Syntax-augmented Multilingual BERT for Cross-lingual Transfer

Wasi Uddin Ahmad[†][*], Haoran Li[‡], Kai-Wei Chang[†], Yashar Mehdad[‡]
[†]University of California, Los Angeles, [‡]Facebook AI
[†]{wasiahmad,kwchang}@cs.ucla.edu, [‡]{aimeeli,mehdad}@fb.com

Abstract

In recent years, we have seen a colossal effort in pre-training multilingual text encoders using large-scale corpora in many languages to facilitate cross-lingual transfer learning. However, due to typological differences across languages, the cross-lingual transfer is challenging. Nevertheless, language syntax, e.g., syntactic dependencies, can bridge the typological gap. Previous works have shown that pre-trained multilingual encoders, such as mBERT (Devlin et al., 2019), capture language syntax, helping cross-lingual transfer. This work shows that explicitly providing language syntax and training mBERT using an auxiliary objective to encode the universal dependency tree structure helps cross-lingual transfer. We perform rigorous experiments on four NLP tasks, including text classification, question answering, named entity recognition, and task-oriented semantic parsing. The experiment results show that syntax-augmented mBERT improves cross-lingual transfer on popular benchmarks, such as PAWS-X and MLQA, by 1.4 and 1.6 points on average across all languages. In the *generalized* transfer setting, the performance boosted significantly, with 3.9 and 3.1 points on average in PAWS-X and MLQA.

1 Introduction

Cross-lingual transfer reduces the requirement of labeled data to perform natural language processing (NLP) in a target language, and thus has the ability to avail NLP applications in low-resource languages. However, transferring across languages is challenging because of linguistic differences at levels of morphology, syntax, and semantics. For example, word order difference is one of the crucial factors that impact cross-lingual transfer (Ahmad et al., 2019). The two sentences in English and Hindi, as shown in Figure 1 have the same meaning but a different word order (while English has an SVO (`Subject-Verb-Object`) order, Hindi follows SOV). However, the sentences have a similar dependency structure, and the constituent words have similar part-of-speech tags. Presumably, language syntax can help to bridge the typological differences across languages.

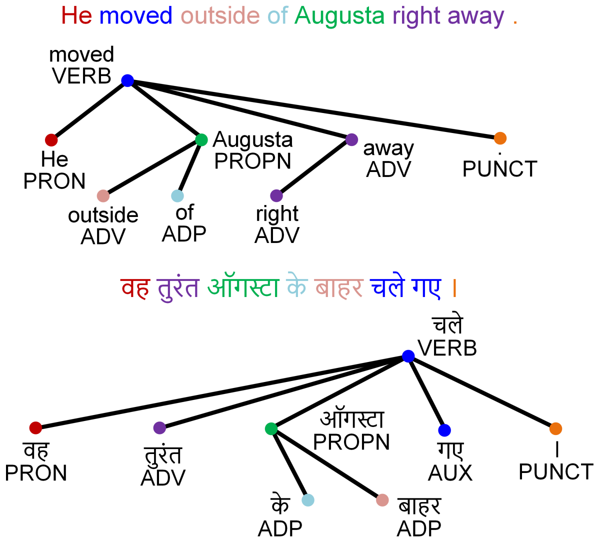

Figure 1: Two parallel sentences in English and Hindi from XNLI (Conneau et al., 2018) dataset. The words highlighted with the same color have the same meaning. Although the sentences have a different word order, their syntactic dependency structure is similar.

In recent years, we have seen a colossal effort to pre-train Transformer encoder (Vaswani et al., 2017) on large-scale unlabeled text data in one or many languages. Multilingual encoders, such as mBERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020) map text sequences into a shared multilingual space by jointly pre-training in many languages. This allows us to transfer the multilingual encoders across languages and have found effective for many NLP applications, including text classification (Bowman et al., 2015; Conneau

[*] Work done during internship at Facebook AI.

Q	English	How many members of the Senate are elected?
	Spanish	Cuántos miembros del Senado son elegidos?
C	English	The Chamber of Deputies has 630 elected members, while the Senate has 315 elected members. . . .
	Spanish	La cámara de los diputados está formada por 630 miembros, mientras que hay 315 senadores más los senadores vitalicios. . . .
A	mBERT	[Q:English-C:English] 315 (✓); [Q:Spanish-C:Spanish] 630 (✗) [Q:Spanish-C:English] 315 (✓); [Q:English-C:Spanish] 630 (✗)
	mBERT + Syn.	[Q:English-C:English] 315 (✓); [Q:Spanish-C:Spanish] 315 (✓) [Q:Spanish-C:English] 315 (✓); [Q:English-C:Spanish] 315 (✓)

Figure 2: A parallel QA example in English (en) and Spanish (es) from MLQA (Lewis et al., 2020) with predictions from mBERT and our proposed syntax-augmented mBERT. In "Q:x-C:y", x and y indicates question and context languages, respectively. Based on our analysis of the highlighted tokens' attention weights, we conjecture that mBERT answers 630 as the token is followed by "miembros", while 315 is followed by "senadores" in Spanish.

et al., 2018), question answering (Rajpurkar et al., 2016; Lewis et al., 2020), named entity recognition (Pires et al., 2019; Wu and Dredze, 2019), and more. Since the introduction of mBERT, several works (Wu and Dredze, 2019; Pires et al., 2019; K et al., 2020) attempted to reason their success in cross-lingual transfer. In particular, Wu and Dredze (2019) showed that mBERT captures language syntax that makes it effective for cross-lingual transfer. A few recent works (Hewitt and Manning, 2019; Jawahar et al., 2019; Chi et al., 2020) suggest that BERT learns compositional features; mimicking a tree-like structure that agrees with the Universal Dependencies taxonomy.

However, fine-tuning for the downstream task in a source language may not require mBERT to retain structural features or learn to encode syntax. We argue that encouraging mBERT to learn the correlation between syntax structure and target labels can benefit cross-lingual transfer. To support our argument, we show an example of question answering (QA) in Figure 2. In the example, mBERT predicts incorrect answers given the Spanish language context that can be corrected by exploiting syntactic clues. Utilizing syntax structure can also benefit *generalized* cross-lingual transfer (Lewis et al., 2020) where the input text sequences belong to different languages. For example, answering an English question based on a Spanish passage or predicting text similarity given the two sentences as shown in Figure 1. In such a setting, syntactic clues may help to align sentences.

In this work, we propose to augment mBERT with universal language syntax while fine-tuning on downstream tasks. We use a graph attention network (GAT) (Veličković et al., 2018) to learn structured representations of the input sequences that are incorporated into the self-attention mechanism. We adopt an auxiliary objective to train GAT such that it embeds the dependency structure of the input sequence accurately. We perform an evaluation on *zero-shot* cross-lingual transfer for text classification, question answering, named entity recognition, and task-oriented semantic parsing. Experiment results show that augmenting mBERT with syntax improves cross-lingual transfer, such as in PAWS-X and MLQA, by 1.4 and 1.6 points on average across all the target languages. Syntax-augmented mBERT achieves remarkable gain in the generalized cross-lingual transfer; in PAWS-X and MLQA, performance is boosted by 3.9 and 3.1 points on average across all language pairs. Furthermore, we discuss challenges and limitations in modeling universal language syntax. We release the code to help future works.[1]

2 Syntax-augmented Multilingual BERT

Multilingual BERT (mBERT) (Devlin et al., 2019) enables cross-lingual learning as it embeds text sequences into a shared multilingual space. mBERT is fine-tuned on downstream tasks, e.g., text classification using *monolingual* data and then directly employed to perform on the target languages. This refers to *zero-shot* cross-lingual transfer. Our main idea is to augment mBERT with language syntax for zero-shot cross-lingual transfer. We employ graph attention network (GAT) (Veličković et al., 2018) to learn syntax representations and fuse them into the self-attention mechanism of mBERT.

