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

Anonymous ACL submission

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.

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., / and 我 in Figure 1; Ma and Xia, 2014; Lacroix et al., 2016; Rasooli et al., 2021, \emph{inter alia}), discarding information in many-to-one alignments (e.g., \emph{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 \emph{hard trees}; in contrast, we refer to collections of per-word head distributions and per-arc label distributions with continuous probabilities as \emph{soft trees}.
ability distribution of the corresponding head given a word.\footnote{Projection of the distribution over whole parse trees has been considered by Ma and Xia (2014), while \textsc{SubDP} has a much lower time complexity – see \S 2 for more discussion.} When the source parse is a hard tree, \textsc{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, \textsc{SubDP} projects the corresponding arcs into soft arc distributions in the target language. Therefore, in \textsc{SubDP}, a target language word may have multiple heads in the projected trees, where their probabilities sum to one. More generally, \textsc{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 \textsc{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, \textsc{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 \textsc{SubDP} also helps improve zero-shot cross-lingual dependency parsing with a small amount of supervised bitext, across a broader range of target languages.

\section{Related Work}

Zero-shot cross-lingual dependency parsing.\footnote{Also referred to as zero-shot dependency parsing in recent literature (Schuster et al., 2019; Wang et al., 2019).} Existing approaches can be classified into the following categories:

1. \textbf{Delexicalized training} (Zeman and Resnik, 2008; McDonald et al., 2011; Cohen et al., 2011; Durrett et al., 2012; Rosa and Žabokrtský, 2015, \textit{inter alia}), which only considers delexicalized features (e.g., part-of-speech tags) in training.

2. \textbf{Transfer with cross-lingual embeddings} (Täckström et al., 2012; Guo et al., 2015; Schuster et al., 2019, \textit{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. \textbf{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. \textbf{Annotation projection},\footnote{We use \textit{annotation projection} to denote the projection of predicted parses following Rasooli and Collins (2019) and Zhang et al. (2019), and \textit{treebank translation} for the projection of human-annotated trees following Tiedemann et al. (2014).} 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, \textit{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, \textit{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, \textit{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, \textsc{SubDP}, falls into the category of annotation projection. Some of the
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 \(O(n^2)\) for non-projective parsing.\(^3\) SUBDP can be easily extended to other tasks, such as sequence labeling, where we can define substructures (Shi et al., 2021) 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), 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 labels) 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 \(P_1\) 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 \(P_2\) 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 \(\{w_1, \ldots, w_n\}\),\(^4\) we denote the word features when acting as heads and dependents by \(H \in \mathbb{R}^{n \times d_h}\) and \(D \in \mathbb{R}^{n \times d_d}\) 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:

\[
\begin{align*}
S^{(arc)}_{i,j} &= DW^{(arc)}H^T \tag{1} \\
P(w_j | w_i) &= \frac{\exp \left( S^{(arc)}_{i,j} \right)}{\sum_{k=1}^{n} \exp \left( S^{(arc)}_{i,k} \right)}, \tag{2}
\end{align*}
\]

where \(W^{(arc)} \in \mathbb{R}^{d_d \times d_h}\) is the parameters of the bi-affine module.\(^7\) Given \(\log P(w_j | w_i)\) 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 \(O(n^2)\) time complexity for each sentence.

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

\[
S^{(label)}_{i,j,\ell} = \sum_p \sum_q D_{i,p} W^{(label)}_{p,q,\ell}H_{j,q} \tag{3}
\]

\[
P(\ell | w_j \rightarrow w_i) = \frac{\exp \left( S^{(label)}_{i,j,\ell} \right)}{\sum_{k=1}^{|L|} \exp \left( S^{(label)}_{i,j,k} \right)}.
\]

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...
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 \( (s, t) \), SimAlign com-
putes a contextualized representation for each to-
ken 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 \( (i,j) \), 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 al-
gorithm to extract word alignment for SubDP.
It is worth noting that pretrained multilingual mod-
els usually use byte-pair encodings (BPEs; Gage,
1994), a more fine-grained level than words, for to-
kenization. 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 align-
ments, 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 depen-
dency annotations: for a sentence \( \langle w_1, \ldots, w_n \rangle \),
there is a dummy word \( w_1 \), whose unique depen-
dent 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 an-
notations to train an \( L_1 \) bi-affine dependency parser
\( P_1 \), following the procedure described in §3.1.

