Towards Instance-Level Parser Selection for Cross-Lingual Transfer of Dependency Parsers

Robert Litschko1, Ivan Vulić2, Željko Agić3, Goran Glavaš1
1 Data and Web Science Group, University of Mannheim, Germany
2 Language Technology Lab, University of Cambridge, UK
3 Corti, Copenhagen, Denmark
{litschko,goran}@informatik.uni-mannheim.de
iv250@cam.ac.uk za@corti.ai

Abstract

Current methods of cross-lingual parser transfer focus on predicting the best parser for a low-resource target language globally, that is, “at treebank level”. In this work, we propose and argue for a novel cross-lingual transfer paradigm: instance-level parser selection (ILPS), and present a proof-of-concept study focused on instance-level selection in the framework of delexicalized parser transfer. We start from an empirical observation that different source parsers are the best choice for different Universal POS sequences in the target language. We then propose to predict the best parser at the instance level. To this end, we train a supervised regression model, based on the Transformer architecture, to predict parser accuracies for individual POS-sequences. We compare ILPS against two strong single-best parser selection baselines (SBPS): (1) a model that compares POS n-gram distributions between the source and target languages (KL) and (2) a model that selects the source based on the similarity between manually created language vectors encoding syntactic properties of languages (L2V). The results from our extensive evaluation, coupling 42 source parsers and 20 diverse low-resource test languages, show that ILPS outperforms KL and L2V on 13/20 and 14/20 test languages, respectively. Further, we show that by predicting the best parser “at the treebank level” (SBPS), using the aggregation of predictions from our instance-level model, we outperform the same baselines on 17/20 and 16/20 test languages.

1 Introduction

A major goal and promise of cross-lingual transfer in NLP is to transfer language technology to as many languages as possible (O’Horan et al., 2016; Ponti et al., 2019). Proper language-specific automated syntactic analyses are still unavailable for 99% of the world’s languages due to the lack of respective treebanks. Therefore, delexicalized transfer of dependency parsers has profiled as the most viable cross-lingual transfer option (Zeman and Resnik, 2008; McDonald et al., 2011; Søgaard, 2011). Delexicalized transfer is conceptually the least demanding option in terms of language-specific resource requirements. The only provision, in order to transfer the parser trained on a delexicalized treebank of a resource-rich language, is a POS tagger in a low-resource target language based on the Universal POS (UPOS) tagset (Petrov et al., 2012). Delexicalized transfer is nowadays used primarily as a simple yet competitive baseline for more sophisticated transfer models. However, in realistic low-resource setups, one cannot guarantee additional resources such as parallel sentences (Ma and Xia, 2014; Rasooli and Collins, 2015, 2017), word alignments (Lacroix et al., 2016), or even sufficiently large monolingual corpus in the target language (Mulcaire et al., 2019). Thus, delexicalized transfer remains a widely useful and plausible option (Johannsen et al., 2016; Agić, 2017).

Delexicalized transfer comes in two main flavors. We either (1) choose the best parser from a set of available parsers, trained on treebanks of various resource-rich languages (single-best parser selection, SBPS) or (2) use the parser trained on a mixture of treebanks of (ideally related) resource-rich languages (multi-source parser transfer, MSP). Other transfer paradigms, like noise-based data augmentation (Şahin and Steedman, 2018), assume the existence of at least a small treebank for a target language, violating the assumption of a (treebank-wise) fully low-resource target language.

Both SBPS and MSP rely on some measure of structural alignment between languages in order to select either the single best source language parser (SBPS) or a set of (syntactically related) source languages (MSP). Existing solutions rely on measures like the Kullback-Leibler (KL) divergence be-
between source- and target-language distributions of POS trigrams (Rosa and Žabokrtský, 2015), which can be unreliable for small target language corpora or instance-level estimation. More recent approaches (Agić, 2017; Lin et al., 2019) choose suitable source languages based on manually coded typological similarities between languages available from databases such as WALS (Dryer and Haspelmath, 2013) or URIEL (Littell et al., 2017). The bottleneck of this approach is the manual effort and linguistic expertise needed to introduce a new language into the database.

**Proof-of-Concept and Contributions.** In this work, we propose a novel paradigm for cross-lingual parser transfer. The idea is to select the source-language parser for each target instance (i.e., POS-sequence), dubbed *instance-level parser selection* (ILPS). This is motivated by a simple observation that different source parsers provide most accurate parses for different target POS-sequences. We empirically show that an oracle ILPS leads to major potential gains compared to an oracle single-best parser selection at the treebank level (SBPS).