[1] https://github.com/wasiahmad/Syntax-MBERT

In this section, we first briefly review the transformer encoder that bases mBERT (§ 2.1), and then describe the graph attention network (GAT) that learns syntax representations from dependency structure of text sequences (§ 2.2). Finally, we describe how language syntax is explicitly incorporated into the transformer encoder (§ 2.3).

2.1 Transformer Encoder

Transformer encoder (Vaswani et al., 2017) is composed of an embedding layer and stacked encoder layers. Each encoder layer consists of two sublayers, a multi-head attention layer followed by a fully connected feed-forward layer. We detail the process of encoding an input token sequence $(w_1, \ldots, w_n)$ into a sequence of vector representations $H = [h_1, \ldots, h_n]$ as follows.

Embedding Layer is parameterized by two embedding matrices — the token embedding matrix $W_e \in R^{U \times d_{model}}$ and the position embedding matrix $W_p \in R^{U \times d_{model}}$ (where U is the vocabulary size and d_{model} is the encoder output dimension). An input text sequence enters into the model as two sequences: the token sequence $(w_1, \ldots, w_n)$ and the corresponding absolute position sequence $(p_1, \ldots, p_n)$. The output of the embedding layer is a sequence of vectors $\{x_i\}_{i=1}^n$ where $x_i = w_i W_e + p_i W_p$. The vectors are packed into matrix $H^0 = [x_1, \ldots, x_n] \in R^{n \times d_{model}}$ and fed to an L-layer encoder.

Multi-head Attention allows to jointly attend to information from different representation subspaces, known as *attention heads*. Multi-head attention layer composed of h attention heads with the same parameterization structure. At each attention head, the output from the previous layer H^{l-1} is first linearly projected into queries, keys, and values as follows.

$$Q = H^{l-1} W_l^Q, K = H^{l-1} W_l^K, V = H^{l-1} W_l^V,$$

where the parameters $W_l^Q, W_l^K \in R^{d_{model} \times d_k}$ and $W_l^V \in R^{d_{model} \times d_v}$ are unique per attention head. Then scaled dot-product attention is performed to compute the output vectors $\{o_i\}_{i=1}^n \in R^{n \times d_v}$.

$$\begin{aligned} &\text{Attention}(Q, K, V, M, d_k) \\ &= \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right) V, \end{aligned} \quad (1)$$

where $M \in \mathbb{R}^{n \times n}$ is the masking matrix that determines whether a pair of input positions can attend each other. In classic multi-head attention, M is a *zero* matrix (all positions can attend each other).

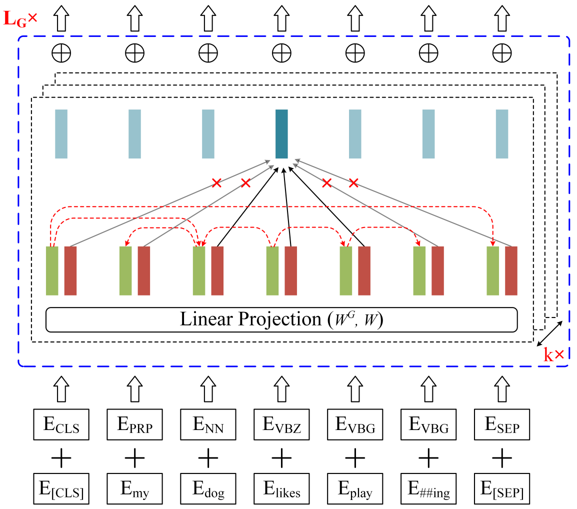

Figure 3: A simplified illustration of the multi-head self-attention in the graph attention network wherein each head attention is allowed between words within δ distance from each other in the dependency graph. For example, as shown, in one of the attention heads, the word “likes” is only allowed to attend its adjacent (δ=1) words “dog” and “play”.

The output vectors from all the attention heads are concatenated and projected into d_{model} dimension using the parameter matrix $W_o \in R^{hd_v \times d_{model}}$. Finally the vectors are passed through a feed-forward network to output $H^l \in R^{n \times d_{model}}$.

2.2 Graph Attention Network

We embed the syntax structure of the input token sequences using their universal dependency parse. A dependency parse is a directed graph where the nodes represent words, and the edges represent dependencies (the dependency relation between the head and dependent words). We use a graph attention network (GAT) (Veličković et al., 2018) to embed the dependency tree structure of the input sequence. We illustrate GAT in Figure 3.

Given the input sequence, the words (w_i) and their part-of-speech tags (pos_i) are embedded into vectors using two parameter matrices: the token embedding matrix W_e and the part-of-tag embedding W_{pos}. The input sequence is then encoded into an input matrix $\mathcal{G}^0 = [g_1, \ldots, g_n]$, where $g_i = w_i W_e + pos_i W_{pos} \in R^{d_{model}}$. Note that token embedding matrix W_e is shared between GAT and the Transformer encoder. Then $\mathcal{G}^0$ is fed into an $L_{\mathcal{G}}$-layer GAT where each layer generates word representations by attending their adjacent words.

GAT uses the multi-head attention mechanism and perform a *dependency-aware* self-attention as

$$\mathcal{O} = \text{Attention}(\mathcal{T}, \mathcal{T}, V, M, d_g) \qquad (2)$$

namely setting the query and key matrices to be the same $\mathcal{T} \in R^{n \times d_g}$ respectively and the mask M by

$$M_{ij} = \begin{cases} 0, & D_{ij} \leq \delta \\ -\infty, & \text{otherwise} \end{cases} \qquad (3)$$

where D is the distance matrix and D_{ij} indicates the shortest path distance between word i and j in the dependency graph structure.

Typically in GAT, δ is set to 1; allowing attention between adjacent words only. However, in our study, we find setting δ to [2, 4] helpful for the downstream tasks. Finally, the vector representations from all the attention heads (as in Eq. (2)) are concatenated to form the output representations $\mathcal{G}^l \in R^{n \times kd_g}$, where k is the number of attention heads employed. The goal of the GAT encoder is to encode the dependency structure into vector representations. Therefore, we design GAT to be light-weight; consisting of much less parameters in comparison to Transformer encoder. Note that, GAT does not employ positional representations and only consists of multi-head attention; there is no feed-forward sublayer and residual connections.

Dependency Tree over Wordpieces and Special Symbols mBERT tokenizes the input sequence into subword units, also known as wordpieces. Therefore, we modify the dependency structure of linguistic tokens to accommodate wordpieces. We introduce additional dependencies between the first subword (head) and the rest of the subwords (dependents) of a linguistic token. More specifically, we introduce new edges from the head subword to the dependent subwords. The inputs to mBERT use special symbols: [CLS] and [SEP]. We add an edge from the [CLS] token to the root of the dependency tree and the [SEP] tokens.

