61.22±0.27
MR	**41.83±3.50**	40.36±2.20	40.97±3.40	38.06±0.13	39.26±1.20	37.18±2.28
UK	58.84±0.59	59.36±0.72	59.66±0.76	59.49±0.56	**59.72±0.31**	58.96±0.07
PT	**66.65±0.44**	66.49±0.33	65.62±0.43	65.99±0.32	65.61±0.15	64.40±0.33
IT	**58.65±0.49**	57.80±0.39	57.43±0.42	58.00±0.42	57.34±0.34	58.02±0.27
ES	64.07±0.51	63.62±0.25	63.87±0.61	**64.29±0.36**	63.91±0.27	62.48±0.29
CS	57.20±0.14	55.80±0.51	57.95±0.52	**58.02±0.21**	57.62±0.28	56.59±0.36
EL	**60.86±0.56**	60.23±0.40	60.56±0.69	60.74±0.48	**60.86±0.34**	59.76±0.45
FI	**55.40±0.16**	55.29±0.54	54.87±0.40	54.62±0.32	54.88±0.20	54.62±0.20
RU	60.12±0.36	**61.13±0.50**	59.98±0.34	60.14±0.16	60.30±0.22	59.73±0.25
NL	63.21±0.38	62.22±0.58	62.94±0.21	**63.53±0.64**	62.97±0.37	62.47±0.36
TE	43.46±2.02	43.88±1.57	46.08±1.07	**46.96±1.49**	**46.96±1.82**	45.95±0.51
DE	57.70±0.43	56.98±1.13	**58.61±0.30**	58.52±0.26	58.52±0.18	57.72±0.23
PL	58.16±0.35	57.28±0.46	**59.08±0.31**	59.00±0.44	58.92±0.52	57.73±0.23
SC	18	16	-	-	-	-

Table 16: F1 scores (%) of best models evaluated on UPB v2 test set with predicted parsers. The bold score and underlined score indicate the highest and second-highest scores. AVG indicates the average F1 scores of a specific model evaluated in target languages. SC indicates the superiority score of each model.

	Frozen mBERT		Fine-Tuned mBERT	
	SAN-RPRs	SAPR-RPRs	SAN-RPRs	SAPR-RPRs
EN	78.26±0.40	78.11±0.42	79.37±0.60	79.04±0.57
AVG	52.73±0.40	52.64±0.38	54.01±0.95	53.82±1.17

Table 17: F1 scores (%) of SAN-RPRs and SAPR-RPRs with frozen mBERT and fine-tuned mBERT. AVG indicates the average F1 scores of a specific model evaluated in target languages.