As a proof-of-concept for ILPS, we present a neural regression model that predicts the accuracy of a source-language parser for a given UPOS-“sentence” (i.e., a sequence of universal POS tags). We measure accuracies of parsers of resource-rich languages on UPOS-sentences from treebanks of other resource-rich languages to create training examples for the regression model. At inference time, we apply the trained regression model to select the best parser for each instance (i.e., each UPOS-sentence) of a low-resource target language.\(^1\)

We perform a large-scale evaluation of delexicalized dependency parser transfer, encompassing 42 source languages with large(r) treebanks, and 20 target (i.e., test) languages with small(er) treebanks from the Universal Dependencies (UD) v2.3 collection (Nivre et al., 2018). We show that, averaged across all test treebanks, our simple ILPS model significantly outperforms strong SBPS baselines (Rosa and Žabokrtský, 2015; Lin et al., 2019). We further demonstrate that we can easily aggregate instance-level predictions into an SBPS model, yielding improvements over the existing SBPS baselines for 16/20 and 17/20 test languages. Finally, we show that by ensembling the parses of few-best parsers according to the ILPS model’s predictions we can significantly outperform (1) the multi-source parser trained on the treebanks of all 42 source languages and (2) even surpass the performance of an oracle single-best treebank-level parser selection (i.e., oracle SBPS).

We believe that this proof-of-concept work uncovers the great potential of instance-based parser selection for cross-lingual parsing transfer for truly low-resource setups. The gaps with respect to the oracle (upper-bound) ILPS performance indicate that we have only scratched the surface of this potential. We hope that our work will inspire further investigations of this promising cross-lingual transfer paradigm in other setups and for other tasks.

**2 Motivation: The Case for Instance-Based Parser Selection**

The idea behind instance-level parser selection is intuitive: given a set of parsers for resource-rich source languages, it is unlikely that the same source-language parser is the best choice for all instances (i.e., UPOS-sentences) of the target language. Therefore, we first investigate the performance of an oracle model that would be able to predict the best source-language parser for each individual POS-sentence from the target-language treebank. To verify this, we rely on the well-known biaffine parser (Dozat and Manning, 2017; Dozat et al., 2017) and train it on delexicalized UD2.3 treebanks (Nivre et al., 2018) of 42 languages.\(^2\) We then parse the delexicalized treebanks of the 20 low-resource languages with all 42 source parsers, and measure their performance per each instance in each target treebank. We compare the performance of two oracle parser selection strategies: (1) single-best parser selection (SBPS), in which for each target test treebank we select the parser that performs best on the entire treebank; and (2) instance-level parser selection strategy (ILPS), where for each UPOS-sentence from each test treebank, we select the parser that produces the best parse for that UPOS-sentence.

The differences in Unlabeled Attachment Scores (UAS) between the two transfer paradigms are shown in Figure 1. This clearly demonstrates a

---

1 In contrast to existing methods which impose additional requirements on the target language (e.g., a sufficiently large target language corpus or an expert linguistic specification of the language’s syntactic properties), our ILPS setting conforms to a more realistic minimal-resource setup: it does not rely on any target-language resource other than a POS tagger.

2 We selected 42 languages with largest treebanks as the training languages. For languages with multiple treebanks (e.g., EN, CS), we finally chose the treebank for which the parser yielded the best monolingual parsing accuracy.
large gap in favor of ILPS: the average gain with ILPS is 14.5 UAS points, and it is prominent for all languages. It suggests that large improvements may be obtained with a model that can predict the best parser at the instance level, that is, for each UPOS-sentence separately. However, these are still oracle scores and we pose the following research question in this paper: (*Q1*) Is it possible to learn an instance-level prediction model to select the best parser given any UPOS-sentence, irrespective to its “language of origin”?\(^3\) In addition, even with noisy automatic instance-level predictions, one could still, by eliminating the noise through aggregation, use them to inform treebank-level source parser selection. In other words, another research question we pose is: (*Q2*) Can we improve single-best global parser selection through aggregating instance-level parser predictions?\(^3\)

