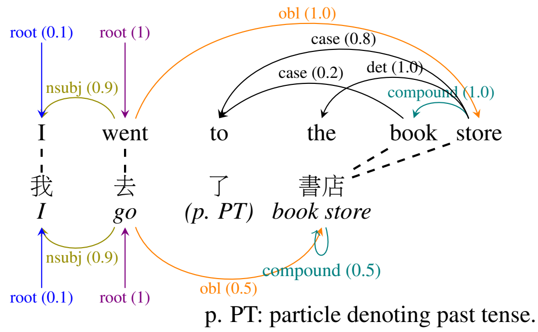
Substructure Distribution Projection for Zero-Shot Cross-Lingual Dependency Parsing

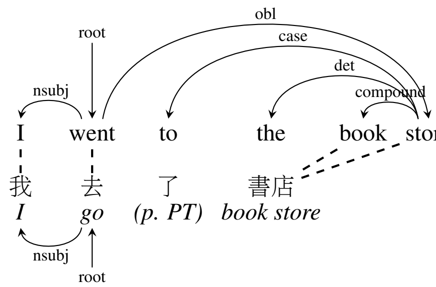
Freda Shi **Kevin Gimpel** **Karen Livescu**
Toyota Technological Institute at Chicago
6045 S Kenwood Ave, Chicago, IL, USA, 60637
{freda, kgimpel, klivescu}@ttic.edu

Abstract

We present substructure distribution projection (SUBDP), a technique that projects a distribution over structures in one domain to another, by projecting substructure distributions separately. Models for the target domain can then be trained, using the projected distributions as soft silver labels. We evaluate SUBDP on zero-shot cross-lingual dependency parsing, taking dependency arcs as substructures: we project the predicted dependency arc distributions in the source language(s) to target language(s), and train a target language parser on the resulting distributions. Given an English treebank as the only source of human supervision, SUBDP achieves better unlabeled attachment score than all prior work on the Universal Dependencies v2.2 (Nivre et al., 2020) test set across eight diverse target languages, as well as the best labeled attachment score on six languages. In addition, SUBDP improves zero-shot cross-lingual dependency parsing with very few (e.g., 50) supervised bitext pairs, across a broader range of target languages.



(a) Soft tree projection with SUBDP. Best viewed in color.



(b) Projection with only one-to-one alignments.

Figure 1: Illustration of SUBDP (top) vs. a representative of annotation projection (bottom; Lacroix et al., 2016). An English parse tree, labeled with the Universal Dependencies conventions (Nivre et al., 2016, 2020), is projected to the parallel Chinese sentence. We denote dependency edges by arrows with the corresponding arc probabilities (if applicable) in parentheses, and word alignments by dashed lines. SUBDP can project either soft or hard trees, whereas most existing work only operates on hard trees.

1 Introduction

Zero-shot cross-lingual dependency parsing is the task that requires prediction of dependency parses without seeing any parsing example in the target language; instead, the model may use annotated parses in other languages. A popular line of work is annotation projection: the parses generated by a source language dependency parser are projected into the target language, where the projected parses are then used to train a new parser. As illustrated in Figure 1b, most annotation projection methods typically output partial hard dependency trees,[1] where there either is or is not an arc between any pair of words. In addition, most bitext-based work has relied on one-to-one word alignment between bitext pairs (e.g., *I* and 我 in Figure 1; Ma and Xia, 2014; Lacroix et al., 2016; Rasooli et al., 2021, *inter alia*), discarding information in many-to-one alignments (e.g., *book store* and 書店 in Figure 1).

In this work, we introduce substructure distribution projection (SUBDP; Figure 1a), where dependency arcs act as substructures. We project substructure distributions, i.e., the conditional prob-

[1]Throughout this paper, we refer to dependency parse trees with 0/1 arc and label probabilities, i.e., conventional dependency trees, as *hard trees*; in contrast, we refer to collections of per-word head distributions and per-arc label distributions with continuous probabilities as *soft trees*.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6547 - 6563

ability distribution of the corresponding head given a word.[2] When the source parse is a hard tree, SUBDP has the same behavior as prior work (e.g., Lacroix et al., 2016) for arcs that are only involved in one-to-one alignments; for many-to-one alignments, SUBDP projects the corresponding arcs into *soft* arc distributions in the target language. Therefore in SUBDP, a target language word may have multiple heads in the projected trees, where their probabilities sum to one. More generally, SUBDP may take dependency arc or label distributions (i.e., soft trees) in the source language(s), instead of hard trees, as the input. As in annotation projection approaches, the projected soft trees are then used to train a target language parser.

We evaluate SUBDP on zero-shot cross-lingual dependency parsing with eight diverse languages from the Universal Dependencies v2.2 (Nivre et al., 2020), where the English treebank is the only source of human supervision. Taking English as the source language, SUBDP significantly outperforms all baseline methods on all distant languages (Arabic, Hindi, Korean, and Turkish) in our experiments, in terms of both labeled attachment scores (LAS) and unlabeled attachment scores (UAS), while achieving superior UAS on all nearby languages (German, French, Spanish, and Italian) as well. Further analysis shows that SUBDP also helps improve zero-shot cross-lingual dependency parsing with a small amount of supervised bitext, across a broader range of target languages.

2 Related Work

Zero-shot cross-lingual dependency parsing.[3] Existing approaches can be classified into the following categories:

1. **Delexicalized training** (Zeman and Resnik, 2008; McDonald et al., 2011; Cohen et al., 2011; Durrett et al., 2012; Rosa and Žabokrtský, 2015, *inter alia*), which only considers delexicalized features (e.g., part-of-speech tags) in training.
2. **Transfer with cross-lingual embeddings** (Täckström et al., 2012; Guo et al., 2015; Schuster et al., 2019, *inter alia*), which assumes that cross-lingual word representations, including word clusters (Täckström et al., 2012; Ammar et al., 2016), word type embeddings (Guo et al., 2015, 2016; Duong et al., 2015; Ammar et al., 2016; Wick et al., 2016), or contextualized cross-lingual word embeddings (Schuster et al., 2019; Wang et al., 2019; He et al., 2019; Ahmad et al., 2019a,b), provide shared features for words with similar syntactic roles.
3. **Treebank translation**, which translates treebanks in the source language(s) into the target language(s) (Tiedemann et al., 2014; Tiedemann, 2015; Tiedemann and Agić, 2016) or a code-switching mode (Zhang et al., 2019), and uses them to train target language parsers.
4. **Annotation projection**,[4] which trains a parser in the source language(s), and projects the predicted source language parse trees to target language(s) using bitext (Hwa et al., 2005; Ma and Xia, 2014; Agić et al., 2016). Additional strategies are usually used to improve the projection quality, such as keeping confident edges only (Li et al., 2014; Lacroix et al., 2016), projection from multiple source languages (Täckström et al., 2013; Agić et al., 2016; Rasooli and Collins, 2017), density based iterative filtering (Rasooli and Collins, 2015, 2017, 2019), and noisy self-training (Kurniawan et al., 2021).

These approaches make different assumptions on annotation availability, such as gold part-of-speech tags (Zeman and Resnik, 2008; Cohen et al., 2011; Durrett et al., 2012, *inter alia*), a reasonably good translator, which uses extra annotated data in the training process (Tiedemann et al., 2014; Tiedemann, 2015; Zhang et al., 2019), high-quality bilingual lexicons (Durrett et al., 2012; Guo et al., 2015, 2016, *inter alia*), or language-specific constraints (Meng et al., 2019). Most bitext-based work assumes annotated bitext (Ma and Xia, 2014; Li et al., 2014; Lacroix et al., 2016, *inter alia*) or bitext constructed from extra signals (e.g., Wikipedia; Rasooli et al., 2021). However, He et al. (2019), Schuster et al. (2019), Ahmad et al. (2019a,b), and Kurniawan et al. (2021) only require minimal annotations (i.e., source language treebanks and unlimited raw text in relevant languages). We are mainly interested in the minimal annotation setting, and will compare to this line of work.

Our proposed method, SUBDP, falls into the category of annotation projection. Some of the

[2] Projection of the distribution over whole parse trees has been considered by Ma and Xia (2014), while SUBDP has a much lower time complexity – see §2 for more discussion.

[3] Also referred to as zero-shot dependency parsing in recent literature (Schuster et al., 2019; Wang et al., 2019).