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

Word alignment. For a bitext pair \( (s, t) \), we gen-
erate the word alignment matrix \( \tilde{A} \in \{0, 1\}^{[s]+[t]} \)
with SimAlign, where \( \tilde{A}_{i,j} = 1 \) 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 \( \Delta(\cdot) \):
\[
\Delta : \mathbb{R}^{r \times c} \rightarrow \mathbb{R}^{(r+1)(c+1)} (\forall r, c \in \mathbb{N}_+ \}
\]
\[
\Delta(M)_{i,j} = M_{i,j}(1 \leq i \leq r, 1 \leq j \leq c); \quad \Delta(M)_{i,c+1} = 0; \quad \Delta(M)_{r+1,j} = 0(1 \leq j \leq c);
\]
\[
\Delta(M)_{r+1,c+1} = 1,
\]
where \( 0[\cdot] = 1 \) when all input values are zero
and otherwise 0.

The row normalization operator \( N^R(\cdot) \):
\[
N^R : \mathbb{R}^{r \times c} \rightarrow \mathbb{R}^{r \times c} (\forall r, c \in \mathbb{N}_+ \}
\]
\[
N^R(M)_{i,j} = \frac{M_{i,j}}{\sum_t M_{i,t}}.
\]
Intuitively, the added dummy positions corre-
spond to null words in the word alignment lit-
erature (Dyer et al., 2013; Schulz et al., 2016;
Jalili Sabet et al., 2020, inter alia). We denote
the source-to-target alignment matrix by \( A^{s \rightarrow t} =
N^R(\Delta(\tilde{A}^t) \), and the target-to-source alignment
matrix by \( A^{t \rightarrow s} = N^R(\Delta(\tilde{A}^s) \). Both are right
stochastic matrices by definition.

3.3 Dependency Distribution Projection

Arc distribution projection. We consider a pair
of bitext \( (s, t) \). Let \( P_1(s_i | s_i) \) denote the arc
probability produced by the parser \( P_1 \). Like the
dummy position notation, we specify a dummy
\( \langle [s]+1 \rangle^t \) word whose head is itself, that is,
\[
P_1(s_i | s_{[s]+1}) = 0, \quad P_1(s_{[s]+1} | s_{[s]+1}) = 1.
\]
We project \( P_1(\cdot | \cdot) \) to \( \hat{P}_2(t_q | t_p) \), the arc proba-
bility distributions in the parallel \( L_2 \) example \( t \),
\[
\hat{P}_2(t_q | t_p) = \sum_{i=1}^{[s]+1} \sum_{j=1}^{[t]+1} A^{t \rightarrow s}_{p_i q_j} P_1(s_j | s_i) A^{s \rightarrow t}_{j q} \quad \text{Eq} (4).
\]
It is guaranteed that \( \hat{P}_2(\cdot | t_p) \) 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 \( \hat{P}_2(t_q | t_p) \) by \( \hat{P}^{(2)}_{p q} \) and denote \( P_1(s_j | s_i) \)
by \( P^{(1)}_{i j} \), Eq (4) is equivalent to
\[
\hat{P}^{(2)} = A^{t \rightarrow s} P^{(1)} A^{s \rightarrow t}.
\]
Label distribution projection. Let \( P_1(\ell \mid s_j \rightarrow s_i) \) denote the label probability produced by \( P_1 \). For dummy positions, we simply add a uniform distribution, that is,
\[
P_1(\ell \mid s_j \rightarrow s_i) = \frac{1}{L} \quad \text{if } i \text{ or } j = |s| + 1.
\]