3 Instance-Based Parser Selection

We now describe a novel ILPS framework based on a supervised regression model that predicts the parser accuracy for any UPOS-sentence. As such, it can be applied on UPOS-sequences of low-resource languages. As described in §2, we first train a (bi-affine) parser on delexicalized treebanks for each of the 42 resource-rich languages from UD2.3.\(^4\) We then parse with each parser the 41 treebanks of the other languages. This way we obtain the labels for training the ILPS regression model. The data preparation step is further detailed in §3.1, while the model is described in §3.2. At inference time, the ILPS model predicts the accuracy of each of the 42 parsers for each UPOS-sentence from delexicalized treebanks of the 20 test languages.\(^5\) Finally, in §3.3 and §3.4, we outline different strategies for merging the parse trees based on the predictions of the ILPS regression model. The full ILPS framework is illustrated in Figure 2.

3.1 Preparing ILPS Training Data

We first delexicalize all treebanks before training the parsers. After training a parser for each of the \(|L|\) training languages, we measure how each of them performs on treebanks of the other \(|L| - 1\) languages. Let \(\text{PARSER}_i\) denote the parser of the \(i\)-th training language and let \(\text{SENT}_j = \{\text{POS}\}_{n=1}^N\) be an UPOS-sentence of length \(N\) from the treebank of the \(j\)-th training language. Next, we must quantify how successful \(\text{PARSER}_i\) is on some UPOS-sentence \(\text{SENT}_j\). To this end, we use the number of correct dependency heads predicted by \(\text{PARSER}_i\) on \(\text{SENT}_j\). Using a raw number of correct heads as training labels for the ILPS regression model comes with one disadvantage: such a label would only indicate the suitability of the parser in isolation and not in comparison with other parsers. Therefore, we normalize the number of correct heads as

\[ y_{i,j} = \frac{\text{correct heads of } \text{PARSER}_i \text{ on } \text{SENT}_j}{\text{average of all parsers}} \]

\(\text{Note that in theory the } \text{oracle} \text{ gaps in favor of ILPS may be out of reach for automatic ILPS models, due to a potential parsing ambiguity introduced through delexicalization – i.e., the same UPOS-sentence (corresponding to different lexicalized sentences) may appear in the same treebank or across different treebanks with different gold parses. However, we have verified that this phenomenon is rare: ambiguous parses are present only for 1.4% UPOS-sentences in the concatenation of treebanks from 42 languages.}

\(\text{Note that this constitutes a minimal-resource transfer setup: our ILPS regression model does not rely on any information about the test languages nor their respective treebanks.}

\(^3\)
Figure 2: Illustration of the ILPS framework (at inference time) with three steps using an example sentence in Armenian (HY): (1) Probing – the ILPS regression model predicts the parsing accuracy on a given test UPOS-sentence for each of the 42 parsers; (2) Ranking – rank the parsers w.r.t. parsing accuracy for the instance and selects one or few best-performing parsers; (3) Reparsing – induce the final tree for the UPOS-sentence by merging trees produced by parsers selected in the previous step (only if more than one parser gets selected in step (2)).

PARSER$_i$ and SENT$_j$ is computed as follows:

$$y_{i,j} = \frac{\# \text{correct-heads}_{i,j}}{1/|L| \cdot \sum_{l=1}^{|L|} \# \text{correct-heads}_{l,j}}$$  (1)

The treebanks of training languages greatly vary in size. To account for the imbalanced treebank sizes, we up-sample all below-average treebanks and down-sample all above-average treebanks.

### 3.2 ILPS Regression Model

Our instance-level parser selection model is a regression model based on a Transformer architecture encoder (Vaswani et al., 2017) for UPOS-sentences. The encoding of the input UPOS-sentence is forwarded, together with the embedding vector representing the parser language, to a multi-layer perceptron. It predicts the score representing the prediction of the normalized number of correct heads that the parser is expected to yield.

**Parser and POS-tag embeddings.** We learn $|L|$ parser embeddings, $\{p_i\}_{i=1}^{|L|}$, one for each language (PARSER$_i$) and $K$ embedding vectors $\{t_k\}_{k=1}^K$, one for each UPOS-tag (Petrov et al., 2012). We initialize both parser and POS-tag embeddings randomly. POS-tag embeddings are then updated during the pretraining of the POS-sentence encoder.