2.3 Syntax-augmented Transformer Encoder

We want the Transformer encoder to consider syntax structure while performing the self-attention between input sequence elements. We use the syntax representations produced by GAT (outputs from the last layer, denoting as $\mathcal{G}$) to *bias* the self-attention.

$$O = \text{Attention}(Q + \mathcal{G}G_l^Q, K + \mathcal{G}G_l^K, V, M, d_k),$$

where $G_l^Q, G_l^K \in R^{d_{kd_g} \times d_k}$ are new parameters that learn representations to bias the self-attention. We consider the addition terms $(\mathcal{G}G_l^Q, \mathcal{G}G_l^K)$ as syntax-bias that provide syntactic clues to guide the self-attention. The high-level intuition behind the syntax bias is to attend tokens with a specific part-of-speech tag sequence or dependencies.[2]

Syntax-heads mBERT employs h (=12) attention heads and the syntax representations can be infused into one or more of these heads, and we refer them as *syntax-heads*. In our experiments, we observed that instilling structural information into many attention heads degenerates the performance. For the downstream tasks, we consider one or two syntax-heads that gives the best performance.[3]

Syntax-layers refers to the encoder layers that are infused by syntax representations from GAT. mBERT has a 12-layer encoder and our study finds considering all of the layers as *syntax-layers* beneficial for cross-lingual transfer.

2.4 Fine-tuning

We jointly fine-tune mBERT and GAT on downstream tasks in the source language (English in this work) following the standard procedure. However, the task-specific training may not guide GAT to encode the tree structure. Therefore, we adopt an auxiliary objective that supervises GAT to learn representations which can be used to decode the tree structure. More specifically, we use GAT's output representations $\mathcal{G} = [g_1, \ldots, g_n]$ to predict the tree distance between all pairs of words (g_i, g_j) and the tree depth $||g_i||$ of each word w_i in the input sequence. Following Hewitt and Manning (2019), we apply a linear transformation $\theta_1 \in R^{m \times kd_g}$ to compute squared distances as follows.

$$d_{\theta_1}(g_i, g_j)^2 = (\theta_1(g_i - g_j))^T(\theta_1(g_i - g_j)).$$

The parameter matrix θ_1 is learnt by minimizing:

$$\min_{\theta_1} \sum_s \frac{1}{n^2} \sum_{i,j} |dist(w_i, w_j)^2 - d_\theta(g_i, g_j)^2|,$$

where s denotes all the text sequences in the training corpus. Similarly, we train another parameter matrix θ_2 to compute squared vector norms, $d_{\theta_2}(g_i) = (\theta_2 g_i)^T(\theta_2 g_i)$ that characterize the tree

[2] In example shown in Figure 2, token dependencies: [en: root → has → has → members → 315], and [es: root → formada → hay → senadores → 315] or corresponding part-of-speech tag sequence [VERB → VERB → NOUN → NUM]) may help mBERT to predict the correct answer.

[3] This aligns with the findings of Hewitt and Manning (2019) as they showed 64 or 128 dimension of the contextual representations are sufficient to capture the syntax structure.

Dataset	Task	|Train|	|Dev|	|Test|	|Lang|	Metric
XNLI	Classification	392K	2.5K	5K	13	Accuracy
PAWS-X	Classification	49K	2K	2K	7	Accuracy
MLQA	QA	87K	34K	4.5K-11K	7	F1 / Exact Match
XQuAD	QA	87K	34K	1190	10	F1 / Exact Match
Wikiann	NER	20K	10K	1K-10K	15	F1
CoNLL	NER	15K	2K-3K	1.5K-5K	4	F1
mTOP	Semantic Parsing	15.7K	2.2K	2.8K-4.4K	5	Exact Match
mATIS++	Semantic Parsing	4.5K	490	893	9	Exact Match

Table 1: Statistics of the evaluation datasets. |Train|, |Dev| and |Test| are the numbers of examples in the training, dev and test sets, respectively. For train set, the number is for the source language, English, while for dev and test set, the number is for each target language. |Lang| is the number of target languages we consider for each task.

depth of the words. We train GAT's parameters and θ_1, θ_2 by minimizing the loss: $\mathcal{L} = \mathcal{L}_{task} + \alpha(\mathcal{L}_{dist} + \mathcal{L}_{depth})$, where α is weight for the tree structure prediction loss.

Pre-training GAT Unlike mBERT's parameters, GAT's parameters are trained from scratch during task-specific fine-tuning. For low-resource tasks, GAT may not learn to encode the syntax structure accurately. Therefore, we utilize the universal dependency parses[4] to pre-train GAT on the source and target languages. Note that, the pre-training objective for GAT is to predict the tree distances and depths as described above.

3 Experiment Setup

To study syntax-augmented mBERT's performance in a broader context, we perform an evaluation on four NLP applications: text classification, named entity recognition, question answering, and task-oriented semantic parsing. Our evaluation focuses on assessing the usefulness of utilizing universal syntax in the *zero-shot* cross-lingual transfer.

3.1 Evaluation Tasks

Text Classification We conduct experiments on two widely used cross-lingual text classification tasks: (i) natural language inference and (ii) paraphrase detection. We use the XNLI (Conneau et al., 2018) and PAWS-X (Yang et al., 2019) datasets for the tasks, respectively. In both tasks, a pair of sentences is given as input to mBERT. We combine the dependency tree structure of the two sentences by adding two edges from the [CLS] token to the roots of the dependency trees. A linear classifier takes the contextual representation for the [CLS] token to predict the target label.

Named Entity Recognition is a structure prediction task that requires to identify the named entities mentioned in the input sentence. We use the Wikiann dataset (Pan et al., 2017) and a subset of two tasks from CoNLL-2002 (Tjong Kim Sang, 2002) and CoNLL-2003 NER (Tjong Kim Sang and De Meulder, 2003). We collect the CoNLL datasets from XGLUE (Liang et al., 2020). In both datasets, there are 4 types of named entities: Person, Location, Organization, and Miscellaneous.[5]

Question Answering We evaluate on two cross-lingual question answering benchmarks, MLQA (Lewis et al., 2020), and XQuAD (Artetxe et al., 2020). We use the SQuAD dataset (Rajpurkar et al., 2016) for training and validation. In the QA task, the inputs are a question and a context passage that consists of many sentences. We formulate QA as a multi-sentence reading comprehension task; jointly train the models to predict the answer sentence and extract the answer span from it. We concatenate the question and each sentence from the context passage and use the [CLS] token representation to score the candidate sentences. We provide more details of the QA models in Appendix.

Task-oriented Semantic Parsing The fourth evaluation task is cross-lingual task-oriented semantic parsing. In this task, the input is a user utterance and the goal is to predict the intent of the utterance and fill the corresponding slots. We conduct experiments on two recently proposed benchmarks: (i) mTOP (Li et al., 2021) and (ii) mATIS++ (Xu et al., 2020). We jointly train the BERT models as suggested in Chen et al. (2019).

We summarize the evaluation task benchmark datasets and evaluation metrics in Table 1.