[4] We use *annotation projection* to denote the projection of predicted parses following Rasooli and Collins (2019) and Zhang et al. (2019), and *treebank translation* for the projection of human-annotated trees following Tiedemann et al. (2014).

benefits of SUBDP relative to prior work are that it works well with minimal annotations, allows soft word alignment (§3.2), supports both labeled and unlabeled parsing, and has a low time complexity $\mathcal{O}(n^2)$ for non-projective parsing.[5] SUBDP can be easily extended to other tasks, such as sequence labeling, where we can define substructures (Shi et al., 2021a) and substructure distributions.

Multilingual contextualized representations. Recent contextualized models pretrained on multilingual text (Devlin et al., 2019; Conneau et al., 2020; Tran et al., 2020, *inter alia*) are effective across a wide range of cross-lingual NLP tasks, including bitext retrieval (Tran et al., 2020), bilingual lexicon induction (Shi et al., 2021b), cross-lingual named entity recognition (Pires et al., 2019; Mulcaire et al., 2019), and cross-lingual dependency parsing (Schuster et al., 2019; Wang et al., 2019). In this work, we apply two of the contextualized pretrained models, XLM-R (Conneau et al., 2020) and CRISS (Tran et al., 2020) to generate unsupervised bitext.

Soft-label methods. Calculating the cross entropy loss between model output and a soft distribution (instead of one-hot labeles) has been applied to knowledge distillation (Hinton et al., 2015; You et al., 2017; Sanh et al., 2019, *inter alia*), cross-lingual named entity recognition (Wu et al., 2020), and for handling annotation discrepancy (Fornaciari et al., 2021). Our approach is a type of soft-label method, with additional post processing to the output of the original models.

3 Proposed Approach: SUBDP

Our pipeline for zero-shot cross-lingual dependency parsing consists of three steps: (1) train a bi-affine dependency parser $\mathcal{P}_1$ in the source language L_1, (2) project annotations on L_1 sentences to their parallel sentences in the target language L_2 (§3.3), and (3) train another bi-affine dependency parser $\mathcal{P}_2$ for L_2 (§3.4). We first present some background (§3.1) and preliminaries (§3.2).

3.1 Background

Bi-affine dependency parser. For a sentence with n words $\langle w_1, \ldots, w_n \rangle$,[6] we denote the word features when acting as heads and dependents by $\boldsymbol{H} \in \mathbb{R}^{n \times d_h}$ and $\boldsymbol{D} \in \mathbb{R}^{n \times d_d}$ respectively, where d_h and d_d denote the dimensionality of the corresponding features. The probability of word w_i having head w_j can be formulated as an n-way classification problem:

$$\boldsymbol{S}^{(arc)} = \boldsymbol{D}\boldsymbol{W}^{(arc)}\boldsymbol{H}^\intercal \tag{1}$$

$$P(w_j \mid w_i) = \frac{\exp\left(\boldsymbol{S}^{(arc)}_{i,j}\right)}{\sum_{k=1}^{n} \exp\left(\boldsymbol{S}^{(arc)}_{i,k}\right)}, \tag{2}$$

where $\boldsymbol{W}^{(arc)} \in \mathbb{R}^{d_d \times d_h}$ is the parameters of the bi-affine module.[7] Given $\log P(w_j \mid w_i)$ for every pair of i and j, the dependency trees can be inferred by finding the spanning arborescence of maximum weight using the Chu–Liu-Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1968). We use the algorithm proposed by Tarjan (1977), which has an $\mathcal{O}(n^2)$ time complexity for each sentence.

We denote the candidate dependency label set by L. Parameterized by $\boldsymbol{W}^{(label)} \in \mathbb{R}^{d_d \times d_h \times |L|}$, we define the probability that the arc from head w_j to dependent w_i has the label ℓ by

$$\boldsymbol{S}^{(label)}_{i,j,\ell} = \sum_p \sum_q \boldsymbol{D}_{i,p} \boldsymbol{W}^{(label)}_{p,q,\ell} \boldsymbol{H}_{j,q}$$

$$P(\ell \mid w_j \to w_i) = \frac{\exp\left(\boldsymbol{S}^{(label)}_{i,j,\ell}\right)}{\sum_{k=1}^{|L|} \exp\left(\boldsymbol{S}^{(label)}_{i,j,k}\right)}, \tag{3}$$

Given the probability definitions above, the model is trained to maximize the log likelihood of the training data. More details can be found in Dozat and Manning (2017).

We use bi-affine dependency parsers as the backbone for all parsers in this work, though it is worth noting that SUBDP works for any parser that produces a set of arc and label distributions.

CRISS. CRISS (Tran et al., 2020) is an unsupervised machine translation model trained with monolingual corpora, starting from mBART (Liu et al., 2020), a multilingual pretrained sequence-to-sequence model with a mask-filling denoising objective. During the training process, CRISS iteratively (1) encodes sentences in the monolingual corpora with its encoder, (2) mines bitext based on encoding similarity, and (3) uses the mined bitext to fine-tune the model with a machine translation objective. In this work, we use CRISS to generate

[5] In contrast, Ma and Xia (2014) require $\mathcal{O}(n^4)$ time for non-projective unlabeled dependency parsing.

[6] For convenience, we assume that w_1 is an added dummy word that has one dependent – the root word of the sentence.

[7] While Eq (1) is in a bi-linear form, in practice, we can always append a constant feature column to both $\boldsymbol{H}$ and $\boldsymbol{D}$, resulting in a bi-affine model.

unsupervised translation of English sentences to construct bitext, and apply its encoder to extract word features for an ablation study.

SimAlign. SimAlign (Jalili Sabet et al., 2020) is a similarity based word aligner: given a pair of source and target sentence $\langle s, t\rangle$, SimAlign computes a contextualized representation for each token in both s and t using multilingual pretrained models (Devlin et al., 2019; Conneau et al., 2020), and calculates the similarity matrix S, where $S_{i,j}$ represents the cosine similarity between tokens s_i and t_j. The `argmax` inference algorithm selects position pairs $\langle i, j\rangle$, where $S_{i,j}$ is both horizontal and vertical maximum, and outputs the word pairs corresponding to such position pairs as the word alignment. In this work, we use XLM-R (Conneau et al., 2020) based SimAlign with the `argmax` algorithm to extract word alignment for SubDP. It is worth noting that pretrained multilingual models usually use byte-pair encodings (BPEs; Gage, 1994), a more fine-grained level than words, for tokenization. The `argmax` algorithm may therefore generate many-to-one alignments. More details can be found in Jalili Sabet et al. (2020).

Unlike bitext based word alignment (Och and Ney, 2003; Dyer et al., 2013), SimAlign does not require any bitext to produce high quality alignments, and therefore better fits the low-resource scenario with very few bitext pairs available.

3.2 Preliminaries

Dependency annotations in L_1. As in the most common data settings for supervised dependency parsing, we assume access to sentences with dependency annotations: for a sentence $\langle w_1, \dots, w_n\rangle$, there is a dummy word w_1, whose unique dependent is the root word; every other word w_i is labeled with h_i and r_i, denoting that the head of w_i is w_{h_i}, with the dependency relation r_i. We use these annotations to train an L_1 bi-affine dependency parser $\mathcal{P}_1$, following the procedure described in §3.1.

Bitext. We denote the available m pairs of bitext by $\mathcal{B} = \{\langle s^{(k)}, t^{(k)}\rangle\}_{k=1}^{m}$, where $\{s^{(k)}\}$ and $\{t^{(k)}\}$ are sentences in L_1 and L_2 respectively.

Word alignment. For a bitext pair $\langle s, t\rangle$, we generate the word alignment matrix $\tilde{A} \in \{0,1\}^{|s|\times|t|}$ with SimAlign, where $\tilde{\boldsymbol{A}}_{i,j} = 1$ denotes that there exists an alignment between s_i and t_j.

We would like the word alignment matrices to be right stochastic, i.e., (1) each element is non-negative and (2) each row sums to one, to ensure that the results after projection remain distributions. To handle words that have zero or more than one aligned words in the other language, we introduce the following two matrix operators.

The *add-dummy-position* operator $\Delta(\cdot)$:

$$\begin{aligned}
\Delta : \mathbb{R}^{r\times c} &\to \mathbb{R}^{(r+1)\times(c+1)} (\forall r, c \in \mathbb{N}_+)\\
\Delta(\boldsymbol{M})_{i,j} &= \boldsymbol{M}_{i,j} (1 \le i \le r, 1 \le j \le c);\\
\Delta(\boldsymbol{M})_{i,c+1} &= \mathbb{0}[\boldsymbol{M}_{i,1}, \dots, \boldsymbol{M}_{i,c}] (1 \le i \le r);\\
\Delta(\boldsymbol{M})_{r+1,j} &= 0 (1 \le j \le c);\\
\Delta(\boldsymbol{M})_{r+1,c+1} &= 1,
\end{aligned}$$

where $\mathbb{0}[\cdot] = 1$ when all input values are zero and otherwise 0.