**UPOS-sentence encoder.** We encode UPOS-sentences with the Transformer encoder. Let the UPOS-sentence SENT$_j = \{t^i_j, t^2_j, \ldots, t^T_j\}$ be a sequence of $T$ UPOS-tags. We encode each token $t^i_j$ ($i \in \{1, \ldots, K\}, j \in \{0, 1, \ldots, T\}$) with a vector $t^i_j$ which is the concatenation of the UPOS-tag embedding and a positional embedding for the position $j$.\(^6\) Let $\text{Transform}$ denote the encoder stack of the Transformer model (Vaswani et al., 2017) with $N_T$ layers, each coupling a multi-head attention net with a feed-forward net. We then apply $\text{Transform}$ to the UPOS-tag sequence and obtain contextualized UPOS-tag representations as follows:

$$\{tt^i_j\}_{j=1}^T = \text{Transform} (\{t^i_j\}_{j=1}^T);$$  (2)

Following Devlin et al. (2019), we pretrain the parameters of the $\text{Transform}$ encoder and the UPOS-tag embeddings via the masked language modeling objective on the concatenation of all training treebanks. As in the original work, we consider 15% of randomly selected tokens in each sentence (but no more than 20 tokens) for replacement. In 80% of the cases, we replace the UPOS-tag with the [MASK] token, in 10% of the cases we keep the original UPOS-tag, and in remaining 10% of the cases we replace it with a randomly chosen tag.

We fine-tune the pretrained $\text{Transform}$ encoder and the UPOS-tag embeddings on the main ILPS regression task. At this step, similar to Devlin et al. (2019), we prepend each UPOS-sentence with a special sentence start token $t^0_j = [ss]$, with the aim of using the transformed representation of that

---

\(^6\)We adopt the wavelength-based positional encoding from the original Transformer model (Vaswani et al., 2017).
We can directly use the vector of scores \( \mathbf{s} \) as the final fixed-size representation of the UPOS-sentence.

**Feed-forward regressor and loss function.** For a training instance \((\text{PARSER}_i, \text{SENT}_j, y_{i,j})\), we concatenate the parser’s embedding \( p_i \) and the UPOS-sentence encoding \( \mathbf{t}_{t} \), and feed it to a feed-forward regression network (i.e., a multi-layer perceptron, MLP), whose goal is to predict \( y_{i,j} \):

\[
\hat{y}_{i,j} = \text{MLP}(p_i; \mathbf{t}_{t})
\]  

(3)

We define the loss function to be a simple root mean square error (RMSE) over the examples in one mini-batch as follows:

\[
L = \sqrt{\frac{1}{NB} \sum_{i,j} (y_{i,j} - \hat{y}_{i,j})^2}
\]  

(4)

where \( NB \) is the number of instances in the batch.

### 3.3 Ranking and Ensembling

We can directly use the vector of scores \( \mathbf{y}_j = \{y_{i,j}\}_{i=1}^{L} \) to rank the \( L \) parsers according to their (predicted) parsing accuracy for the UPOS-sentence \( \text{SENT}_j \) from some test treebank.

**Pure ILPS.** This local parser ranking, based only on the predicted parser performance for the current UPOS-sentence \( \text{SENT}_j \), is used to select one or a few best parsers for that UPOS-sentence. If we select only a single best parser and only according to the instance-level predictions, we refer to the pure instance-level parser selection (ILPS) setup:

\[
i_{\text{ILPS}}(j) = \arg \max_i \{\hat{y}_{i,j} | i \in \{1, 2, \ldots, |L|\}\}
\]  

(5)

**SBPS from ILPS predictions.** ILPS predictions can be easily aggregated to produce a treebank-level estimate of the source parsers’ performance for a text language. This brings the ILPS paradigm back into the single-best parser selection (SBPS) realm, hopefully with SBPS estimates originating from our ILPS predictions being more robust than competing SBPS metrics (Rosa and Žabokrtský, 2015; Lin et al., 2019). For a treebank of an unseen test language consisting of \( M \) POS-sentences, we get the global parser’s performance estimates \( \hat{y}_i \) simply by averaging ILPS predictions for that parser, \( \hat{y}_{i,j} \), over all \( M \) test POS-sentences:

\[
\hat{y}_i = \frac{1}{M} \sum_{j=1}^{M} \hat{y}_{i,j}
\]  

(6)

The best treebank-level parser is then selected as the one with the highest aggregate score \( \hat{y}_i \):

\[
i_{\text{SBPS}_{\text{ILPS}}} = \arg \max_i \{\hat{y}_i | i \in \{1, 2, \ldots, |L|\}\}
\]  

(7)

