Cross-lingual parser selection for low-resource languages

Željko Agić
Department of Computer Science
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark
zeag@itu.dk

Abstract

In multilingual dependency parsing, transferring delexicalized models provides unmatched language coverage and competitive scores, with minimal requirements. Still, selecting the single best parser for any target language poses a challenge. Here, we propose a lean method for parser selection. It offers top performance, and it does so without disadvantaging the truly low-resource languages. We consistently select appropriate source parsers for our target languages in a realistic cross-lingual parsing experiment.

1 Introduction

Treebanks are available for only ∼1% of the languages spoken in the world today, the resource-rich *sources*. One major goal of cross-lingual transfer learning is to provide robust NLP for all the *targets*, or the remaining ∼99%.

If we want to parse *any* language for syntactic dependencies, the only principled method that currently enables it is delexicalized model transfer. By relying on uniform POS tags only, it offers unprecedented language coverage. First introduced by Zeman and Resnik (2008), and consolidated by the seminal works of McDonald et al. (2011; 2013) and Søgaard (2011), delexicalized parsing is nowadays considered to be a simple baseline.

Recent work promises cross-lingual methods that score almost as high as supervised parsers. Unfortunately, it also introduces requirements that a vast majority of languages cannot meet. The systems proposed by, e.g., Ma and Xia (2014) or Rasooli and Collins (2015) require:

- very large parallel corpora, often in excess of 2M parallel sentences for each language pair, coupled with near-perfect tokenization and sentence splitting;
- high-quality sentence and word alignments for all the language pairs, provided by aligners that favor closely related languages;
- accurate POS tagging using fully supervised taggers that score ∼95% on held-out data.

Latest work by Johannsen et al. (2016), among a few others, shows that in a real-world scenario, where no such unrealistic assumptions are made, delexicalized transfer still constitutes a very competitive choice for multilingual parsing.

Here, we assert that even simple delexicalized parsing might be in need of a reality check.

Realistic delexicalized parsing? The idea behind delexicalization is very simple: we omit all lexical features from the parsers, both at training and at runtime, so that they operate on POS sequences only. All that is then needed to parse an unknown language is a tagger using a uniform POS representation such as the “universal” POS tagset by Petrov et al. (2011).

Delexicalized parsing itself comes in two distinct basic variants:

i) **multi-source**, where we train a single parser by joining multiple delexicalized source-language treebanks, and

ii) **single-source**, where each source-language treebank contributes a single parser, and then we select the one to use from this pool of parsing models.

Most often, we pick the **single-best** source parser for a given target language. The rankings of the candidate source parsers are determined by evaluation on target language test data.

Single-best parsers generally perform better than multi-source parsers. For example, in their experiment, Agić et al. (2016) show that the single-source variant beats multi-source delexicalization in 23/27 languages and scores +3 points higher in UAS on average. For fairness, their parsers all work with cross-lingual POS taggers.

Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017), pages 1–10,
Gothenburg, Sweden, 22 May 2017.

However, we argue that single-best source parsing is **not realistic**. Only in an evaluation framework do we possess prior knowledge of i) which target language we are parsing, and ii) what the source rankings are for the targets. Single-best parsing thus amounts to an oracle. By contrast, in the real world, we expect to parse by i) **predicting** the target language name from the text input at runtime, and by ii) **selecting** the most appropriate source parser for that language from the parser pool. If the prediction or selection turn out incorrect, we are likely to end up producing a suboptimal parse. Furthermore, while parsing accuracy is measured on test sets of $\sim$1000 sentences on average, the real input can take a much wider size range. This variation in size may challenge the validity of any design choices made on test set-level only.

The cross-lingual parsing community has largely ignored this problem, focusing instead on test set-based evaluation by proxy. This, in addition to a list of methodological biases, has spawned a number of complex models incapable of scaling down to real low-resource languages.

Our contributions. How do we single out the best source parser if the language of the input text has to be predicted at runtime?

To answer this question, our paper makes the following contributions:

i) We propose a set of methods for matching texts to source parsers. Our methods are simple, as they rely on nothing but character-based language identification and typological similarity. They consistently find the best parsers for the target languages.

ii) We set aside the test-set granularity assumption. Instead, we assume that the parser input can vary in size from as little as one sentence. Our methods prove to be remarkably adaptable to this size variation.

iii) By combining our approaches, our best system even manages to exceed the performance of single-best oracle source parsers.