We project \( P_1(\cdot \mid \cdot \rightarrow \cdot) \) to \( \hat{P}_2(\ell \mid t_q \rightarrow t_p) \), the label distributions in the parallel \( L_2 \) example \( t \), by
\[
\hat{P}_2(\ell \mid t_q \rightarrow t_p) = \sum_{i=1}^{|s|+1} \sum_{j=1}^{|s|+1} A_{p,i}^{t-\rightarrow s} P_1(\ell \mid s_j \rightarrow s_i) A_{q,j}^{t-\rightarrow s}.
\]

\( \hat{P}_2(\cdot \mid t_q \rightarrow t_p) \) 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 \( P_2 \) 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:
\[
L_{arc}(P_2, \hat{P}_2) = -\sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \hat{P}_2(t_q \mid t_p) \log P_2(t_q \mid t_p).
\]

Similarly, the partial label cross entropy loss can be computed as follows:
\[
L_{label}(P_2, \hat{P}_2) = -\sum_{\ell=1}^{|L|} \sum_{p=1}^{|t|} \sum_{q=1}^{|t|} \hat{P}_2(\ell \mid t_q \rightarrow t_p) \log P_2(\ell \mid t_q \rightarrow t_p).
\]

Finally, we train the parameters of \( P_2 \) to minimize
\[
\sum_{(s,\ell) \in B} L_{arc}(P_2, \hat{P}_2) + L_{label}(P_2, \hat{P}_2).
\]

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 \( 2 \times 10^{-3} \) following the baseline implementation by Zhang et al. (2020), and the target language parser is trained for 30 epochs with initial learning rate \( 5 \times 10^{-4} \). We use the loss against silver projected distributions on the development set for \( \text{SubDP} \) and the development LAS against projected trees for baselines for early stopping.\(^8\)

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,\(^9\)

- 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 \( \text{argmax} \) 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 \( \text{SubDP} \) 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.\(^8\) 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 \( \text{SubDP} \) is tuned to optimize the development loss for German, where the German gold trees remain unused.

\(^9\)\( \text{SubDP} \) does not provide a set of hard silver trees for LAS and UAS calculation.

\(^8\)In experiments, we translate English treebank sentences; in more general cases, any source language sentence can be taken for bitext construction.
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, 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>distant languages</th>
<th>nearby languages</th>
<th>distant languages</th>
<th>nearby languages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ar    hi    ko    tr</td>
<td>de    es    fr    it</td>
<td>ar    hi    ko    tr</td>
<td>de    es    fr    it</td>
</tr>
<tr>
<td>Meng et al.</td>
<td>—     —     —     —</td>
<td>—     —     —     —</td>
<td>47.3  52.4  37.1  35.2</td>
<td>70.8  75.8  79.1  82.0</td>
</tr>
<tr>
<td>He et al.</td>
<td>27.9  28.0  16.1  —</td>
<td>61.8  65.8  73.3  75.6</td>
<td>27.9  28.0  16.1  —</td>
<td>61.8  65.8  73.3  75.6</td>
</tr>
<tr>
<td>Ahmad et al.</td>
<td>38.5  28.3  16.1  20.6</td>
<td>63.5  69.2  74.5  77.7</td>
<td>48.3  36.4  34.6  38.4</td>
<td>74.1  78.3  80.6  83.7</td>
</tr>
<tr>
<td>Kurniawan et al.</td>
<td>41.3  38.9  31.2  33.5</td>
<td>71.7  70.4  71.0  75.0</td>
<td>63.8  58.3  54.3  56.9</td>
<td>82.8  83.9  84.8  88.2</td>
</tr>
</tbody>
</table>

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. 

4.2 Ablation Study

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

---

**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, 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; Rasouli et al., 2021, *inter alia*), we would like to analyze the effect of introducing many-to-one alignment edges in SUBDP.
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),\footnote{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.} 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.

### 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).\footnote{We do not report performances 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.} 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-
We further evaluate SUBDP outperforms the direct transfer baseline by a nontrivial margin with a small amount (e.g., 800-1.6K pairs) of bitext.

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 supports a new protocol for dependency annotations 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.

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.
References


A Proofs of the Propositions in the Main Content

In this section, we show that both $P_2(\cdot \mid \cdot)$ and $P_2(\cdot \mid \cdot \rightarrow \cdot)$ 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 $A^{t \rightarrow s}$ and $A^{s \rightarrow t}$ are right-stochastic matrices (i.e., each row of the matrices defines a probability distribution).