The *row normalization* operator $\mathcal{N}^{\mathcal{R}}(\cdot)$:

$$\begin{aligned}
\mathcal{N}^{\mathcal{R}} : &\mathbb{R}^{r\times c} \to \mathbb{R}^{r\times c} (\forall r, c \in \mathbb{N}_+)\\
\mathcal{N}^{\mathcal{R}}(\boldsymbol{M})_{i,j} &= \frac{\boldsymbol{M}_{i,j}}{\sum_{\ell} \boldsymbol{M}_{i,\ell}}.
\end{aligned}$$

Intuitively, the added dummy positions correspond to *null* words in the word alignment literature (Dyer et al., 2013; Schulz et al., 2016; Jalili Sabet et al., 2020, *inter alia*). We denote the source-to-target alignment matrix by $\boldsymbol{A}^{s\to t} = \mathcal{N}^{\mathcal{R}}\left(\Delta(\tilde{\boldsymbol{A}})\right)$, and the target-to-source alignment matrix by $\boldsymbol{A}^{t\to s} = \mathcal{N}^{\mathcal{R}}\left(\Delta(\tilde{\boldsymbol{A}}^{\intercal})\right)$. Both are right stochastic matrices by definition.

3.3 Dependency Distribution Projection

Arc distribution projection. We consider a pair of bitext $\langle s, t\rangle$. Let $P_1(s_j \mid s_i)$ denote the arc probability produced by the parser $\mathcal{P}_1$. Like the dummy position notation, we specify a dummy $(|s|+1)^{th}$ word whose head is itself, that is,

$$P_1(s_i \mid s_{|s|+1}) = 0,\ P_1(s_{|s|+1} \mid s_{|s|+1}) = 1.$$

We project $P_1(\cdot \mid \cdot)$ to $\hat{P}_2(t_q \mid t_p)$, the arc probability distributions in the parallel L_2 example t,

$$\hat{P}_2(t_q \,|\, t_p) = \sum_{i=1}^{|s|+1} \sum_{j=1}^{|s|+1} \boldsymbol{A}^{t\to s}_{p,i} P_1(s_j \,|\, s_i) \boldsymbol{A}^{s\to t}_{j,q}. \quad (4)$$

It is guaranteed that $\hat{P}_2(\cdot \mid t_p)$ is a distribution for any t_p – a proof can be found in Appendix A.1. Note that if we adopt matrix notations, where we denote $\hat{P}_2(t_q \mid t_p)$ by $\hat{\boldsymbol{P}}^{(2)}_{p,q}$ and denote $P_1(s_j \mid s_i)$ by $\boldsymbol{P}^{(1)}_{i,j}$, Eq (4) is equivalent to

$$\hat{\boldsymbol{P}}^{(2)} = \boldsymbol{A}^{t\to s} \boldsymbol{P}^{(1)} \boldsymbol{A}^{s\to t}.$$

Label distribution projection. Let $P_1(\ell \mid s_j \to s_i)$ denote the label probability produced by $\mathcal{P}_1$. For dummy positions, we simply add a uniform distribution, that is,

$$P_1(\ell \mid s_j \to s_i) = \frac{1}{L} \quad \text{if } i \text{ or } j = |s| + 1.$$

We project $P_1(\cdot \mid \cdot \to \cdot)$ to $\hat{P}_2(\ell \mid t_q \to t_p)$, the label distributions in the parallel L_2 example t, by

$$\hat{P}_2(\ell \mid t_q \to t_p) = \sum_{i=1}^{|s|+1} \sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(\ell \mid s_j \to s_i) \boldsymbol{A}_{q,j}^{t\to s}$$

$\hat{P}_2(\cdot \mid t_q \to t_p)$ is provably a distribution for any pair of t_p and t_q (see Appendix A.2).

3.4 Optimization

We train another bi-affine dependency parser $\mathcal{P}_2$ on language L_2, by minimizing the cross entropy between its produced probability P_2 and the soft silver labels $\hat{P}_2$. Note that the added dummy word denoting the null alignment is not eventually used in the final dependency inference process and may introduce extra noise to the model, so we instead calculate the *partial* cross entropy loss, which does not consider elements involving dummy words. Concretely, we compute the partial arc cross entropy loss for one example t as follows:

$$\mathcal{L}_{arc}^{(t)}(P_2, \hat{P}_2) = -\sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \hat{P}_2(t_q \mid t_p) \log P_2(t_q \mid t_p)$$

Similarly, the partial label cross entropy loss can be computed as follows:

$$\mathcal{L}_{label}^{(t)}(P_2, \hat{P}_2) = -\sum_{\ell=1}^{|L|} \sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \hat{P}_2(\ell \mid t_q \to t_p) \log P_2(\ell \mid t_q \to t_p)$$

Finally, we train the parameters of $\mathcal{P}_2$ to minimize

$$\sum_{\langle s,t\rangle \in \mathcal{B}} \mathcal{L}_{arc}^{(t)}(P_2, \hat{P}_2) + \mathcal{L}_{label}^{(t)}(P_2, \hat{P}_2). \quad (5)$$

4 Experiments

Throughout all experiments, the subword representation is a weighted sum of layer-wise representation from a frozen pretrained model, where each layer has a scalar weight optimized together with other network parameters to minimize Eq. (5). We convert subword features to word features by endpoint concatenation, following Toshniwal et al. (2020). We use the Adam optimizer (Kingma and Ba, 2015) to train all models, where the source language parser is trained for 100 epochs with initial learning rate 2×10^{-3} following the baseline implementation by Zhang et al. (2020), and the target language parser is trained for 30 epochs with initial learning rate 5×10^{-4}.[8] We use the loss against silver projected distributions on the development set for SubDP and the development LAS against projected trees for baselines for early stopping.[9] For evaluation, we ignore all punctuation following the most common convention (Ma and Xia, 2014; Rasooli and Collins, 2015; Kurniawan et al., 2021, *inter alia*). If not specified,

- All models in target languages are initialized with the trained source language parser.
- All word alignments are obtained by XLM-R based SimAlign (Jalili Sabet et al., 2020), using BPE tokenization and the `argmax` algorithm.
- XLM-R is used as the feature extractor.

For analysis, we report results on the standard development sets to avoid tuning on the test sets.

4.1 Results: Fully Unsupervised Transfer

We compare SubDP to prior work in the minimal annotation setting (Table 1), where an English dependency treebank is the only annotation that involves human effort. We select target languages from the overlap between those considered by Kurniawan et al. (2021), those covered by XLM-R (Conneau et al., 2020) training corpora, and those supported by CRISS (Tran et al., 2020), resulting in eight languages: Arabic (ar), Hindi (hi), Korean (ko), Turkish (tr), German (de), Spanish (es), French (fr), and Italian (it).

We translate English sentences using the unsupervised model CRISS to construct the required bitext.[10] To ensure the quality of the unsupervised bitext, we discard (1) translations where at least 80% of words appear in the corresponding source sentences, which are likely to be copies, (2) those

[8] We do not observe further training loss decrease when training for more epochs. The learning rate for SubDP is tuned to optimize the development loss for German, where the German gold trees remain unused.

[9] SubDP does not provide a set of hard silver trees for LAS and UAS calculation.

[10] In experiments, we translate English treebank sentences; in more general cases, any source language sentence can be taken for bitext construction.

Method	LAS								UAS							
	distant languages				*nearby languages*				*distant languages*				*nearby languages*			
	ar	hi	ko	tr	de	es	fr	it	ar	hi	ko	tr	de	es	fr	it
Meng et al.	—	—	—	—	—	—	—	—	47.3	52.4	37.1	35.2	70.8	75.8	79.1	82.0
He et al.	—	—	—	—	—	—	—	—	55.4	33.2	37.0	36.1	69.5	64.3	67.7	70.7
Ahmad et al.	27.9	28.0	16.1	—	61.8	65.8	73.3	75.6	27.9	28.0	16.1	—	61.8	65.8	73.3	75.6
Kurniawan et al.	38.5	28.3	16.1	20.6	63.5	69.2	**74.5**	**77.7**	48.3	36.4	34.6	38.4	74.1	78.3	80.6	83.7
SubDP (ours)	**41.3**	**38.9**	**31.2**	**33.5**	**71.7**	**70.4**	71.0	75.0	**63.8**	**58.3**	**54.3**	**56.9**	**82.8**	**83.9**	**84.8**	**88.2**

Table 1: Labeled attachment scores (LAS) and unlabeled attachment scores (UAS) on the Universal Dependencies v2.2 (Nivre et al., 2020) standard test set, transferring from English. Following Kurniawan et al. (2021), our results are averaged across 5 runs with different random seeds; the best number in each column is in boldface.

containing a CRISS language token other than the target language, which are likely to be false translation into another language, and (3) those with 80% or more words appearing in the translated sentence more than once, which are likely to be repetitions.