In our submission, we strive to introduce only the minimal requirements, and to maintain a realistic setup. For example, in all the experiments, we apply cross-lingual POS taggers for truly low-resource languages. By controlling for POS sources, we show how an ingrained bias towards direct supervision of taggers may render any parsing results irrelevant in a low-resource context. Our code and data are freely available.[1]

2 Method

Say we had to find a suitable source parser for the following sentence, written in an unknown target language:

Knjiga	*ima*	*12*	*svezaka*	.
NOUN	VERB	NUM	NOUN	PUNCT

Intuitively, and following the language relatedness hypothesis of McDonald et al. (2013), among the source languages, we would single out the one typologically closest to the target sentence, and apply its delexicalized parser. Further, we build our approach on this intuition.

More formally, let $S \in \mathcal{S}$ be a source language treebank, and $T \in \mathcal{T}$ a POS-tagged target text to parse. Further, let $\mathrm{dist} : \mathcal{S} \times \mathcal{T} \rightarrow [0, +\infty)$ be a cross-lingual distance measure.[2] In this framework, finding the single-best parser amounts to minimizing the distance over all sources:

$$\hat{S}_{min} = \arg\min_{S \in \mathcal{S}} \mathrm{dist}(S, T)$$

2.1 Distance measures

In estimating distance, or similarity as its inverse, we consider two basic sources of available information for sources and targets: i) the raw texts and ii) the POS tag sequences.

We proceed to define three distance measures over these information sources. The first measure is based on sequences of POS tags. The other two model character sequences and typological information, and they are novel to our work.

KL-POS. This is the POS trigram-based distance metric of Rosa and Žabokrtský (2015a). Essentially, it expresses the Kullback–Leibler (KL) divergence between distributions of source and target trigrams of POS tags:

$$\mathrm{dist}_{\mathrm{K}}(S, T) = \sum_{t_i \in T} f_T(t_i) \log \frac{f_T(t_i)}{f_S(t_i)}$$

The relative frequencies f_S and f_T of trigrams t_i in source and target data are estimated on the re-

[1] `https://bitbucket.org/zeljko_agic/freasy`

[2] Rather than *metric*, we use the term *measure*, as not all conditions for metrics are satisfied by all proposed measures. Namely, KL divergence is not symmetric.

spective POS sequences:

$$f(t_i) = \frac{\text{count}(t_i)}{\sum\limits_{\forall t_j} \text{count}(t_j)}$$

We inherit the properties of the original KL-POS proposal, but we introduce one minor change: while i) special tag values are used to encode sentence beginnings and endings, and ii) the source counts for unseen trigrams are smoothed for the distance to be well-defined, we use linear interpolation smoothing following Brants (2000) rather than set these counts to 1 in the Rosa and Žabokrtský (2015a) implementation.

In plain words, this measure compares the relative frequencies of target POS trigrams $t_i \in T$ to the frequencies of these trigrams in all the sources $S \in \mathcal{S}$, and then we select the one associated with the lowest KL divergence. For our example sentence, KL-POS predicts Finnish to be the best source parser. The sentence is, however, in Croatian, for which the Finnish parser ranks as 19/26 in our experiment. In contrast, if we feed KL-POS five sentences at a time, it selects Slovene (1/26).

We expect KL-POS to be sensitive to both the sample size and the POS tagging quality. The latter is of particular importance for low-resource dependency parsing. Incidentally, the POS tags in our Croatian example are all correct. For these reasons, we propose the following two measures. The first one (LANG-ID) is based on character, i.e., byte n-grams, while the other one (WALS) augments the n-grams approach by leveraging typological data.

LANG-ID. The approach is very straightforward: We use Lui and Baldwin's (2012) `langid.py` module to identify the best source language for the given input. They employ a naive Bayes classifier with a multinomial event model, and feed it a mixture of byte n-grams ($1 \leq n \leq 4$).

More specifically, `langid.py` has predefined models for ∼100 languages, but we constrain it to predict into the set of source languages $\mathcal{S}$ only. We also use its probability re-normalization feature. Our distance measure then amounts to:[3]

$$\text{dist}_\text{L}(S, T) = 1 - p_S, (S, p_S) \in \texttt{langid.rank}(T)$$

As `langid.py` estimates p_S, the probability of input T belonging to a source language S, we convert it to a distance $(1 - p_S)$.

For our Croatian sample sentence, LANG-ID predicts Slovene with a confidence of 0.99, and it converges already for the first token.