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

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

On the other hand,

$$\sum_{q=1}^{t+1} P_2(t_q \mid t_p)$$

$$= \sum_{q=1}^{t+1} \sum_{s=1}^{s+1} \sum_{j=1}^{s+1} A^{t \rightarrow s}_{p,i} P_1(s_j \mid s_i) A^{s \rightarrow t}_{j,q}$$

$$= \sum_{i=1}^{s+1} \sum_{j=1}^{s+1} A^{t \rightarrow s}_{p,i} P_1(s_j \mid s_i) \left( \sum_{q=1}^{t+1} A^{s \rightarrow t}_{j,q} \right)$$

$$= \sum_{i=1}^{s+1} \sum_{j=1}^{s+1} A^{t \rightarrow s}_{p,i} P_1(s_j \mid s_i)$$

$$= \sum_{i=1}^{s+1} A^{t \rightarrow s}_{p,i} \left( \sum_{j=1}^{s+1} P_1(s_j \mid s_i) \right)$$

$$= \sum_{i=1}^{s+1} A^{t \rightarrow s}_{p,i}$$

$$= 1. \quad \Box$$

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

Proposition 2 Suppose that $P_1(\cdot \mid s_j \rightarrow s_i)$ is a probability distribution for any combination of $s_i$ and $s_j$, and that $A^{t \rightarrow s}$ is a right-stochastic matrix.

Let $P_2(\ell \mid t_q \rightarrow t_p) = \sum_{i=1}^{s+1} \sum_{j=1}^{s+1} A^{t \rightarrow s}_{p,i} P_1(\ell \mid s_j \rightarrow s_i) A^{t \rightarrow s}_{q,j}$. We have that $P_2(\cdot \mid t_q \mid t_p)$ is a probability distribution for any $t_p$ and $t_q$.  

\textbf{Proof.} Similarly to the proof in §A.1, it is easy to show that for any \( \ell, t_p, t_q \),
\[
P_2(\ell \mid t_q \rightarrow t_p) \geq 0.
\]

We next consider the sum over \( \ell \) for a specific pair of \( t_p \) and \( t_q \), where we have
\[
\sum_{\ell=1}^{n} P_2(\ell \mid t_q \rightarrow t_p) = \sum_{\ell=1}^{n} \sum_{j=1}^{n} A_{\ell q j}^{t_q s} P_1(\ell \mid s_j \rightarrow s_i) A_{q t j}^{s t}
\]

\[
= \sum_{i=1}^{n} A_{p i j}^{t s} A_{q j}^{t q} \left( \sum_{\ell=1}^{n} P_1(\ell \mid s_j \rightarrow s_i) \right)
\]

\[
= \sum_{i=1}^{n} A_{p i j}^{t s}
\]

\[
= 1.
\]

\( \Box \)

\section{Properties of Dependency Distribution Projection}

\textbf{Proposition 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.

\textbf{Proof.} We prove the proposition 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 above-mentioned three conditions, we have
\[
P_1(s_j \mid s_i) = 1,
\]
\[
A_{p i j}^{t s} = 1,
\]
\[
A_{q j}^{t q} = 0(i' \neq i),
\]
\[
A_{q j}^{s t} = 1,
\]
\[
A_{j q}^{s t} = 0(j' \neq j).
\]

Therefore,
\[
\hat{P}_2(t_q \mid t_p) = \sum_{i'q''=1}^{n+1} \sum_{i''q''=1}^{n+1} A_{p i j}^{t s} P_1(s_j \mid s_i) A_{q j}^{t q} A_{j q}^{s t} = 1.
\]

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

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

\textbf{Proof.} Similarly to the proof to Proposition 3, if hard projection generates an arc \( t_q \rightarrow t_p \), there exists a pair of \( \langle i, j \rangle \) such that
\[
P_1(s_j \mid s_i) = 1,
\]
\[
A_{p i j}^{t s} = 1 \Rightarrow A_{q j}^{t q} > 0,
\]
\[
A_{j q}^{s t} = 1 \Rightarrow A_{j q}^{s t} > 0,
\]