Transferring from an English parser, SubDP achieves the best UAS across all eight target languages, and the best LAS on six languages out of eight. In addition, we find that SubDP is consistent across random seeds, with a standard deviation less than 0.8 for every number in Table 1.

4.2 Ablation Study

We introduce the following baselines with the same annotated data availability for an ablation study:

1. **Direct transfer of English models (DT).** We train a bi-affine dependency parser on English treebanks, and test the model on other languages. This approach is expected to outperform a random baseline as it has a pretrained cross-lingual language model-based feature extractor, which may implicitly enable cross-lingual transfer. For this baseline, we test both XLM-R and CRISS encoders, as SubDP benefits from both models.
2. **Self-training (ST).** Following Kurniawan et al. (2021), we apply an XLM-R DT parser to the target language,[11] and train another parser on the predicted hard trees.
3. **Hard projection (Hard).** It is intuitive to compare SubDP against the hard tree projection baseline (Lacroix et al., 2016), where we use the same set of bitext and alignments to project trees to the target languages, keeping only the edges with both sides aligned in a one-to-one alignment. We use the projected trees to train a parser in the target language.
4. **Random target parser initialization (RandI).** Instead of using the trained English model as the initialization of target parsers, we randomly initialize the weights in this baseline. This approach matches with SubDP in every component except the target parser initialization.

All of the baselines use bi-affine dependency parsers, with pretrained cross-lingual language models (XLM-R or CRISS) as feature extractors.

We compare the LAS between SubDP and the baselines above (Figure 2), and find that

- Across all languages, SubDP significantly outperforms DT with either XLM-R or CRISS word feature extractor. ST does improve over DT consistently, but is much less competitive than SubDP. This indicates that the gain of SubDP over prior work is not simply from more powerful word features.
- While hard treebank projection using the method proposed by Lacroix et al. (2016) is quite competitive, SubDP consistently produces competitive (Arabic, German, Spanish) or better (Hindi, Korean, Turkish, French, Italian) results.
- Comparing SubDP to RandI, we find that initializing the target language parser with a trained source language (English in this work) parser helps improve performance across the board; therefore, source parser initialization should be considered as a general step in future work on zero-shot cross-lingual dependency parsing.

4.3 Analysis: Effect of Alignment Methods

Since most existing work has used only one-to-one alignment for annotation projection (Ma and Xia, 2014; Lacroix et al., 2016; Rasooli et al., 2021, *inter alia*), we would like to analyze the effect of introducing many-to-one alignment edges in SubDP.

[11] We only consider XLM-R as the feature extractor for ST as it achieves better average DT results.

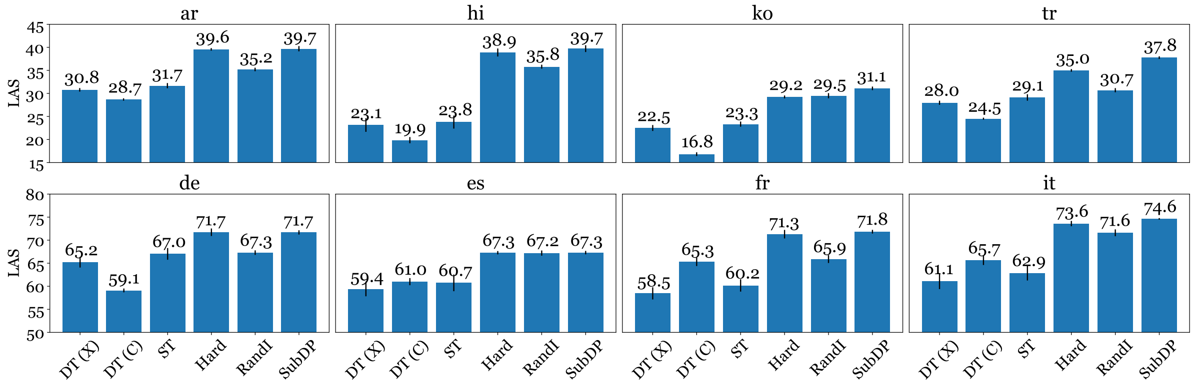


Figure 2: LAS on the Universal Dependencies v2.2 standard development set. The standard deviations are denoted by black lines at the top of bars. All numbers are averaged across 5 runs. Corresponding UAS plots can be found in Appendix F. DT(X): direct transfer by XLM-R representations; DT (C): direct transfer by CRISS representations.

Lang.	`BPE argmax` LAS	UAS	`1:1 only` LAS	UAS
ar	39.7	60.7	**40.2**	**61.1**
hi	**39.7**	**57.4**	38.7	56.5
ko	**31.1**	**51.3**	27.3	49.6
tr	**37.8**	**56.7**	33.3	55.8
avg. *distant*	**37.1**	**56.5**	34.8	55.8
de	71.7	81.6	**72.6**	**83.8**
es	67.3	79.7	**70.4**	**84.2**
fr	71.8	85.3	**72.6**	**87.7**
it	74.6	85.9	**76.0**	**88.8**
avg. *nearby*	71.4	83.1	**72.9**	**86.1**

Table 2: LAS and UAS on the Universal Dependencies v2.2 (Nivre et al., 2020) standard development set, averaged across 5 runs with different random seeds. 1:1 only denotes the filtered one-to-one alignments. The best LAS and UAS for each language are in boldface.

We filter SimAlign BPE `argmax` to obtain a more conservative version, dropping all many-to-one edges (i.e., those that have a word linked to multiple edges),[12] and compare it to the BPE `argmax` algorithm (Table 2).

While the confident one-to-one alignment achieves further improvement on Arabic and all four nearby languages, we find that the many-to-one BPE `argmax` alignment is important to the superior transfer performance on Hindi, Korean, and Turkish. Given the fact that the scores are quite similar for Arabic, the results generally suggest using the many-to-one SimAlign BPE `argmax` alignments for transferring from English to distant languages, while using the more confident one-to-one alignments for nearby languages.

Method	de	es	fr	it
Zhang and Barzilay (2015)	62.5	78.0	**78.9**	79.3
Guo et al. (2016)	65.0	79.0	77.7	78.5
Schuster et al. (2019)‡	61.7	76.6	76.3	77.1
DT (XLM-R)‡,∗	73.1	82.2	75.5	79.5
SubDP (XLM-R)‡,∗	**78.5**	72.1	73.1	74.3
DT w/ SubDP init.‡,∗	76.1	**82.6**	77.7	**81.9**

Table 3: LAS on Universal Dependencies v2.0 (McDonald et al., 2013) standard test set. ‡: methods with minimal annotation. ∗: results from our experiments; other results are taken from Schuster et al. (2019). The best number for each language is in boldface.

4.4 Results: Multiple Source Languages

Following Schuster et al. (2019), we use Universal Dependencies v2.0 (McDonald et al., 2013) to evaluate zero-shot cross-lingual transfer from multiple source languages (Table 3).[13] For each language among German (de), Spanish (es), French (fr), Italian (it), Portuguese (pt), and Swedish (sv), annotated treebanks from all other languages and English can be used for training and development purposes. For SubDP, we generate bitext from all applicable source languages with CRISS.

SubDP outperforms the previous state-of-the-art on German by 13.5 LAS, but under-performs the DT baseline on the other three languages. However, if we start with a trained SubDP parser for a target language, and use the standard training data (i.e., treebanks in other languages) to further train a bi-

[12] This approach is different from Hard as it takes soft source trees as the input, yielding soft target trees as silver labels to train target language parsers.

[13] We do not report performance for Portuguese and Swedish as they are not covered by CRISS; however, the annotated treebanks in these languages are used as source treebanks when applicable.

affine dependency parser (DT w/ SUBDP init.), we are able to achieve better results than DT across the board, obtaining competitive or even better LAS than methods that use extra annotations other than source treebanks (Zhang and Barzilay, 2015; Guo et al., 2016).