**Ensembling.** It is often the case – both at the instance level and at the treebank level – that two or more parsers yield similar performance. In such cases, one would expect to benefit from aggregating the predictions made by those parsers. We refer to the settings in which we consider more than one parser as ensembling (Ens) settings. Note that ensembling is equally applicable to both the pure ILPS setup as well as to the previously outlined SBPS\(_{\text{ILPS}}\) setup in which we aggregate instance-level predictions to select the best “treebank-level” parser. In both cases, we must determine a threshold \( \tau \in [0, 1] \) that defines the set of “good enough” parsers, in relative terms w.r.t. the performance of the best parser. The sets of parsers whose trees are to be merged are obtained as follows:

\[
\{i_{\text{ILPS}}^{\tau}(j)| i \} = \{i | \forall i : \hat{y}_{i,j} \geq \max(\hat{y}_{i,j}) \cdot \tau\},
\]  

(8)

\[
\{i_{\text{SBPS}_{\text{ILPS}}}^{\tau}| i \} = \{i | \forall i : \hat{y}_i \geq \max(\hat{y}_i) \cdot \tau\}.
\]  

(9)

where Eq (8) refers to the pure ILPS setting, and Eq. (9) refers to the SBPS\(_{\text{ILPS}}\) setting.

### 3.4 Reparsing

After selecting multiple parsers in the ensemble settings, we need to merge their produced parse trees into a final tree. It is commonly referred to as reparsing (Sagae and Lavie, 2006). We resort to a standard reparsing procedure in which we: (1) merge the trees produced by individual parsers into a weighted graph \( G \) – the parser \( i \) contributes to an edge with the weight \( w_i = \hat{y}_{i,j} \) (for pure ILPS; for SBPS\(_{\text{ILPS}}\), \( w_i = \bar{y}_i \)) if the parser \( i \) predicted that edge, and with \( w_i = 0 \) otherwise; (2) induce the Maximum Spanning Tree (MST) of \( G \) (Edmonds, 1967) as the final parse of the input UPOS-sentence, see again Figure 2.

### 4 Experimental Setup

**Data.** We perform all experiments on the UD v2.3 dataset,\footnote{https://universaldependencies.org/} as it contains a wide array of both...
resource-rich languages with large treebanks – split into train, development, and test portions – and low-resource languages with small test treebanks. For our experiments, we select 42 languages with the largest treebanks as our resource-rich source languages for training, and a set of 20 typologically diverse low-resource languages for testing. Following established practice (Wang and Eisner, 2018b), at inference we use gold UPOS-tags of test treebanks for all models in comparison.

**ILPS Hyperparameters** are optimized via fixed-split cross-validation on our training set (see §3.1). We set the embedding size for both parser embeddings and UPOS-tag embeddings, as well as the hidden size of the feed-forward Transformer layers to 256. The transformer encoder has \( N_T = 3 \) layers with 8 attention heads in each layer. We update the model in mini-batches of 16 examples, using Adam (Kingma and Ba, 2015) with the default parameters: \( \beta_1 = 0.9, \beta_2 = 0.999, \) and \( \epsilon = 10^{-8} \), with an initial learning rate set to \( 10^{-4} \). The regression MLP has 2 hidden layers with 256 units each, plus a linear projection layer that compresses the 256-dimensional vector into a single prediction score. We perform early stopping based on the loss on the development set. For all ensembles, we set the parser inclusion threshold \( \tau \) to 0.9.

**Oracle scores and baselines.** In order to provide more context for the reported ILPS scores, we also report the results of two oracle methods described in §2: the oracle single-best parser selection (OR-SBPS), and the oracle instance-level best parser selection (OR-ILPS). We compare to three competitive baselines: (1) the standard multi-source parser (MSP) baseline which trains a single parser model on the concatenation of all training treebanks;\(^{11}\) and two competitive SBPS baselines, (2) KL-SBPS – treebank-level parser selection based on the Kullback-Leibler divergence between the UPOS-tag trigram distributions of the source and target language treebanks (Rosa and Žabokrtský, 2015) and (3) L2V-SBPS – treebank-level parser selection based on the cosine similarity between the syntax-based vectors of the source and target language from WALS (Lin et al., 2019).