Therefore,
\[
\hat{P}_2(t_q \mid t_p) = \sum_{i'q''=1}^{n+1} \sum_{i''q''=1}^{n+1} A_{p i j}^{t s} P_1(s_j \mid s_i) A_{q j}^{t q} A_{j q}^{s t} \geq A_{p i j}^{t s} P_1(s_j \mid s_i) A_{q j}^{t q} A_{j q}^{s t} > 0.
\]

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

\( \Box \)
\section{Intuition on Dummy Positions and Partial Cross Entropy}

In this section, we provide more intuition on the added dummy positions (\S 3.3), and the partial cross entropy optimization (\S 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 $\hat{A}$, obtain source-to-target and target-to-source alignment matrices $\hat{A}_{s\to t} = \mathcal{N}^R(\hat{A})$ and $\hat{A}^{t\to s} = \mathcal{N}^R(A^t)$ without adding dummy positions, keeping the zero rows unchanged when applying $\mathcal{N}^R(\cdot)$.

2. Project the source distributions to target by

$$\hat{P}_2(t_q \mid t_p) = \sum_{i=1}^{\lfloor |s|/2 \rfloor} \sum_{j=1}^{\lfloor |s|/2 \rfloor} \hat{A}_{p,i,j}^{t\to s} P_1(s_j \mid s_i) \hat{A}_{j,q}^{s\to t}$$

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

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

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”.

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

$$\hat{A}_{p,i}^{t\to s} = A_{p,i}^{t\to s},$$

$$\hat{A}_{i,p}^{s\to t} = A_{i,p}^{s\to t},$$

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

Therefore,

$$\tilde{P}_2(t_q \mid t_p) = \sum_{i=1}^{\lfloor |s|/2 \rfloor} \sum_{j=1}^{\lfloor |s|/2 \rfloor} A_{p,i,j}^{t\to s} P_1(s_j \mid s_i) A_{j,q}^{s\to t}$$

The last two terms in Eq (6) can be dropped since $P_1(s_{|s|+1} \mid s_i) = 0$ for any $i (1 \leq i \leq |s|)$, and $A_{|s|+1,q}^{s\to t} = 0$ for any $q (1 \leq q \leq |s|)$. That is, $\tilde{P}_2(\cdot \mid t_p)$, normalization of $\tilde{P}_2(\cdot \mid t_p)$, can be also calculated by normalization of $P_2(\cdot \mid t_p)$, where $q = 1, \ldots, |t|$.

Therefore, for any $p = 1, \ldots, |t|$, there exists $\alpha_p$ such that $\tilde{P}_2(\cdot \mid t_p) = \alpha_p \hat{P}_2(\cdot \mid t_p)$.

By definition,

$$\mathcal{L}^{(t)}_{arc}(P_2, \tilde{P}_2) = - \sum_{p=1}^{|t|} \alpha_p H(\tilde{P}_2(\cdot \mid t_p), P_2(\cdot \mid t_p))$$

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 to generate a very low non-zero arc probability for a random word pair with no direct dependency relation, e.g., (study $\rightarrow$ about): normalization of $P(t_{|t|+1} \mid t_p)$ absorbs some original probability corresponding to unaligned words.

\footnote{We may here intuitively view that the dummy position $P(t_{|t|+1} \mid t_p)$ absorbs some original probability corresponding to unaligned words.}
We study syntax and everything about it.

我 们 研 究 句 法 和 相 关 的 一 切

**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 $V \in \mathbb{R}^{|s| \times d}$, 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 $H$ and $D$: \[ V = \text{LSTM}(\text{MLP}_{\text{feature}}(V)) \]
\[ H = \text{MLP}_{\text{head}}(\tilde{V}) \]
\[ D = \text{MLP}_{\text{dependent}}(\tilde{V}). \]

**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

---

\(^{15}\)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
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.
Figure 6: UAS on the Universal Dependencies v2.2 standard development set. All numbers are averaged across 5 runs.

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

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