4.5 Results: Transfer with Supervised Bitext

We further evaluate SUBDP in another scenario where a few bitext pairs are available. We consider a larger set of eighteen target languages, including Arabic (ar), Czech (cs), German (de), Spanish (es), Finnish (fi), French (fr), Hindi (hi), Hugarian (hu), Italian (it), Japanese (ja), Korean (ko), Norwegian (no), Portuguese (pt), Russian (ru), Tamil (ta), Telugu (te), Vietnamese (vi), and Chinese (zh). We transfer from English to each target language with Wikimatrix bitext (Schwenk et al., 2021), where the examples are mined with an encoding similarity based bitext miner trained with annotated bitext. We vary the number of Wikimatrix bitext pairs, selecting the number of pairs within the geometric sequence $\{50 \times 2^k\}_{k=0}^{9}$, leaving 10% of the examples for development.

On average and for nearby languages (Figure 3), we find that the performance of SUBDP with 50 pairs of bitext is quite close to that with 25K pairs of bitext. Although some distant languages generally require more bitext for further improvement, SUBDP outperforms the direct transfer baseline by a nontrivial margin with a small amount (e.g., 800-1.6K pairs) of bitext.

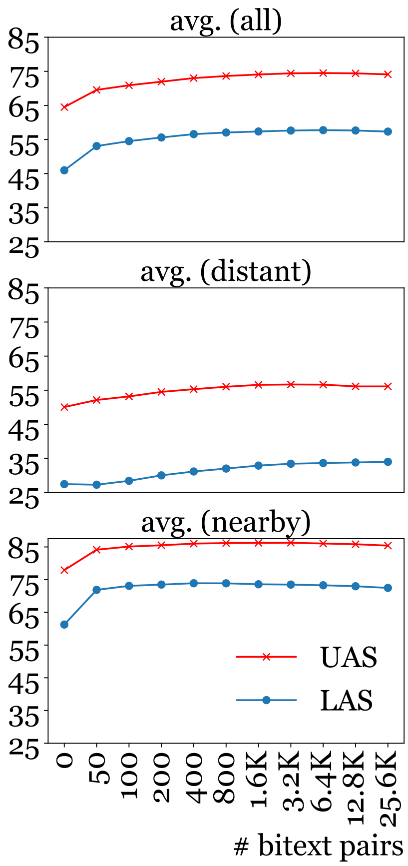


Figure 3: Averaged LAS and UAS on the Universal Dependencies v2.2 standard development set, with respect to the number of bitext pairs. For each language, we run 5 times with different random seeds. The x-axis is on a log scale. Using zero bitext pairs corresponds to the direct transfer (DT; §4.2) baseline. All European languages are categorized as nearby languages, while the remaining are treated as distant languages. Plots for individual languages can be found in Appendix E.

5 Discussion

Our work is in line with recent work (Rasooli et al., 2021) which shows that cross-lingual transfer can be done effectively with weak supervision such as Wikipedia links. Our results go further and study the setting of zero additional supervision beyond the source language treebank, demonstrating the potential of zero-shot cross-lingual dependency parsing with zero additional supervision, even between distant languages that do not share vocabulary or subwords. Our work suggests a new protocol for dependency annotation of low-resource languages: (1) train a pretrained multilingual model following existing work such as XLM-R (Conneau et al., 2020) and CRISS (Tran et al., 2020), (2) annotate a small number of bitext pairs or generate bitext with trained unsupervised translation models, and (3) train a zero-shot cross-lingual dependency parser using SUBDP.

Our contribution to zero-shot cross-lingual dependency parsing is arguably orthogonal to contextualized representation alignment (Schuster et al., 2019; Wang et al., 2019), where pretrained multilingual language models are finetuned for better transfer. In contrast, we use the frozen pretrained models to extract features. In addition, projection quality controls by heuristic rule–based filtering (Rasooli and Collins, 2015) may also be combined with SUBDP to further improve the performance.

Our results, on the other hand, demonstrate that multilingual pretrained models may have more applications beyond representation-based direct transfer—information extracted from these models without further supervision (e.g., word alignment in this work) may further benefit downstream tasks (e.g., zero-shot cross-lingual dependency parsing in this work) with appropriate usage.

While this work depends on pretrained multilingual models such as CRISS (Tran et al., 2020), which require extensive computational resources to train from scratch, SubDP may be applied whenever bitext alignment and cross-lingual word embeddings are available. In addition, the required pretrained cross-lingual models are useful for general purposes, and can be applied to other downstream NLP tasks.

We suggest that SubDP can be extended to other scenarios wherever relevant parallel signals are available, such as cross-lingual named entity recognition, cross-lingual constituency parsing, or zero-shot scene graph parsing for images using only the dependency supervision in text. We leave the further exploration of SubDP on other tasks for future work.

Acknowledgments

We thank Kartik Goyal and Shubham Toshniwal for helpful suggestions on this work. This work is supported in part by a Google Fellowship to FS.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019a. 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, Zhisong Zhang, Xuezhe Ma, Kai-Wei Chang, and Nanyun Peng. 2019b. Cross-lingual dependency parsing with unlabeled auxiliary languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 372–382, Hong Kong, China. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Yoeng-Jin Chu and Tseng-hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 50–61, Edinburgh, Scotland, UK. 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.

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.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 113–122, Beijing, China. Association for Computational Linguistics.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, Jeju Island, Korea. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Jack Edmonds. 1968. Optimum branchings. *Mathematics and the Decision Sciences, Part*, 1(335-345):25.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. 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 1234–1244, Beijing, China. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Junxian He, Zhisong Zhang, Taylor Berg-Kirkpatrick, and Graham Neubig. 2019. Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3223, Florence, Italy. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(3):311–325.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*.

Kemal Kurniawan, Lea Frermann, Philip Schulz, and Trevor Cohn. 2021. PPT: Parsimonious parser transfer for unsupervised cross-lingual adaptation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2907–2918, Online. Association for Computational Linguistics.

Ophélie Lacroix, Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Frustratingly easy cross-lingual transfer for transition-based dependency parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1058–1063, San Diego, California. Association for Computational Linguistics.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Soft cross-lingual syntax projection for dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 783–793, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348, Baltimore, Maryland. Association for Computational Linguistics.

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

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Tao Meng, Nanyun Peng, and Kai-Wei Chang. 2019. Target language-aware constrained inference for cross-lingual dependency parsing. 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 1117–1128, Hong

Kong, China. Association for Computational Linguistics.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, Minneapolis, Minnesota. 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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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.

Mohammad Sadegh Rasooli, Chris Callison-Burch, and Derry Tanti Wijaya. 2021. "wikily" neural machine translation tailored to cross-lingual tasks. *CoRR*, abs/2104.08384.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 328–338, Lisbon, Portugal. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Michael Collins. 2017. Cross-lingual syntactic transfer with limited resources. *Transactions of the Association for Computational Linguistics*, 5:279–293.

Mohammad Sadegh Rasooli and Michael Collins. 2019. Low-resource syntactic transfer with unsupervised source reordering. 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 3845–3856, Minneapolis, Minnesota. Association for Computational Linguistics.

Rudolf Rosa and Zdeněk Žabokrtský. 2015. KLcpos3 - a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, Beijing, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

Philip Schulz, Wilker Aziz, and Khalil Sima'an. 2016. Word alignment without NULL words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 169–174, Berlin, Germany. Association for Computational Linguistics.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2021a. Substructure substitution: Structured data augmentation for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3494–3508, Online. Association for Computational Linguistics.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021b. Bilingual lexicon induction via unsupervised bitext construction and word alignment. 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 813–826, Online. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada. Association for Computational Linguistics.

Robert Endre Tarjan. 1977. Finding optimum branchings. *Networks*, 7(1):25–35.

Jörg Tiedemann. 2015. Improving the cross-lingual projection of syntactic dependencies. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 191–199, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

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

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140, Ann Arbor, Michigan. Association for Computational Linguistics.

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel. 2020. A cross-task analysis of text span representations. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 166–176, Online. Association for Computational Linguistics.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Proc. of NeurIPS*.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. 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 5721–5727, Hong Kong, China. Association for Computational Linguistics.

Michael Wick, Pallika Kanani, and Adam Pocock. 2016. Minimally-constrained multilingual embeddings via artificial code-switching. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Qianhui Wu, Zijia Lin, Börje Karlsson, Jian-Guang Lou, and Biqing Huang. 2020. Single-/multi-source cross-lingual NER via teacher-student learning on unlabeled data in target language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6505–6514, Online. Association for Computational Linguistics.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1285–1294.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2019. Cross-lingual dependency parsing using code-mixed TreeBank. 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 997–1006, Hong Kong, China. Association for Computational Linguistics.

Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Efficient second-order TreeCRF for neural dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3295–3305, Online. Association for Computational Linguistics.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

A Proofs of the Propositions in the Main Content

In this section, we show that both $P_2(\cdot \mid \cdot)$ and $P_2(\cdot \mid \cdot \to \cdot)$ are probability distributions, where the key idea is applying the sum-product algorithm.

A.1 Distribution property of $P_2(\cdot \mid \cdot)$

Proposition 1 Suppose that $P_1(\cdot \mid s_i)$ is a probability distribution for any s_i, and that $\boldsymbol{A}^{t\to s}$ and $\boldsymbol{A}^{s\to t}$ are right-stochastic matrices (i.e., each row of the matrices defines a probability distribution). Let $P_2(t_p \mid t_q) = \sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(s_j \mid s_i)\boldsymbol{A}_{j,q}^{s\to t}$. We have that $P_2(\cdot \mid t_p)$ is a distribution for any t_p.

Proof. First, for any combination of i, j, p, q, we have that $\boldsymbol{A}_{p,i}^{t\to s} \geq 0$, $P_1(s_j \mid s_i) \geq 0$, $\boldsymbol{A}_{j,q}^{s\to t} \geq 0$, therefore,

$$P_2(t_q \mid t_p) = \sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(s_j \mid s_i)\boldsymbol{A}_{j,q}^{s\to t} \geq 0$$

On the other hand,

$$\begin{aligned}
&\sum_{q=1}^{|t|+1} P_2(t_q \mid t_p)\\
=&\sum_{q=1}^{|t|+1}\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(s_j \mid s_i)\boldsymbol{A}_{j,q}^{s\to t}\\
=&\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(s_j \mid s_i)\left(\sum_{q=1}^{|t|+1}\boldsymbol{A}_{j,q}^{s\to t}\right)\\
=&\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(s_j \mid s_i)\\
=&\sum_{i=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\left(\sum_{j=1}^{|s|+1} P_1(s_j \mid s_i)\right)\\
=&\sum_{i=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\\
=&1.
\end{aligned}$$

□

A.2 Distribution property of $P_2(\cdot \mid \cdot \to \cdot)$

Preposition 2 Suppose that $P_1(\cdot \mid s_j \to s_i)$ is a probability distribution for any combination of s_i and s_j, and that $\boldsymbol{A}^{t\to s}$ is a right-stochastic matrix. Let $P_2(\ell \mid t_q \to t_p) = \sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(\ell \mid s_j \to s_i)\boldsymbol{A}_{q,j}^{t\to s}$. We have that $P_2(\cdot \mid t_q \mid t_p)$ is a probability distribution for any t_p and t_q.

Proof. Similarly to the proof in §A.1, it is easy to show that for any ℓ, t_p, t_q,

$$P_2(\ell \mid t_q \to t_p) \geq 0.$$

We next consider the sum over ℓ for a specific pair of t_p and t_q, where we have

$$\begin{aligned}
&\sum_{\ell=1}^{|L|} P_2(\ell \mid t_q \to t_p)\\
=&\sum_{\ell=1}^{|L|}\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s} P_1(\ell \mid s_j \to s_i)\boldsymbol{A}_{q,j}^{t\to s}\\
=&\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\boldsymbol{A}_{q,j}^{t\to s}\left(\sum_{\ell=1}^{|L|} P_1(\ell \mid s_j \to s_i)\right)\\
=&\sum_{i=1}^{|s|+1}\sum_{j=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\boldsymbol{A}_{q,j}^{t\to s}\\
=&\sum_{i=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\left(\sum_{j=1}^{|s|+1}\boldsymbol{A}_{q,j}^{t\to s}\right)\\
=&\sum_{i=1}^{|s|+1} \boldsymbol{A}_{p,i}^{t\to s}\\
=&1.
\end{aligned}$$

□

B Properties of Dependency Distribution Projection

Preposition 3 Dependency distribution projection reduces to hard projection (Lacroix et al., 2016) when (1) the source is a hard parse tree, and (2) there are only one-to-one word alignment.

Proof. We prove the preposition for arc distributions here, which can be immediately generalized to label distributions due to the discreteness property.

For a pair of bitext $\langle s, t\rangle$, under hard projection (Lacroix et al., 2016), there exists an edge from t_q to t_p when and only when there exist i, j such that (1) there exists an edge from s_j to s_i, (2) s_i is aligned to t_p, and (3) s_j is aligned to t_q. It is worth noting that for any pair of p, q, there is at most one pair of $\langle i, j\rangle$ satisfying the above conditions (otherwise it violates the one-to-one alignment assumption).

We consider the case of SUBDP. If there exists a (unique) pair of $\langle i, j\rangle$ that satisfies all the aforementioned three conditions, we have

$$
\begin{aligned}
P_1(s_j \mid s_i) &= 1, \\
\boldsymbol{A}^{t\to s}_{p,i} = 1, &\quad \boldsymbol{A}^{t\to s}_{p,i'} = 0 (i' \neq i), \\
\boldsymbol{A}^{s\to t}_{j,q} = 1, &\quad \boldsymbol{A}^{s\to t}_{j',q} = 0 (j' \neq j).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\hat{P}_2(t_q \mid t_p) &= \sum_{i''=1}^{|s|+1} \sum_{j''=1}^{|s|+1} \boldsymbol{A}^{t\to s}_{p,i''} P_1(s_{j''} \mid s_{i''}) \boldsymbol{A}^{s\to t}_{j'',q} \\
&= \boldsymbol{A}^{t\to s}_{p,i} P_1(s_j \mid s_i) \boldsymbol{A}^{s\to t}_{j,q} \\
&= 1
\end{aligned}
$$

On the other hand, if there do not exist a pair of $\langle i, j\rangle$ that satisfies all three conditions, for any pair of $\langle i, j\rangle$, at least one of the following is true,

$$
\begin{aligned}
P_1(s_j \mid s_i) &= 0, \\
\boldsymbol{A}^{t\to s}_{p,i} &= 0, \\
\boldsymbol{A}^{s\to t}_{j,q} &= 0.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\hat{P}_2(t_q \mid t_p) &= \sum_{i''=1}^{|s|+1} \sum_{j''=1}^{|s|+1} \boldsymbol{A}^{t\to s}_{p,i''} P_1(s_{j''} \mid s_{i''}) \boldsymbol{A}^{s\to t}_{j'',q} \\
&= 0.
\end{aligned}
$$

That is, SUBDP has the same behavior as Lacroix et al. (2016) under the given assumptions. □

Preposition 4 Given a hard source tree, SUBDP assigns non-zero probability to any dependency arc generated by hard projection (Lacroix et al., 2016).

Proof. Similarly to the proof to Preposition 3, if hard projection generates an arc $t_q \to t_p$, there exists a pair of $\langle i, j\rangle$ such that

$$
\begin{aligned}
P_1(s_j \mid s_i) &= 1, \\
\tilde{\boldsymbol{A}}_{i,p} = 1 &\Rightarrow \boldsymbol{A}^{t\to s}_{p,i} > 0, \\
\tilde{\boldsymbol{A}}_{j,q} = 1 &\Rightarrow \boldsymbol{A}^{s\to t}_{j,q} > 0,
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\hat{P}_2(t_q \mid t_p) &= \sum_{i''=1}^{|s|+1} \sum_{j''=1}^{|s|+1} \boldsymbol{A}^{t\to s}_{p,i''} P_1(s_{j''} \mid s_{i''}) \boldsymbol{A}^{s\to t}_{j'',q} \\
&\geq \boldsymbol{A}^{t\to s}_{p,i} P_1(s_j \mid s_i) \boldsymbol{A}^{s\to t}_{j,q} > 0.
\end{aligned}
$$

This can be immediately generalized to label distribution due to the discreteness of the input tree. □

C Intuition on Dummy Positions and Partial Cross Entropy

In this section, we provide more intuition on the added dummy positions (§3.3), and the partial cross entropy optimization (§3.4) used in SUBDP.

Consider an alternative approach $\mathcal{A}$, which projects a source tree distribution by the following steps, taking arc distribution projection as an example:

1. Given $\tilde{\boldsymbol{A}}$, obtain source-to-target and target-to-source alignment matrices $\bar{\boldsymbol{A}}^{s\to t} = \mathcal{N}^{\mathcal{R}}(\tilde{\boldsymbol{A}})$ and $\bar{\boldsymbol{A}}^{t\to s} = \mathcal{N}^{\mathcal{R}}(\tilde{\boldsymbol{A}}^\intercal)$ without adding dummy positions, keeping the zero rows unchanged when applying $\mathcal{N}^{\mathcal{R}}(\cdot)$.