**Ensembles.** We evaluate two ensembles based on the predictions of our ILPS-based regression model, described in §3.3: (1) an instance-level ensemble in which we merge the trees of the best parsers for each sentence (ENS-ILPS) and (2) ENS-SBPS\(_{ILPS}\) – an ensemble merging the trees of treebank-level best parsers, where the treebank-level estimates are aggregated from the instance-level predictions). We evaluate comparable ensembles (i.e., with the same parser inclusion performance threshold \( \tau = 0.9 \)) for both SBPS baselines: ENS-KL-SBPS and ENS-L2V-SBPS.

## 5 Results and Discussion

We first show the results for single-best parser selection models. We then proceed to a more realistic ensemble setup in which the models are allowed to select more than just one parser.
Single-parser selection. We report results (UAS) for all single-parser selection methods (i.e., no ensembles) along with the oracle scores on all 20 test treebanks. Table 1 provides performance per language, and Figure 3 shows the summary of the results. Our pure instance-based parser selection model (ILPS) significantly\(^1\) outperforms both SBPS baselines (KL-SBPS and L2V-SBPS) averaged across all languages (see Figure 3). Individual instance-level predictions made by ILPS, however, do seem to be rather noisy. This is supported by the observation that SBPS\(_\text{ILPS}\) significantly outperforms ILPS. Since SBPS\(_\text{ILPS}\) is a simple treebank-level aggregation of ILPS sentence-level predictions, the gain can only be explained as the product of noise elimination through aggregation. ILPS outperforms KL-SBPS and L2V-SBPS on 13/20 and 14/20 test languages, respectively, whereas SBPS\(_\text{ILPS}\) improves on 17/20 and 16/20 languages over the respective baselines. This first set of results, the preliminary comparison with well-established and competitive baselines for delexicalized parser transfer, seems encouraging and validates the viability of the instance-based parser selection paradigm.

ILPS and SBPS\(_\text{ILPS}\) still do not match the performance of the multi-source parser (MSP) in this simple single-parser-selection setup. We find this somewhat expected: ILPS and SBPS\(_\text{ILPS}\) are based on parsers trained on single treebanks, whereas MSP is trained on the concatenation of all training treebanks. Therefore, we include MSP as a baseline in our ensemble evaluation as well.

Ensemble evaluation results. We show the results for the ensemble models in Table 2. A summary of results for this setup is provided in Figure 4. Allowing for the selection of more than a single parser in cases in which our ILPS-based predictions warrant so (i.e., when two or more parsers yield similarly good performance for some low-resource language) allows SBPS\(_\text{ILPS}\) (i.e., its ensemble version, Ens-SBPS\(_\text{ILPS}\)) to significantly outperform the strong MSP baseline. The two SBPS baseline methods in their ensemble variants (Ens-KL-SBPS and Ens-L2V-SBPS) reduce the gap in comparison with the previous single-parser selection setup (see Table 1 again). However, our treebank-level parser selection model based on instance-level predictions (Ens-SBPS\(_\text{ILPS}\)) still significantly outperforms the ensembles of the other two SBPS methods.

As an encouraging finding, both Ens-ILPS and Ens-SBPS\(_\text{ILPS}\) outperform the oracle baseline Ens-Or-All, which merges parses produced by all training parsers, using their gold performance on the test treebanks for weighting the individual parser contributions. Furthermore, Ens-SBPS\(_\text{ILPS}\) also improves over the oracle single-parser selection OR-SBPS reported in Table 1. In summary, we believe these results provide sufficient evidence for the viability of the ILPS transfer paradigm and warrant further research efforts in this direction.

6 Related Work

Parsing languages with no training data has been a very active topic of research for nearly a decade since the pivotal works by McDonald et al. (2011) and Petrov et al. (2012). Many diverse approaches are explored along the lines of model transfer, annotation projection, machine translation (Täckström et al., 2013; Guo et al., 2015; Zhang and Barzilay, 2015; Tiedemann and Agić, 2016; Rasooli and Collins, 2017), and selective sharing based on language typology (Naseem et al., 2012) and structural similarity (Ponti et al., 2018). However, vast majority of prior work involves bulk evaluation, whereby transfer parsers are validated by mean accuracy on test data. Such evaluation protocols stand in contrast with the fact that languages exhibit high variance in syntactic structure, which calls for a sensitive treatment of every sentence. While an oracle single-source parser may be appropriate for the majority of sentences in a given dataset, instance-based treatment closes the gap to the best achievable result given an array of pretrained parsers, as we also show in §2.