On the downside, limiting `langid.py` predictions to sources S only might negatively impact parsing. The classifier commits early on to one answer, assigning it a high confidence, and for languages with fewer related source languages in the model, source selection might be significantly off. For example, take this Hungarian sentence:

Ettől	*a*	*győzelemtől*	*magabiztos*	*lettem*	.
ADV	VERB	NOUN	NOUN	NOUN	PUNCT

While KL-POS selects Estonian (1/26), our source-constrained LANG-ID predicts Spanish (19/26) as the best source. However, if we allow LANG-ID to predict beyond the list of source languages only, it guesses Hungarian with $p = 1$.

Since Hungarian poses as a target language, we cannot use this correct guess to select a model directly, but we can exploit it downstream. Our next distance measure does so by leveraging typology data on top of LANG-ID.

WALS. Our typology-based approach relies on a simple premise: If we can guess the language of the target text, we can employ a language database to match the input with a similar source language. This language database should encode various linguistic properties across many languages in a principled way. One such resource is WALS (Dryer and Haspelmath, 2013). Currently it contains structured data for 2,679 languages.[4] Each language is described through 202 features: they include various structural properties in several categories, most notably in phonology, morphology, and syntax.

Now we describe the WALS-reliant distance measure. For any target language T, we predict the language name using LANG-ID. For this prediction, we retrieve the corresponding feature vector $\mathbf{v_T}$ from WALS, provided that WALS contains some information on T. Our distance measure then amounts to comparing the target WALS vector $\mathbf{v_T}$ to source WALS vectors $\mathbf{v_S}, \forall S$:

$$\text{dist}_\text{W}(S, T) = d_h(\mathbf{v_S}, \mathbf{v_T}),$$

where d_h is the Hamming distance between WALS source and target vectors $\mathbf{v_S}$ and $\mathbf{v_T}$.

[3] Note that `langid.rank(T)` returns pairs of source languages and respective probabilities $(S, p_S), \forall S \in \mathcal{S}$. We apply `langid.py` with the options `-d -l -n`, see `https://github.com/saffsd/langid.py`.

[4] `http://wals.info/download`

For each S and T, we only compare the subsets of features for which both $\mathbf{v_S}$ and $\mathbf{v_T}$ are nonempty. If there would be no WALS entries for T, we would fall back to LANG-ID. In our experiments, however, all the languages are already represented in WALS.

For our Croatian example input, WALS predicts Slovene as source (1/26), while it chooses Finnish (6/26) for the Hungarian sentence.

2.2 Combining the measures

Both LANG-ID and WALS suffer a same constraint: in contrast to KL-POS, they do not abstract away from the alphabet. This may cause issues for languages with distinct alphabets. On the other hand, KL-POS needs more data for estimation, and might deteriorate with POS tagging accuracy.

Since the strengths and drawbacks of the three approaches appear to be complementary, here we propose their linear combination.

Normalization. The distances that our measures output are not directly comparable, even if their source language rankings are. We normalize the distances into probability distributions by applying a softmax function:

$$\hat{P}(S|T) = \mathrm{softmax}(\mathrm{dist}^{-1}(S,T), \tau) = \frac{\exp \frac{\mathrm{dist}^{-1}(S,T)}{\tau}}{\sum_{X \in \mathcal{S}} \exp \frac{\mathrm{dist}^{-1}(X,T)}{\tau}}$$

Note that we invert the distances (dist^{-1}) as a small distance between S and T translates into a high probability of S lending its parser to T. We use the softmax temperature τ for controlling the contributions of the sources. For very large τ, $\tau \to +\infty$, the probabilities for the individual sources all even out at $p \to 1/|\mathcal{S}|$, while $\tau \to 0^+$ isolates the most probable source at $p \to 1$.

The change from dist to $\hat{P}(S|T)$ changes our objective from minimizing the distance between sources and targets to maximizing the probability of S lending a parser to T:

$$\hat{S}_{max} = \arg\max_{S \in \mathcal{S}} \hat{P}(S|T)$$

COMBINED. With the probability normalization in place, we now introduce the linear combination of the three approaches:

$$\hat{P}(S|T) = \sum_i \lambda_i \hat{P}_i(S|T), \text{with} \sum_i \lambda_i = 1$$

The values λ_i can be tuned empirically on development data, with i indexing our three distance measures. That way, we can control the amounts of contributions for the individual methods, similar to tuning the contributions of the individual sources through softmax temperature τ.