[4] https://universaldependencies.org/

[5] Miscellaneous entity type covers named entities that do not belong to the other three types

Model	en	ar	bg	de	el	es	fr	hi	ru	tr	ur	vi	zh	ko	ja	nl	pt	AVG
Classification - XNLI (Conneau et al., 2018)																		
[1]	80.8	64.3	68.0	70.0	65.3	73.5	73.4	58.9	67.8	60.9	57.2	69.3	67.8	-	-	-	-	67.5
mBERT	81.8	63.8	68.0	70.7	65.4	73.8	72.4	59.3	68.4	60.7	56.7	68.6	67.8	-	-	-	-	67.5
+ Syn.	81.6	**65.4**	**69.3**	70.7	**66.5**	**74.1**	**73.2**	**60.5**	68.8	**62.4**	**58.7**	**69.9**	**69.3**	-	-	-	-	68.5
Classification - PAWS-X (Yang et al., 2019)																		
[1]	94.0	-	-	85.7	-	87.4	87.0	-	-	-	-	-	77.0	69.6	73.0	-	-	82.0
mBERT	93.9	-	-	85.7	-	88.4	87.6	-	-	-	-	-	78.0	73.6	73.1	-	-	82.9
+ Syn.	94.0	-	-	85.9	-	**89.1**	**88.2**	-	-	-	-	-	**80.7**	**76.3**	**75.8**	-	-	84.3
NER - Wikiann (Pan et al., 2017)																		
[1]	85.2	41.1	77.0	78.0	72.5	77.4	79.6	65.0	64.0	71.8	36.9	71.8	-	59.6	-	81.8	80.8	69.5
mBERT	83.6	38.8	77.0	76.0	70.4	74.7	78.9	63.4	63.5	70.9	37.7	73.5	-	59.3	-	81.9	78.7	68.5
+ Syn.	84.4	**40.0**	77.0	**77.0**	**71.5**	**76.1**	79.3	64.2	63.8	71.4	37.3	72.7	-	59.3	-	81.9	79.0	69.0
NER - CoNLL (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003)																		
[2]	90.6	-	-	69.2	-	75.4	-	-	-	-	-	-	-	-	-	77.9	-	78.2
mBERT	90.7	-	-	68.3	-	74.5	-	-	-	-	-	-	-	-	-	77.6	-	77.8
+ Syn.	90.6	-	-	**69.1**	-	73.6	-	-	-	-	-	-	-	-	-	**78.5**	-	78.0
QA - MLQA (Lewis et al., 2020)																		
[3]	77.7	45.7	-	57.9	-	64.3	-	43.8	-	-	-	57.1	57.5	-	-	-	-	57.7
mBERT	80.5	47.2	-	59.0	-	63.9	-	47.5	-	-	-	56.5	56.6	-	-	-	-	58.7
+ Syn.	80.4	**48.9**	-	**60.8**	-	**65.9**	-	46.7	-	-	-	**59.3**	**60.1**	-	-	-	-	60.3
QA - XQuAD (Artetxe et al., 2020)																		
[1]	83.5	61.5	-	70.6	62.6	75.5	-	59.2	71.3	55.4	-	69.5	58.0	-	-	-	-	66.7
mBERT	84.2	54.8	-	68.9	60.2	71.1	-	55.7	68.6	48.9	-	64.0	57.2	-	-	-	-	63.4
+ Syn.	84.0	55.5	-	**71.4**	**61.3**	**72.8**	-	54.6	68.4	**49.8**	-	**67.6**	56.1	-	-	-	-	64.2
Semantic Parsing - mTOP (Li et al., 2021)																		
mBERT	81.0	-	-	28.1	-	40.2	38.8	9.8	-	-	-	-	-	-	-	-	-	39.6
+ Syn.	81.3	-	-	**30.0**	-	**43.0**	**41.2**	**11.5**	-	-	-	-	-	-	-	-	-	41.4
Semantic Parsing - mATIS++ (Xu et al., 2020)																		
mBERT	86.0	-	-	38.1	-	43.7	36.9	16.2	-	1.3	-	-	7.8	-	28.2	-	38.2	32.9
+ Syn.	86.2	-	-	**40.1**	-	**44.5**	**38.9**	**18.7**	-	1.5	-	-	8.0	-	27.3	-	37.3	33.6

Table 2: Cross-lingual transfer results for all the evaluation tasks (on test set) across 17 languages. We report F1 score for the question answering (QA) datasets (for other datasets, see Table 1). We train and evaluate mBERT on the same pre-processed datasets and considers its performance as the *baseline* (denoted by "mBERT" rows in the table) for syntax-augmented mBERT (denoted by "+ Syn." rows in the table). Bold-faced values indicate that the syntax-augmented mBERT is statistically significantly better (by paired bootstrap test, $p < 0.05$) than the *baseline*. We include results from published works ([1]: Hu et al. (2020), [2]: Liang et al. (2020), and [3]: Lewis et al. (2020)) as a reference. Except for the QA datasets, all our results are averaged over three different seeds.

3.2 Implementation Details

We collect the universal part-of-speech tags and the dependency parse of sentences by pre-processing the datasets using UDPipe.[6] We fine-tune mBERT on the pre-processed datasets and consider it as the baseline for our proposed syntax-augmented mBERT. We extend the XTREME framework (Hu et al., 2020) that is developed based on `transformers` API (Wolf et al., 2020). We use the same hyper-parameter setting for mBERT models, as suggested in XTREME. For the graph attention network (GAT), we set $L_G = 4, k = 4$, and $d_g = 64$ (resulting in $\sim$0.5 million parameters). We tune δ[7] (shown in Eq. (3)) and α (weight of the tree structure prediction loss) in the range $[1, 2, 4, 8]$ and $[0.5 - 1.0]$, respectively. We detail the hyper-parameters in the Appendix.

[6] https://ufal.mff.cuni.cz/udpipe/2

[7] We observed that the value of δ depends on the downstream task and the source language. For example, a larger δ value is beneficial for tasks taking a pair of text sequences as inputs, while a smaller δ value results in better performances for tasks taking single text input. Experiments on PAWS-X using each target language as the source language indicate that δ should be set to a larger value for source language with longer text sequences (e.g., Arabic) and vice versa.

s_1/s_2	en	de	es	fr	ja	ko	zh
en	-	0.7	1.6	1.4	4.7	2.5	5.4
de	0.5	-	2.0	2.1	5.1	3.5	5.9
es	1.0	2.1	-	1.7	4.6	3.0	6.6
fr	0.9	1.7	1.9	-	5.0	2.7	5.4
ja	5.2	5.3	5.6	5.1	-	5.9	5.1
ko	3.1	2.8	4.3	3.9	6.4	-	5.1
zh	5.8	5.5	6.3	6.0	6.1	4.5	-

(a) PAWS-X

q/c	en	es	de	ar	hi	vi	zh
en	-	-0.2	0.3	0.4	0.9	0.6	1.1
es	4.1	-	3.5	5.4	5.3	7.3	7.6
de	3.5	2.8	-	4.0	2.9	4.0	5.0
de	1.8	2.4	1.1	-	-0.1	6.2	4.4
hi	1.0	1.8	0.5	0.2	-	-0.6	1.0
vi	5.6	4.5	5.5	6.9	4.2	-	5.5
zh	3.8	3.3	4.4	2.4	0.9	5.4	-

(b) MLQA

Table 3: The performance difference between syntax-augmented mBERT and mBERT in the *generalized* cross-lingual transfer setting. The rows and columns indicate (a) language of the first and second sentences in the candidate pairs and (b) context and question languages. The gray cells have a value greater than or equal to the average performance difference, which is 3.9 and 3.1 for (a) and (b).