Early efforts in this line of research include data point selection where language models are used to capture the prevalent syntactic structure of a language and score potential training instances such that a multi-source parser is trained on the mixture of training instances that are most similar to the test language instances (Søgaard, 2011). Instead of instance selection one can also apply instance reweighting in accordance to their similarity to the test language (Søgaard and Wulff, 2012). Regardless of whether we attempt to align languages on an instance-level or on a treebank-level there is a need for a similarity measure between languages. Prior work relied on existing manually curated resources such as the URIEL database (Littell et al., 2017),

\(^{12}\)Significance tested with the Student’s two-tailed t-test at \(p = 0.01\) for sets of sentence-level UAS scores.
using the KL-Divergence on POS-trigrams (Rosa and Žabokrtský, 2015), or handcrafted features derived from the datasets at hand.

Our work is most similar to the recent work of Lin et al. (2019): they learn to score and rank languages in order to predict the top transfer languages. However, contrary to their work, our approach does not employ a model to learn the ranking, but transforms the labels to directly reflect the ranking when we train the scoring model. In addition, we stress the importance of instance-based learning for cross-lingual parser transfer in particular. Another core difference is that our approach is an end-to-end system without external resources or handcrafted static features. Instead, our framework relies on trainable parser embeddings that encode the necessary features in a single representation.

From another viewpoint, the work of Wang and Eisner (2016, 2018a,b) explores the potential of synthesizeing and reordering delexicalized POS sequences to come up with better parser transfer without unrealistic assumptions on target-language resources. Their work in synthetic delexicalization is compatible with ours as it lends itself entirely to instance-based parsing. Finally, the line of work by Ammar et al. (2016) in learning monolithic models over multiple training languages, and its continuation for zero-shot learning by Kondratyuk and Straka (2019) also promises to abstract away from language boundaries, but still records significantly lower zero-shot scores than our proposal.

### 7 Conclusion and Future Work

In this work, we indicated that there is a large disparity between mean test-set and per-instance accuracy in cross-lingual parser transfer setups. We showed convincing evidence that one source parser is not the optimal choice for all target-language sentences. Motivated by the analysis, we proposed a novel approach to close this gap: instance-based parser selection. Our framework provides competitive results, where in the ensemble setting we
outperform all baselines, and markedly even the single-source oracle parser selection, while using a simple thresholding heuristic to select the parsers.

We see the proposed model as the first exploratory step in the direction of robust instance-level parser transfer, which opens several avenues for future research. While this proof-of-concept work assumed the existence of gold POS tags, we will also experiment with the same approach “in the wild”, with learned or transferred POS taggers, and we will also extend the study to lexicalized parser transfer. Future work may also include learning to rank parsers instead of applying simple heuristics. Further improvements may be obtained by using the accuracy predictions from our model as a feature and combining it with external linguistic features (Ponti et al., 2019). The idea of instance-based parser selection lends itself also to domain adaptation settings, following Plank and Van Noord (2011), even for well-resourced languages.

Acknowledgments

IV is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909).

References

Željko Agić. 2017. Cross-lingual parser selection for low-resource languages. In Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies, pages 1–10.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pages 4171–4186.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Proceedings of ICLR.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In Proceedings of ACL, pages 3125–3135.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In Proceedings of NAACL-HLT, pages 4171–4186.
Helen O’Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2016. Survey on the use of typological information in natural language processing. In Proceedings of COLING, pages 1297–1308.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In Proceedings of LREC, pages 2089–2096.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. In Proceedings of ACL, pages 1566–1576.

Edoardo Maria Ponti, Helen O’Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, Thierry Poibeau, Ekaterina Shutova, and Anna Korhonen. 2019. Modeling language variation and universals: A survey on typological linguistics for natural language processing. Computational Linguistics, 45(3):559–601.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In Proceedings of EMNLP, pages 328–338.

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

Rudolf Rosa and Zdeněk Žabokrtský. 2015. KLcpos3 - a language similarity measure for delexicalized parser transfer. In Proceedings of ACL, pages 1531–1542.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsoring. In Proceedings of NAACL-HLT, pages 129–132.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In Proceedings of EMNLP, pages 5004–5009.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In Proceedings of ACL, pages 682–686.

Anders Søgaard and Julie Wulff. 2012. An empirical study of non-lexical extensions to delexicalized transfer. In Proceedings of COLING, pages 1181–1190.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In Proceedings of NAACL-HLT, pages 1061–1071.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of NeurIPS, pages 5998–6008.