Algorithm 1: Source selection and reparsing.

```
Data: Target language sample $T$, source
      language treebanks $S$, and parsers $h_S$
Result: Predicted single-best parses $G^t_{max}$,
        reparsed trees $\mathrm{DMST}(G^t), \forall t \in T$
Create the sources distribution.
$\hat{P}(S|T) \leftarrow \mathrm{softmax}(\mathrm{dist}^{-1}(S,T), \tau), \forall S$
Find the best source.
$\hat{S}_{max} \leftarrow \arg\max_S \hat{P}(S|T)$
for each sentence $t$ in $T$ do
    Get all parses, build the graph.
    $G^t = (V, E), E = \{(u_S, v) \in h_S(t), \forall S\}$
    Get the single-best parse.
    $G^t_{max} = (V, E_{max})$,
        where $E_{max} = \{(u_{\hat{S}_{max}}, v)\}$
end
return $G^t_{max}, \mathrm{DMST}(G^t), \forall t \in T$
```

With the COMBINED approach, we aim specifically at providing "the best of both worlds" in source discovery: an improved robustness to orthographies on one side, and an added stability to varying input sample sizes on the other.

3 Experiments

In our setup, we parse the target texts T with multiple source parsers h_S, and we seek to predict the best source parses for all the targets. We now expose the details of this experiment outline.

Data. We use the Universal Dependencies (UD) treebanks (Nivre et al., 2016) version 1.3.[5] UD currently offers 54 dependency treebanks for 41 different languages.

Since our experiment requires realistic cross-lingual POS taggers, we use the freely available collection of training sets by Agić et al. (2016).[6] It is built through low-resource annotation projection over parallel texts from The Watchtower online library (WTC).[7] Thus, we intersect the lan-

[5] hdl.handle.net/11234/1-1699

[6] https://bitbucket.org/lowlands/release/

[7] http://wol.jw.org/

guages with POS tagging support from WTC with the UD treebanks for a total of 26 languages whose training and testing sets that we proceed to use in the experiment. We make use of the English UD development data in hyper-parameter tuning.

Tools. For POS tagging, we use a state-of-the-art CRF-based tagger MarMoT[8] (Müller et al., 2013). We use Bohnet's (2010) `mate-tools` graph-based dependency parser. Both tools are run with their default settings.

In our experiments, we control for the sources of POS tags. We distinguish i) direct in-language supervision, where the taggers are trained on target language UD training data, from ii) cross-lingually predicted POS, where we train the taggers on WTC-projected annotations.

For training the delexicalized source parsers, we use the following standard features, in reference to the CoNLL 2009 file format:[9] ID, POS, HEAD, and DEPREL (Hajič et al., 2009). In specific, we don't leverage the UD morphological features (FEATS) as not all languages support them in the 1.3 release. We subsample the treebanks for parser training with a ceiling of 10k sentences, so as to avoid the bias towards the largest treebanks such as Czech with 68k training set sentences.

Baselines and upper bounds. We set the oracle **SINGLE-BEST** source parsing results as the main reference point for our evaluation. We compare all systems to these scores, as our benchmarking goals are to i) reach SINGLE-BEST performance through best source prediction and to ii) surpass it by weighted reparsing.

We compare our approach to the standard multi-source delexicalized parser of McDonald et al. (2011) (*multi-dir* in their paper, **MULTI** here). In training, we uniformly sample from the contributing sources up to 10k sentences.

Reparsing. We collect all single-source parses of target sentences $t \in T$ into a dependency graph. The graph $G^t = (V, E)$ has target tokens as vertices V. The edges $(u_S, v) \in E$ originate in the delexicalized source parsers $h_S, \forall S$.

Following Sagae and Lavie (2006), we can apply directed maximum spanning tree decoding $\text{DMST}(G^t)$, resulting in a voted dependency parse for a target sentence t, where each source contributes a unit vote. Such unit voting presumes that all edges have a weight of 1. We refer to this approach as **UNIFORM** reparsing. We also experiment with weighing the edges in G^t through the distance measures KL-POS, WALS, and COMBINED:

$$\text{weight}(u_S, v) = \hat{P}(S|T), \forall u_S \in G^t, \forall t \in T$$

The weights in turn depend on the granularity, as varying sizes of T influence the similarity estimates coming from KL-POS and WALS.

Parameters. We tune the softmax temperature to $\tau = 0.2$ for both KL-POS and WALS by using the English UD development data. For simplicity, we fix $\lambda_K = \lambda_W = 0.5, \lambda_L = 0$ without tuning, i.e., in the COMBINED system we give equal weight to KL-POS and WALS. We exclude LANG-ID from reparsing as it is subsumed by WALS.