4 Experiment Results

We aim to address the following questions.

1. Does augmenting mBERT with syntax improve (generalized) cross-lingual transfer?
2. Does incorporating syntax benefit specific languages or language families?
3. Which NLP tasks or types of tasks get more benefits from utilizing syntax?

4.1 Cross-lingual Transfer

Experiment results to compare mBERT and syntax-augmented mBERT are presented in Table 2. Overall, the incorporation of language syntax in mBERT improves cross-lingual transfer for the downstream tasks, in many languages by a significant margin ($p < 0.05$, t-test). The average performances across all languages on XNLI, PAWS-X, MLQA, and mTOP benchmarks improve significantly (by at least 1 point). On the other benchmarks: Wikiann, CoNLL, XQuAD, and mATIS++, the average performance improvements are 0.5, 0.2, 0.8, and 0.7 points, respectively. Note that the performance gains in the source language (English) for all the datasets except Wikiann is ≤ 0.3. This indicates that cross-lingual transfer gains are not due to improving the downstream tasks, but instead, language syntax helps to transfer across languages.

4.2 Generalized Cross-lingual Transfer

In the generalized cross-lingual transfer setting (Lewis et al., 2020), the input text sequences for the downstream tasks (e.g., text classification, QA) may come from different languages. As shown in Figure 2, given the context passage in English, a multilingual QA model should answer the question written in Spanish. Due to the parallel nature of the existing benchmark datasets: XNLI, PAWS-X, MLQA, and XQuAD, we evaluate mBERT and its' syntax-augmented variant on the generalized cross-lingual transfer setting. The results for PAWS-X and MLQA are presented in Table 3 (results for the other datasets are provided in Appendix).

In both text classification and QA benchmarks, we observe significant improvements for most language pairs. In the PAWS-X text classification task, language pairs with different typologies (e.g., en-ja, en-zh) have the most gains. When Chinese (zh) or Japanese (ja) is in the language pairs, the performance is boosted by at least 4.5%. The dataset characteristics explain this; the task requires modeling structure, context, and word order information. On the other hand, in the XNLI task, the performance gain pattern is scattered, and this is perhaps syntax plays a less significant role in the XNLI task. The largest improvements result when the languages of the premise and hypothesis sentences belong to {Bulgarian, Chinese} and {French, Arabic}.

In both QA datasets, syntax-augmented mBERT boosts performance when the question and context languages are typologically different except the Hindi language. Surprisingly, we observe a large performance gain when questions in Spanish and German are answered based on the English context. Based on our manual analysis on MLQA, we suspect that although questions in Spanish and German are translated from English questions (by human), the context passages are from Wikipedia that often are not exact translation of the corresponding English passage. Take the context passages in Figure 2 as an example. We anticipate that syntactic clues help a QA model in identifying the correct answer span when there are more than one semantically equivalent and plausible answer choices.

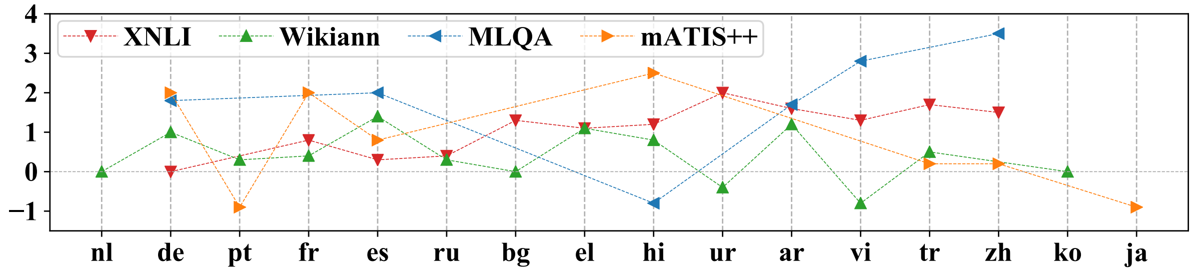

Figure 4: Performance improvements for XNLI, Wikiann, MLQA, and mATIS++ across languages.The languages in x-axis are grouped by language families: IE.Germanic (nl, de), IE.Romance (pt, fr, es), IE.Slavic (ru, bg), IE.Greek (el), IE.Indic (hi, ur), Afro-asiatic (ar, vi), Altaic (tr), Sino-tibetan (zh), Korean (ko), and Japanese (ja).

4.3 Analysis & Discussion

We discuss and analyze our findings on the following points based on the empirical results.

Impact on Languages We study if fine-tuning syntax-augmented mBERT on English (source language) impacts specific target languages or families of languages. We show the performance gains on the target languages grouped by their families in four downstream tasks in Figure 4. There is no observable trend in the overall performance improvements across tasks. However, the XNLI curve weakly indicates that when target languages are typologically different from the source language, there is an increase in the transfer performance (comparing left half to the right half of the curve).

Impact of Pre-training GAT Before fine-tuning syntax-augmented mBERT, we pre-train GAT on the 17 target languages (discussed in § 2.4). In our experiments, we observe such pre-training boosts semantic parsing performance, while there is a little gain on the classification and QA tasks. We also observe that pre-training GAT diminishes the gain of fine-tuning with the auxiliary objective (predicting the tree structure). We hypothesize that pre-training or fine-tuning GAT using auxiliary objective helps when there is limited training data. For example, semantic parsing benchmarks have a small number of training examples, while XNLI has many. As a result, the improvement due to pre-training or fine-tuning GAT in the semantic parsing tasks is significant, and in the XNLI task, it is marginal.

Discussion To foster research in this direction, we discuss additional experiment findings.

• A natural question is, instead of using GAT, why we do not modify attention heads in mBERT to embed the dependency structure (as shown in Eq. 3). We observed a consistent performance drop across all the tasks if we intervene in self-attention (blocking pair-wise attention). We anticipate fusing GAT encoded syntax representations helps as it adds bias to the self-attention. For future works, we suggest exploring ways of adding structure bias, e.g., scaling attention weights based on dependency structure (Bugliarello and Okazaki, 2020).

• Among the evaluation datasets, Wikiann consists of sentence fragments, and the semantic parsing benchmarks consist of user utterances that are typically short in length. Sorting and analyzing the performance improvements based on sequence lengths suggests that the utilization of dependency structure has limited scope for shorter text sequences. However, part-of-speech tags help to identify span boundaries improving the slot filling tasks.

4.4 Limitations and Challenges

In this work, we assume we have access to an off-the-shelf universal parser, e.g., UDPipe (Straka and Straková, 2017) or Stanza (Qi et al., 2020) to collect part-of-speech tags and the dependency structure of the input sequences. Relying on such a parser has a limitation that it may not support all the languages available in benchmark datasets, e.g., we do not consider Thai and Swahili languages in the benchmark datasets.