Dingquan Wang and Jason Eisner. 2016. The galactic dependencies treebanks: Getting more data by synthesizing new languages. Transactions of the Association for Computational Linguistics, 4:491–505.

Dingquan Wang and Jason Eisner. 2018a. Surface statistics of an unknown language indicate how to parse it. Transactions of the Association for Computational Linguistics, 6:667–685.

Dingquan Wang and Jason Eisner. 2018b. Synthetic data made to order: The case of parsing. In Proceedings of EMNLP, pages 1325–1337.

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.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In Proceedings of EMNLP, pages 1857–1867.
| Train Language | #sentences | #tokens   |
|----------------|------------|-----------|
| Estonian (et)  | 24384      | 341122    |
| Korean (ko)    | 23010      | 296446    |
| Latin (la)     | 16809      | 293306    |
| Norwegian (no) | 15696      | 243887    |
| Finnish (fi)   | 14981      | 127602    |
| French (fr)    | 14450      | 354699    |
| Spanish (es)   | 14305      | 444617    |
| German (de)    | 13814      | 263804    |
| Polish (pl)    | 13774      | 104750    |
| Hindi (hi)     | 13304      | 281057    |
| Catalan (ca)   | 13123      | 417587    |
| Italian (it)   | 13121      | 276019    |
| English (en)   | 12543      | 204585    |
| Dutch (nl)     | 12269      | 186046    |
| Czech (cs)     | 10160      | 133637    |
| Portuguese (pt)| 9664       | 255755    |
| Bulgarian (bg) | 8907       | 124336    |
| Slovak (sk)    | 8483       | 80575     |
| Romanian (ro)  | 8043       | 185113    |
| Latvian (lv)   | 7163       | 113405    |
| Japanese (ja)  | 7133       | 160419    |
| Croatian (hr)  | 6983       | 150455    |
| Slovenian (sl) | 6478       | 112530    |
| Arabic (ar)    | 6075       | 223881    |
| Basque (eu)    | 5396       | 72974     |
| Ukrainian (uk) | 5290       | 88043     |
| Hebrew (he)    | 5241       | 137721    |
| Persian (fa)   | 4798       | 121064    |
| Indonesian (id)| 4477       | 97531     |
| Danish (da)    | 4383       | 80378     |
| Swedish (sv)   | 4303       | 66645     |
| Urdu (ur)      | 4043       | 108690    |
| Chinese (zh)   | 3997       | 98608     |
| Russian (ru)   | 3850       | 75964     |
| Turkish (tr)   | 3685       | 37918     |
| Serbian (sr)   | 2935       | 65764     |
| Galician (gl)  | 2272       | 79327     |
| Greek (el)     | 1662       | 42326     |
| Uyghur (ug)    | 1656       | 19262     |
| Vietnamese (vi)| 1400       | 20285     |
| Afrikaans (af)| 1315       | 33894     |
| Hungarian (hu) | 910        | 20166     |
| Average (avg)  | 8483       | 158233    |

| Test Language  | #sentences | #tokens   |
|----------------|------------|-----------|
| Erzya (myv)    | 1550       | 15790     |
| Faroese (fo)   | 1208       | 10002     |
| Amharic (am)   | 1074       | 10010     |
| Kazakh (kk)    | 1047       | 10007     |
| Bambara (bm)   | 1026       | 13823     |
| Thai (th)      | 1000       | 22322     |
| Buryat (bfr)   | 908        | 10032     |
| Breton (br)    | 888        | 10054     |
| North Sami (sme)| 865     | 10010     |
| Cantonese (yue)| 650       | 6264      |
| Upper Sorbian (hsb)| 623 | 10736     |
| Maltese (mt)   | 518        | 11073     |
| Armenian (hy)  | 470        | 11438     |
| Irish (ga)     | 454        | 10138     |
| Coptic (cop)   | 267        | 6541      |
| Telugu (te)    | 146        | 721       |
| Tamil (ta)     | 120        | 1989      |
| Yoruba (yo)    | 100        | 2666      |
| Belarusian (be)| 68         | 1382      |
| Lithuanian (lt)| 55         | 1060      |
| Average        | 1094       | 8802      |

Table 3: 42 source languages on which we trained monolingual parsers. Number of sentences (#sentences) and total token count (#tokens).

Table 4: 20 unseen test languages. Number of sentences (#sentences) and total token count (#tokens).