2. Project the source distributions to target by

$$
\bar{P}_2(t_q \mid t_p) = \sum_{i=1}^{|s|} \sum_{j=1}^{|s|} \bar{\boldsymbol{A}}^{t\to s}_{p,i} P_1(s_j \mid s_i) \bar{\boldsymbol{A}}^{s\to t}_{j,q}
$$

   Note that $\bar{P}_2(\cdot \mid \cdot)$ is not guaranteed to be a well formed distribution due to the potential existence of zero rows/columns in $\tilde{\boldsymbol{A}}$.

3. Normalize $\bar{P}_2(\cdot \mid t_p)$ to $\tilde{P}_2(\cdot \mid t_p)$ for each p separately, ignoring every "*zero position*" p that $\sum_q \bar{P}_2(t_q \mid t_p) = 0$.

4. Compute the cross entropy loss between the target parser probability P_2 and $\tilde{P}_2$ for all *non-zero positions* p.

We argue that SUBDP is equivalent to a weighted sum version to the above approach: that is, there exists a group of weight $(\alpha_1, \ldots, \alpha_{|t|})$ such that the SUBDP arc loss $\mathcal{L}^{(t)}_{arc}(P_2, \hat{P}_2) = \sum_{p=1}^{|t|} \alpha_p H(\tilde{P}_2(\cdot \mid t_p), P_2(\cdot \mid t_p))$, where $H(\cdot,\cdot)$ denotes cross entropy, and $H(\cdot,\cdot) := 0$ when the first argument is a ill-formed zero "distribution".

Proof First, we note that for all $p = 1, \ldots |t|$ and $i = 1, \ldots, |s|$,

$$
\bar{\boldsymbol{A}}^{t\to s}_{p,i} = \boldsymbol{A}^{t\to s}_{p,i},
$$

$$
\bar{\boldsymbol{A}}^{s\to t}_{i,p} = \boldsymbol{A}^{s\to t}_{i,p},
$$

as adding dummy positions does not affect the row normalization result for non-dummy positions.

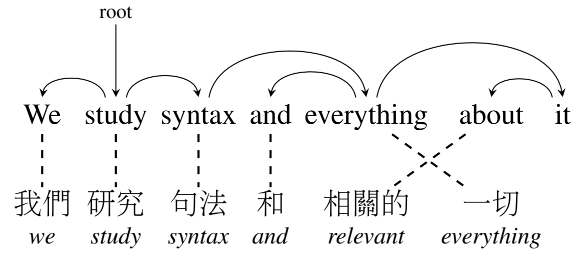
Therefore,

$$
\begin{aligned}
\hat{P}_2(t_q \mid t_p) &= \sum_{i=1}^{|s|+1} \sum_{j=1}^{|s|+1} \boldsymbol{A}^{t \to s}_{p,i} P_1(s_j \mid s_i) \boldsymbol{A}^{s \to t}_{j,q} \\
&= \sum_{i=1}^{|s|} \sum_{j=1}^{|s|+1} \boldsymbol{A}^{t \to s}_{p,i} P_1(s_j \mid s_i) \boldsymbol{A}^{s \to t}_{j,q} + \\
&\quad \sum_{j=1}^{|s|+1} \boldsymbol{A}^{t \to s}_{p,|s|+1} P_1(s_j \mid s_{|s|+1}) \boldsymbol{A}^{s \to t}_{j,q} \\
&= \sum_{i=1}^{|s|} \sum_{j=1}^{|s|} \boldsymbol{A}^{t \to s}_{p,i} P_1(s_j \mid s_i) \boldsymbol{A}^{s \to t}_{j,q} + \\
&\quad \sum_{i=1}^{|s|} \boldsymbol{A}^{t \to s}_{p,i} P_1(s_{|s|+1} \mid s_i) + \\
&\quad \boldsymbol{A}^{t \to s}_{p,|s|+1} \boldsymbol{A}^{s \to t}_{|s|+1,q} \qquad (6) \\
&= \sum_{i=1}^{|s|} \sum_{j=1}^{|s|} \bar{\boldsymbol{A}}^{t \to s}_{p,i} P_1(s_j \mid s_i) \bar{\boldsymbol{A}}^{s \to t}_{j,q} \\
&= \bar{P}_2(t_q \mid t_p).
\end{aligned}
$$

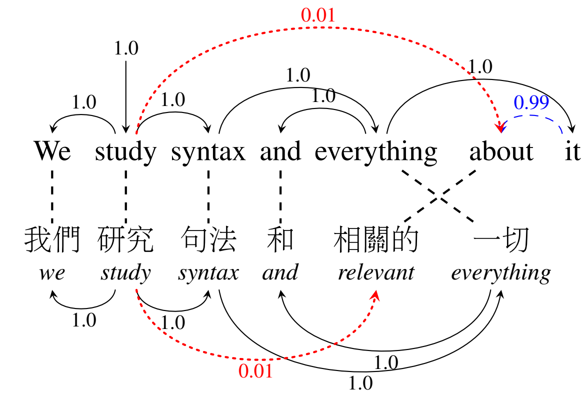
The last two terms in Eq (6) can be dropped since $P_1(s_{|s|+1} \mid s_i) = 0$ for any $i(1 \le i \le |s|)$, and $\boldsymbol{A}^{s \to t}_{|s|+1,q} = 0$ for any $q(1 \le q \le |s|)$. That is, $\tilde{P}_2(\cdot \mid t_p)$, normalization of $\hat{P}_2(\cdot \mid t_p)$, can be also calculated by normalization of $\bar{P}_2(\cdot \mid t_p)$, where $q = 1, \ldots, |t|$.[14] Therefore, for any $p = 1, \ldots, |t|$, there exists α_p such that $\hat{P}_2(\cdot \mid t_p) = \alpha_p \tilde{P}_2(\cdot \mid t_p)$.

By definition,

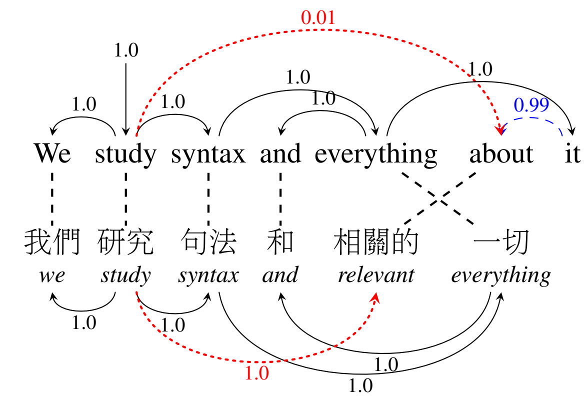
$$
\begin{aligned}
&\mathcal{L}^{(t)}_{arc}(P_2, \hat{P}_2) \\
&= -\sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \hat{P}_2(t_q \mid t_p) \log P_2(t_q \mid t_p) \\
&= -\sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \alpha_p \tilde{P}_2(t_q \mid t_p) \log P_2(t_q \mid t_p) \\
&= \sum_{p=1}^{|t|} \alpha_p H(\tilde{P}_2(\cdot \mid t_p), P_2(\cdot \mid t_p)).
\end{aligned}
$$



(a) Ground-truth unlabeled parse tree and alignment.



(b) Projection by SUBDP of the arc distributions predicted by a neural parser. Label denotes edge probability.



(c) Projection by the alternative algorithm $\mathcal{A}$ of the arc distributions. Label denotes edge probability.

Figure 4: Intuition on the reason that we do not apply normalization in SUBDP as described in the alternative algorithm $\mathcal{A}$.

We use a toy example (Figure 4) to show the intuition for using SUBDP instead of the alternative approach $\mathcal{A}$. It is common for neural network–based parsers generate a very low non-zero arc probability for a random word pair with no direct dependency relation, e.g., (study $\to$ about): normalization of arc probability, may significantly enlarge the noise level when the correct arc (it $\to$ about in this case) is not projected due to alignment mismatch, weighting undesirable target language arcs (e.g., 研究$\to$相關的;Figure 4c) as much as those with high quality in the training loss; in contrast, while SUBDP (Figure 4b) may also introduce such noise, the corresponding weight remains in the same scale as the probability predicted by the neural parser.

[14] We may here intuitively view that the dummy position $P(t_{|t|+1} \mid t_p)$ absorbs some original probability corresponding to unaligned words.