There are a couple of challenges in utilizing the universal parsers. First, universal parsers tokenize the input sequence into words and provide part-of-speech tags and dependencies for them. The tokenized words may not be a part of the input.[8] As a result, tasks requiring extracting text spans (e.g., QA) need additional mapping from input tokens to words. Second, the parser's output word sequence is tokenized into wordpieces that often results in

[8] For example, in the German sentence "Wir gehen zum kino" (we are going to the cinema), the token "zum" is decomposed into words "zu" and "dem".

inconsistent wordpieces resulting in degenerated performance in the downstream tasks.[9]

5 Related Work

Encoding Syntax for Language Transfer Universal language syntax, e.g., part-of-speech (POS) tags, dependency parse structure, and relations are shown to be helpful for cross-lingual transfer (Kozhevnikov and Titov, 2013; Pražák and Konopík, 2017; Wu et al., 2017; Subburathinam et al., 2019; Liu et al., 2019; Zhang et al., 2019; Xie et al., 2020; Ahmad et al., 2021). Many of these prior works utilized graph neural networks (GNN) to encode the dependency graph structure of the input sequences. In this work, we utilize graph attention networks (GAT) (Veličković et al., 2018), a variant of GNN that employs the multi-head attention mechanism.

Syntax-aware Multi-head Attention A large body of prior works investigated the advantages of incorporating language syntax to enhance the self-attention mechanism (Vaswani et al., 2017). Existing techniques can be broadly divided into two types. The first type of approach relies on an external parser (or human annotation) to get a sentence's dependency structure during inference. This type of approaches embed the dependency structure into contextual representations (Wu et al., 2017; Chen et al., 2017; Wang et al., 2019a,b; Zhang et al., 2019, 2020; Bugliarello and Okazaki, 2020; Sachan et al., 2021; Ahmad et al., 2021). Our proposed method falls under this category; however, unlike prior works, our study investigates if fusing the universal dependency structure into the self-attention of existing multilingual encoders help cross-lingual transfer. Graph attention networks (GATs) that use multi-head attention has also been adopted for NLP tasks (Huang and Carley, 2019) also fall into this category. The second category of approaches does not require the syntax structure of the input text during inference. These approaches are trained to predict the dependency parse via supervised learning (Strubell et al., 2018; Deguchi et al., 2019).

6 Conclusion

Tóm lược lại nội dung nghiên cứu đạt được

In this work, we propose incorporating universal language syntax into multilingual BERT (mBERT) by infusing structured representations into its multi-head attention mechanism. We employ a modified graph attention network to encode the syntax structure of the input sequences. The results endorse the effectiveness of our proposed approach in the cross-lingual transfer. We discuss limitations and challenges to drive future works. Research gap

Ý nghĩa, đóng góp

Acknowledgments

We thank Yuqing Tang for his insightful comments on our paper and anonymous reviewers for their helpful feedback. We also thank UCLA-NLP group for helpful discussions and comments.

Broader Impact

In today's world, the number of speakers for some languages is in billions, while it is only a few thousands for many languages. As a result, a few languages offer large-scale annotated resources, while for many languages, there are limited or no labeled data. Due to this disparity, natural language processing (NLP) is extremely challenging in the low-resourced languages. In recent years, cross-lingual transfer learning has achieved significant improvements, enabling us to avail NLP applications to a wide range of languages that people use across the world. However, one of the challenges in cross-lingual transfer is to learn the linguistic similarity and differences between languages and their correlation with the target NLP applications. Modern transferable models are pre-trained on unlabeled humongous corpora such that they can learn language syntax and semantic and encode them into universal representations. Such pre-trained models can benefit from explicit incorporation of universal language syntax during fine-tuning for different downstream applications. This work presents a thorough study to analyze the pros and cons of utilizing Universal Dependencies (UD) framework that consists of grammar annotations across many human languages. Our work can broadly impact the development of cross-lingual transfer solutions and making them accessible to people across the globe. In this work, we discuss the limitations and challenges in utilizing universal parsers to benefit the pre-trained models. Among the negative aspects of our work is the lack of explanation that why some languages get more benefits over others due to universal syntax knowledge incorporation.

[9] This happen for languages, such as Arabic as parsers normalize the input that lead to inconsistent characters between input text and the output tokenized text.

References

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. 2021. GATE: graph attention transformer encoder for cross-lingual relation and event extraction. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Emanuele Bugliarello and Naoaki Okazaki. 2020. Enhancing machine translation with dependency-aware self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1618–1627, Online. Association for Computational Linguistics.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855, Melbourne, Australia. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Hiroyuki Deguchi, Akihiro Tamura, and Takashi Ninomiya. 2019. Dependency-based self-attention for transformer NMT. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 239–246, Varna, Bulgaria. INCOMA Ltd.

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.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 4411–4421. PMLR.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. 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 5469–5477, Hong Kong, China. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Mikhail Kozhevnikov and Ivan Titov. 2013. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1200, Sofia, Bulgaria. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019. Neural cross-lingual event detection with minimal parallel resources. 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 738–748, Hong Kong, China. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Ondřej Pražák and Miloslav Konopík. 2017. Cross-lingual SRL based upon Universal Dependencies. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 592–600, Varna, Bulgaria. INCOMA Ltd.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. 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 313–325, Hong Kong, China. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

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 30*, pages 5998–6008. Curran Associates, Inc.

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

Chengyi Wang, Shuangzhi Wu, and Shujie Liu. 2019a. Source dependency-aware transformer with supervised self-attention. *arXiv preprint arXiv:1909.02273*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China. Association for Computational Linguistics.

Shuangzhi Wu, Ming Zhou, and Dongdong Zhang. 2017. Improved neural machine translation with source syntax. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4179–4185.

Zhiwen Xie, Runjie Zhu, Kunsong Zhao, Jin Liu, Guangyou Zhou, and Jimmy Xiangji Huang. 2020. A contextual alignment enhanced cross graph attention network for cross-lingual entity alignment. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5918–5928, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Yue Zhang, Rui Wang, and Luo Si. 2019. Syntax-enhanced self-attention-based semantic role labeling. 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 616–626, Hong Kong, China. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020. SG-Net: Syntax-guided machine reading comprehension. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Supplementary Material: Appendices

A Model Implementations

We follow the standard way to model text classification, named entity recognition, and task-oriented semantic parsing using mBERT. However, since our proposed model uses the input sentences' dependency structure, we frame question answering (QA) as multi-sentence reading comprehension. The input context is split into a list of sentences and train the mBERT model to predict the answer sentence and extract the answer span from the selected sentence following Clark and Gardner (2018). We concatenate the question and each sentence from the context passage and use the [CLS] token representation to score the candidate sentences. We adopt the *shared-normalization* approach from the "confidence method" as suggested in Clark and Gardner (2018) and pick the highest-scored sentence to extract the answer span during inference. Our approach of utilizing syntax can be extended to apply to passages directly. To combine all the sentences' dependency structure in the passage, we can add edges from the [CLS] token to the roots of all the sentences' dependency tree. However, would that approach work in practice requires empirical study, and we leave this as future work.

B Hyper-prameter Details

We present the hyper-parameter details in Table 4.