Our experiment assumes the variability of input size in sentences. We use the full UD test sets for all 26 languages. However, we vary the sample size or **granularity** g in best source prediction. It is implemented as a moving window over the test sets, with sizes of 1 to 100.

The experiment workflow is condensed in Algorithm 1. It shows how we arrive at best source predictions and reparsed trees for a target sample T. In the algorithm sketch, we assume $g = |T|$, i.e., the granularity is implied by the sample size, but further we provide results for varying g. Any edge weighting in reparsing is made internal to DMST.

4 Results

First, we provide a summary of our experiment results in Table 1. We then proceed to break down the scores by language in Table 2.

Summary. We discern that our COMBINED approach yields the best overall scores in the realistic scenario, both in source selection and in reparsing. The latter score remarkably even surpasses the informed upper bound SINGLE-BEST system by 0.36 points UAS. It reaches the highest UAS over cross-lingual POS in both selection and reparsing, while KL-POS closely beats it in reparsing over fully supervised POS.

We form a general ordering of the four approaches following these summary results: COMBINED>KL-POS$\geq$WALS>LANG-ID.

[8] `https://github.com/muelletm/cistern/blob/wiki/marmot.md`

[9] `https://ufal.mff.cuni.cz/conll2009-st/task-description.html`

POS source:	*Direct supervision* 95.33±2.42	*Cross-lingual* 71.65±5.65	*g*
Delexicalized			
MULTI	62.04±4.67	49.48±5.37	–
SINGLE-BEST	65.53±2.96	51.96±4.35	–
Source selection			
KL-POS	63.68±3.30	49.84±5.25	100
LANG-ID	60.12±3.83	48.14±5.25	5
WALS	60.37±4.36	48.87±5.45	3
COMBINED	64.18±3.04	50.20±5.20	50
Reparsing			
KL-POS	66.55±3.70	51.55±5.46	4
UNIFORM	64.10±4.68	50.92±5.47	–
WALS	65.17±4.56	51.54±5.52	2
COMBINED	66.50±3.56	52.32±5.30	50

Table 1: Summary UAS parsing scores for all 26 languages, over two underlying sources of POS tags. Gray: highest scores grouped by POS and method. ±: 95% confidence intervals. g: sample size (granularity) associated with the best score.

Looking into the optimal target sample sizes g, the COMBINED system peaks at 50 sentences. KL-POS works best with samples of 100 sentences in source prediction, and only 4 sentences in reparsing. In contrast, WALS needs only 2-3 for both, while LANG-ID peaks at 5 sentences.

Split by languages. The results in Table 2 are provided as differences in UAS to our reference point: the oracle SINGLE-BEST system. In source selection, we aim to match the oracle scores, while we seek to surpass them through reparsing.

The top-performing **source prediction** system is the COMBINED one: it comes closest to the oracle score for 10/26 languages. The other three approaches manage the same feat for 5-7 languages, while the MULTI-source delexicalized parsers still pose a challenge for 8/26 languages.[10]

Notably, KL-POS even beats the SINGLE-BEST oracle by 0.4 UAS for one language (Danish), a score that is made possible by changes in source selection for different portions of the test set due to sample granularity. KL-POS and LANG-ID reach the oracle score for 3 languages, WALS for 5, and COMBINED for 9 languages. In **reparsing**, the COMBINED system once again produces the absolute best scores, here for 15/26 languages. MULTI parsers are unable to match the reparsing systems, as both KL-POS and WALS also come very close to the upper bound on average. Viewed separately, these two reparsing systems are evenly split with 13/26 languages for each, and reach almost identical average scores, with KL-POS ahead WALS by only 0.01 point UAS.

5 Discussion

We reflect on the results of our experiment from the viewpoints of i) POS tagging impact, and ii) input size or granularity.

Sources of POS tags. Throughout the paper, we emphasized the importance of using cross-lingual POS tagging in dependency parsing work that features truly low-resource languages.

We conducted triple runs of all our experiments, by changing the underlying POS tags from cross-lingual to i) tags obtained through direct in-language supervision via the UD training data and ii) gold POS tags. The respective average tagging accuracies over the 26 test languages thus changed from 71.65% to 95.33% and 100%. As the observations were virtually unchanged between fully supervised tagging and gold tagging, we reported the former together with cross-lingual tagging.