D Implementation Details of the Bi-Affine Dependency Parser

Given a sentence s, we extract the subword representations by a pretrained multilingual contextualized representation model (XLM-R or CRISS), and take endpoint concatenation of corresponding subwords representations as word representations, yielding contextualized word features $\boldsymbol{V} \in \mathbb{R}^{|s| \times d}$, where $|s|$ denotes the number of words in s, and d denotes the dimensionality of the extracted features. We further perform non-linear transformation on the features with multi-layer perceptrons (MLPs) with ReLU activation and a long short-term memory module (LSTM; Hochreiter and Schmidhuber, 1997), to obtain head and dependent features $\boldsymbol{H}$ and $\boldsymbol{D}$:[15]

$$\begin{aligned} \tilde{\boldsymbol{V}} &= \text{LSTM}(\text{MLP}_{\textit{feature}}(\boldsymbol{V})) \\ \boldsymbol{H} &= \text{MLP}_{\textit{head}}(\tilde{\boldsymbol{V}}) \\ \boldsymbol{D} &= \text{MLP}_{\textit{dependent}}(\tilde{\boldsymbol{V}}). \end{aligned}$$

E Cross-Lingual Transfer Results on Individual Languages

We present the SUBDP zero-shot cross-lingual dependency parsing performance for each individual language with respect to the numbers of bitext pairs (Figure 5). SUBDP with supervised bitext outperforms the direct transfer baseline (using 0 pair of bitext) for all languages. For most languages, SUBDP starts improves over direct transfer with only 50 pairs of bitext.

F Ablation Study in UAS

We present the corresponding UAS results to the LAS in Figure 2 in Figure 6. We arrive at similar conclusions to those reached by LAS trends: SUBDP is the only model that consistently ranks among the top contenders and outperforms the direct transfer baseline in all languages.

G Treebank Selection on Universal Dependencies

We use the same UD v2.2 treebanks as Kurniawan et al. (2021) for the eight main languages for fair comparison,[16] and select treebanks for additional languages based on domain similarity and associated quality score provided by the Universal Dependencies project (Nivre et al., 2020). Details are listed in Table 4.

Language	UD Treebank Name
Eight main languages	
Arabic	PADT
German	GSD
Spanish	GSD, AnCora
French	GSD
Hindi	HDTB
Korean	GSD, Kaist
Italian	ISDT
Turkish	IMST
Additional languages	
Czech	PDT
Finnish	TDT
Hungarian	Szeged
Japanese	GSD
Norwegian	Nynorsk
Portuguese	GSD
Russian	Syntagrus
Simplified Chinese	GSD
Tamil	TTB
Telugu	MTG
Vietnamese	VTB

Table 4: Treebank selection on the Universal Dependencies v2.2 (Nivre et al., 2020), following (Kurniawan et al., 2021).

[15] We find that the LSTM module is important, removing it will result in 1-2 points drop in terms of both UAS and LAS, in the supervised training settings for English.

[16] https://github.com/kmkurn/ppt-eacl2021/blob/master/readers.py

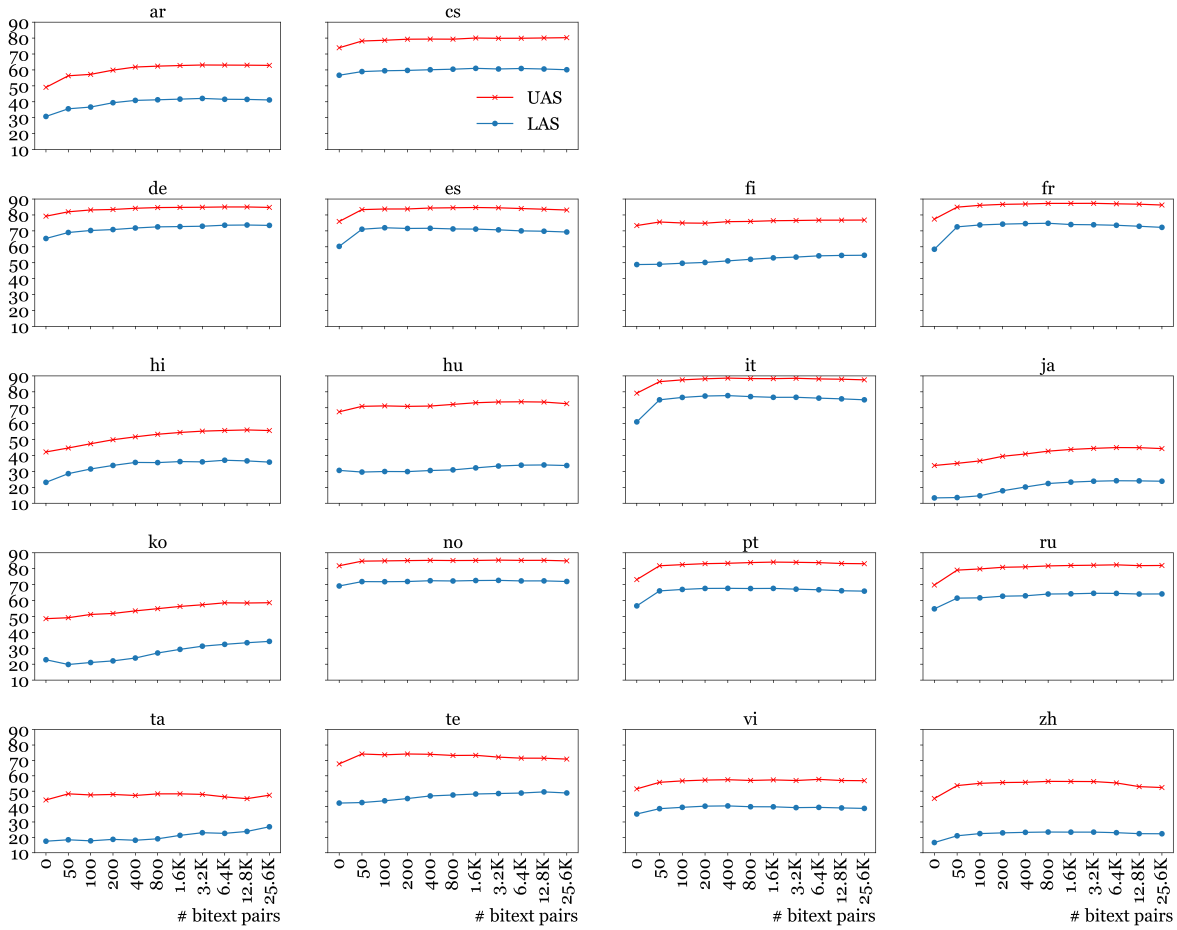


Figure 5: LAS and UAS on the Universal Dependencies v2.2 standard development sets (best viewed in color), where the x-axis represents the number of bitext pairs used, and y-axis corresponds to attachment scores. All numbers are averaged across 5 runs with different random seeds and different sets of bitext if applicable.

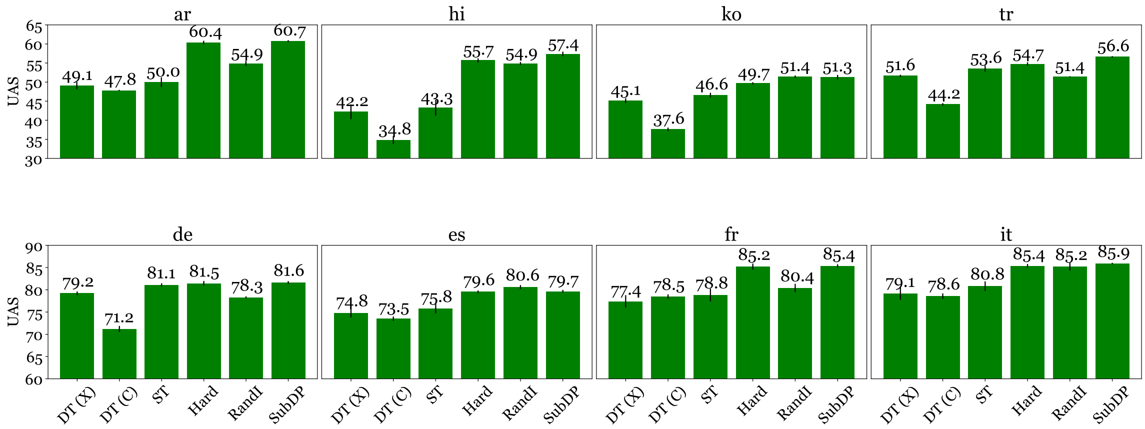


Figure 6: UAS on the Universal Dependencies v2.2 standard development set. All numbers are averaged across 5 runs.