C Additional Experiment Results

Cross-lingual Transfer We provide the exact match (EM) and F1 accuracy of the MLQA dataset in Table 5. Intent classification accuracy, slot F1, and exact match (EM) accuracy for the task-oriented semantic parsing is reported in Table zero-shot cross-lingual transfer results for the evaluation tasks in Table 6. We highlight the cross-lingual transfer gap for mBERT and syntax-augmented mBERT on the evaluation tasks in Table 7.

Generalized Cross-lingual Transfer In generalized cross-lingual transfer, we assume the task inputs are a pair of text that belong to two different languages, e.g., answering Spanish question based on an English context (Lewis et al., 2020). We present the generalized cross-lingual transfer performance of syntax-augmented mBERT on XNLI, MLQA, and XQuAD in Table 8, 9, 10, and 11, respectively. The performance differences between syntax-augmented mBERT and mBERT on the generalized cross-lingual transfer on XNLI and XQuAD is presented in Table 12 and 13.

Different Source Languages In our study, we primarily use English as the source language as training examples used in all the benchmarks are in English. However, authors of many of these benchmarks released translated-train examples in the target languages. This allows us to train mBERT and syntax-augmented mBERT in different languages (as source) and examine how it impacts cross-lingual transfer. We perform experiments on PAWS-X task and present the results in Figure 5. We observe the largest transfer performance improvements when English and German are used as the source language. The improvements are relatively smaller when Japanese, Korean, and Chinese languages are used as the source language. We suspect that the dependency parser may not accurately parse translated sentences, and as a result, we do not see an explainable trend in the improvements.

Graph Attention Network (GAT)	
# layers (L_G)	4 (tuned on [2, 4, 8])
# heads (k)	4 (tuned on [1, 2, 4, 8])
d_g	64 (tuned on [32, 64, 128])
δ	4 (classification, QA), 1 (NER, semantic parsing)
Syntax-augmented mBERT	
# syntax-layers	12 (tuned in the range 1 – 12)
# syntax-heads	1 (XNLI, PAWS-X, Wikiann, CoNLL, mATIS++), 2 (MLQA, XQuAD, mTOP)
α	0.0 (XNLI), 0.2 (mATIS++), 0.5 (PAWS-X, Wikiann, MLQA, XQuAD), 1.0 (CoNLL), 2.0 (mTOP)
# epochs	3 (QA), 5 (classification), 10 (NER, semantic parsing)

Table 4: Details of the hyper-parameters used during fine-tuning syntax-augmented mBERT.

models	en	es	de	ar	hi	vi	zh	avg
Lewis et al.	77.7/65.2	64.3/46.6	57.9/44.3	45.7/29.8	43.8/29.7	57.1/38.6	57.5/37.3	57.7/41.6
mBERT (ours)	80.5/67.2	63.9/44.1	59.0/43.3	47.2/28.6	47.5/32.1	56.5/35.2	57.8/33.0	58.9/40.5
mBERT+Syn.	80.4/67.3	65.9/47.1	60.8/45.1	48.9/30.3	46.7/32.4	59.3/38.0	60.1/35.5	60.3/42.2

Table 5: Zero-shot cross-lingual transfer performance (F1/EM) of mBERT on MLQA dataset.

	mTOP (Li et al., 2021)						mATIS++ (Xu et al., 2020)					
	mBERT (ours)			mBERT+Syn.			mBERT (ours)			mBERT+Syn.		
en	95.5	90.0	81.0	95.6	90.2	81.3	97.3	94.9	86.0	97.3	94.9	86.2
fr	63.8	63.4	38.8	67.8	64.1	41.2	92.9	70.2	36.9	90.4	74.1	38.9
es	68.7	62.1	40.2	73.1	62.9	43.0	94.1	74.0	43.7	90.8	77.2	44.5
de	58.2	60.2	28.1	63.2	59.4	30.0	89.7	68.2	38.1	89.5	69.4	40.1
hi	41.2	30.7	9.8	44.2	31.4	11.5	80.4	49.4	16.2	80.4	54.7	18.7
ja	-	-	-	-	-	-	83.6	60.7	28.2	81.9	61.3	27.3
pt	-	-	-	-	-	-	94.8	66.3	38.2	92.7	66.7	37.3
tr	-	-	-	-	-	-	71.3	16.9	1.3	68.7	18.6	1.5
zh	-	-	-	-	-	-	87.6	24.3	7.8	86.0	21.2	8.0
Avg.	65.5	61.3	39.6	68.8	61.6	41.4	88.0	58.3	32.9	86.4	59.8	33.6

Table 6: Zero-shot cross-lingual task-oriented semantic parsing results. The values for each model indicates intent accuracy, slot F1, and exact match, respectively.

Model	XNLI	PAWS-X	Wikiann	CoNLL	MLQA	XQuAD	mTOP	mATIS++
mBERT (ours)	15.5	12.8	16.1	17.2	25.4	23.2	51.8	59.7
mBERT+Syn.	14.2	11.3	15.7	16.9	23.5	22.1	49.9	59.2

Table 7: The cross-lingual transfer gap of mBERT and syntax-augmented mBERT on the evaluation tasks. The transfer gap is the difference between performance on the English test set and the other languages' average performance. A transfer gap of 0 indicates perfect cross-lingual transfer. For the QA datasets, we use F1 scores.

s_1/s_2	en	de	es	fr	ja	ko	zh
en	-	85.6	87.2	86.3	66.1	68	70.5
de	86.8	-	82	82	65	67.7	68.6
es	87.2	82	-	85.5	63.7	66.1	68.3
fr	86	81.8	84.8	-	64.2	66.6	67.9
ja	65.6	64.5	64.6	64.2	-	67.3	68
ko	69.3	67.3	67.9	67.7	69	-	66.4
zh	71.4	68.1	68.8	68.9	67.5	65.9	-

Table 8: Generalized cross-lingual transfer performance of syntax-augmented mBERT on PAWS-X. The row and column indicates the language of the input sentence pairs.

p/h	en	fr	es	de	ru	el	bg	ar	tr	hi	ur	vi	zh
en	-	70.1	70.1	66.2	64.8	57.3	61.8	59.1	53.8	53.5	51.2	64.4	65.2
fr	72.5	-	69	63.9	63.2	56.5	60.2	58.6	52.5	52.1	49.9	62.4	61.3
es	72.5	68.6	-	63	63.7	57.7	60.8	59.1	52.6	51.5	48.5	61.5	60.9
de	71.1	65.7	65.1	-	63.1	56.1	60	58	52.8	53.2	50.6	60.3	60.5
ru	69.3	64.5	65.5	62.5	-	55.9	62.7	57	51	51.2	48.2	58.8	58.5
el	63	59.8	61	57.5	56.9	-	56.9	55.9	50.6	49.8	47.8	56.6	54.7
bg	68.4	63	64.6	61.2	64	57	-	57.4	51	52.5	48.1	58.8	59.1
ar	63.7	59.2	59.9	56.6	55.8	53.5	54.2	-	50.1	50.9	49.7	55.8	55.5
tr	60	55.2	54.9	53.9	51.9	51.7	52.5	53.2	-	50.4	48.4	53.3	53.7
hi	61.1	55	54.7	54.7	53.7	52	52.3	53.7	50	-	53.2	54	53.7
ur	59.9	55.1	54.1	53.5	50.7	49.9	49	52.9	48.6	54.6	-	50.7	52.6
vi	65.9	60.2	59.3	56.3	55.5	53.2	52.7	54.2	47.8	49.7	46.8	-	62.3
zh	66.8	58.9	58.4	56.1	54.8	50.9	53.4	54.5	48.8	49.4	47.2	61.5	-

Table 9: Generalized cross-lingual transfer performance of syntax-augmented mBERT on XNLI. The row and column indicates the language of premise and hypothesis.

q/c	en	es	de	ar	hi	vi	zh
en	80.4	67.6	63.1	53.3	55.1	64.0	59.9
es	69.3	65.9	58.1	47.8	46.2	56.0	52.2
de	69.2	62.9	60.8	49.5	50.6	56.2	52.3
ar	47.0	44.1	41.3	48.9	35.3	38.9	40.8
hi	41.6	36.5	35.8	32.4	46.7	35.1	33.8
vi	56.2	49.7	47.5	40.6	39.8	59.3	47.9
zh	58.5	52.0	49.0	41.0	38.8	53.0	60.1

Table 10: F1 score for generalized cross-lingual transfer of syntax-augmented on MLQA. Columns show context language, rows show question language.

q/c	en	es	de	ru	el	ar	hi	tr	vi	zh
en	84.2	74.4	70.7	65.1	59.4	52.6	52.9	53.1	63.9	51.7
es	67.1	71.3	58.0	55.6	49.0	45.1	41.9	41.9	48.9	39.5
de	66.6	59.9	69.2	56.0	50.7	43.3	45.6	41.9	50.6	41.7
ru	63.7	61.6	58.0	69.9	49.1	42.8	45.9	41.5	51.0	40.4
el	48.6	46.7	42.9	41.7	63.4	33.9	37.2	30.4	36.9	28.1
ar	47.0	47.4	41.3	42.3	37.6	55.6	37.7	29.1	33.9	31.7
hi	39.1	36.7	37.1	33.9	28.8	29.2	56.2	29.7	31.2	24.2
tr	38.7	35.2	33.9	31.1	26.9	25.1	25.9	49.2	26.2	21.6
vi	53.5	49.0	44.2	43.9	38.2	34.3	36.5	33.7	64.2	37.5
zh	54.1	48.8	45.9	45.6	36.6	37.6	38.3	35.5	48.7	58.0

Table 11: F1 score for generalized cross-lingual transfer for XQuAD. Columns show context language, rows show question language.

p/h	en	fr	es	de	ru	el	bg	ar	tr	hi	ur	vi	zh
en	-	0.9	0.2	0.6	0.5	0.6	0.6	0.9	1.2	0.8	1.1	1.2	1.3
fr	-0.5	-	1.1	1.6	1.3	1.2	1.6	2.3	1.8	1.6	0.9	2.0	1.4
es	-0.4	1.5	-	1.5	0.5	0.9	1.5	1.9	1.9	1.7	1.0	1.7	1.6
de	0.3	1.7	1.2	-	0.7	0.7	1.4	2.2	1.7	1.0	0.8	1.2	1.2
ru	0.8	2.2	1.4	1.4	-	1.5	1.2	2.2	2.0	1.6	1.7	1.8	1.8
el	0.7	2.2	1.8	1.1	1.1	-	1.8	1.8	1.7	1.4	1.4	1.4	1.3
bg	1.1	3.5	2.6	2.3	0.1	1.4	-	2.3	1.7	2.1	1.5	2.9	2.6
ar	0.8	2.7	1.8	2.1	1.3	1.2	1.8	-	2.0	1.5	1.6	1.8	1.6
tr	0.6	2.4	2.0	1.9	1.1	1.5	1.5	1.6	-	1.7	1.5	1.7	1.4
hi	0.9	1.5	1.5	1.4	0.9	0.6	1.4	1.5	1.3	-	1.0	1.1	1.2
ur	1.0	1.9	1.7	1.4	0.4	0.6	0.6	1.2	1.3	1.6	-	0.7	1.0
vi	0.8	3.1	2.0	2.0	1.4	1.4	1.8	2.4	1.4	1.2	1.3	-	2.1
zh	1.7	2.8	2.3	2.2	1.5	1.3	2.2	1.9	1.6	1.4	1.6	3.0	-

Table 12: Cross-lingual transfer performance difference between syntax-augmented mBERT and mBERT on the XNLI dataset in the *generalized* setting. The row and column indicates the language of premise and hypothesis. The gray cell have a value ≥ 1.5 (average difference).

q/c	en	es	de	ru	el	ar	hi	tr	vi	zh
en	-0.3	-1.0	-0.3	0.9	2.2	2.4	1.6	1.3	1.7	0.0
es	4.4	1.6	3.7	4.3	4.4	5.6	6.1	4.3	7.4	4.5
de	4.4	3.9	2.3	3.1	2.7	4.3	2.4	5.4	4.5	3.4
ru	1.8	0.2	1.6	-0.3	1.2	4.7	-0.6	3.5	2.2	4.1
el	-0.2	0.8	1.3	4.2	0.2	4.5	0.4	3.1	4.1	3.0
ar	0.6	-0.1	2.4	1.4	1.1	0.8	-0.3	5.2	3.6	0.9
hi	1.1	1.6	-1.5	2.0	4.2	2.2	-1.2	0.2	2.2	2.7
tr	3.3	4.3	4.0	3.8	5.3	5.9	2.4	0.6	4.1	4.7
vi	4.5	5.1	6.9	4.8	5.7	6.5	3.3	3.7	3.5	5.3
zh	4.0	4.4	5.4	3.7	5.3	5	1.6	3.8	3.6	-1.2

Table 13: F1 score difference for generalized crosslingual transfer for XQuAD. Columns show context language, rows show question language. The gray cells have a value ≥ 3.1 (average difference).

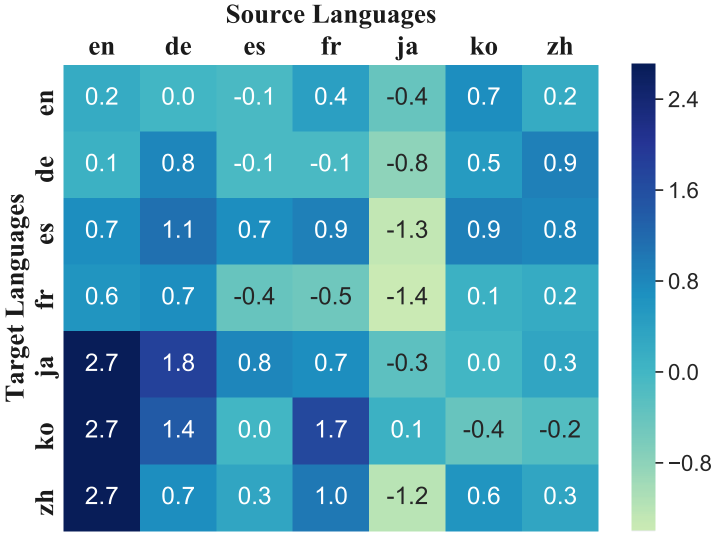

Figure 5: Zero-shot cross-lingual transfer performance difference between syntax-augmented mBERT and mBERT for PAWS-X task using different languages as source.