Table 1 adds insight into the influence of tagging quality. In **source selection**, KL-POS outperforms LANG-ID and WALS by ∼2.5 points UAS over monolingual POS, but this advantage drops to less than 1 point with cross-lingual POS.

There is an even more notable turnabout following the underlying POS source change in **reparsing**. With direct supervision, KL-POS beats UNIFORM and WALS by 2.35 and 1.38 points UAS, and even surpasses the COMBINED system by 0.05 points. However, when working with cross-lingually induced POS tags, the COMBINED approach beats the three other systems by 0.77–1.40 points UAS, and KL-POS and WALS even out.

We expected KL-POS to show less resilience to changes in POS tagging quality compared to the other methods. The significant change in the observations highlights the need for more careful treatment of low-resource languages in contributions to cross-lingual parsing.

Granularity. Input size in sentences, or granularity g as a model of input size variation, is an important feature in our experiments. In Table 1 and 2, we only reported the scores with optimal granularities for each method. Here, we add insight by observing the link between g and UAS in

[10] Note that in some cases more than one system records the same score for a language.

	POS	*Delexicalized*			*Source selection*			*Reparsing*			COMBINED	
		MULTI	SINGLE-BEST		LANG-ID	KL-POS	WALS	UNIFORM	KL-POS	WALS	*selection*	*reparsing*
Arabic (ar)	51.48	-2.55	37.02	id	-8.29	-2.51	**-0.24**	-0.25	0.26	**0.40**	-0.90	0.57
Bulgarian (bg)	69.99	-0.42	49.94	cs	-1.79	**-0.45**	-10.00	1.05	**1.50**	1.22	-1.45	1.32
Czech (cs)	78.24	0.38	50.13	sl	-1.14	**-0.17**	-1.79	2.24	2.64	**2.72**	-0.05	2.79
Danish (da)	84.89	0.13	58.74	no	0.00	**0.40**	-0.19	1.29	1.20	**2.00**	0.20	1.87
German (de)	67.54	-1.32	44.64	no	**0.18**	-0.63	-0.11	0.37	0.52	**0.90**	-0.13	1.16
Greek (el)	62.44	2.04	54.98	it	-3.82	**0.00**	-0.82	3.70	**4.00**	3.81	-0.11	3.93
English (en)	79.75	-0.55	56.34	no	-6.17	-1.98	**-1.27**	0.47	0.70	**0.79**	-2.18	0.94
Spanish (es)	86.60	-1.76	69.29	it	-1.34	-1.01	**0.00**	0.36	**1.43**	1.16	0.00	1.27
Estonian (et)	76.11	-7.54	52.34	fi	-0.45	-0.60	**0.00**	-5.75	**-3.88**	-4.68	0.00	0.09
Persian (fa)	28.04	-1.23	25.33	ar	**0.00**	-2.67	-8.88	-0.88	**-0.25**	-0.82	-6.61	-0.13
Finnish (fi)	68.23	-6.03	45.01	et	**0.00**	-0.22	**0.00**	-4.59	**-3.14**	-3.47	0.00	1.00
French (fr)	78.80	-0.64	54.37	es	-2.25	-0.59	**-0.51**	0.02	**0.75**	0.63	-0.51	1.10
Hebrew (he)	62.64	-0.04	44.35	ro	-9.36	**0.00**	-5.12	1.91	1.82	**2.02**	0.00	2.03
Hindi (hi)	51.62	-20.74	37.07	ta	**-0.16**	-21.73	-21.52	-20.20	-20.35	**-19.99**	-19.54	-20.33
Croatian (hr)	75.95	-0.02	49.89	sl	**0.00**	-0.27	**0.00**	2.59	2.52	**2.79**	0.00	2.79
Hungarian (hu)	68.38	-8.08	46.07	et	-10.86	-8.08	**-3.92**	-5.86	-6.02	**-5.48**	0.00	-5.83
Indonesian (id)	77.78	-1.95	56.47	ro	-21.38	**-0.01**	-6.28	1.61	**2.13**	1.87	0.00	2.09
Italian (it)	87.69	-0.21	67.60	es	-0.48	**-0.10**	-0.15	0.69	**2.16**	1.88	-0.36	2.32
Dutch (nl)	71.49	1.08	54.15	es	-1.45	**-0.34**	-0.91	2.70	2.96	**3.29**	-0.91	3.10
Norwegian (no)	86.31	-0.29	63.99	sv	-3.10	**-1.69**	-3.09	0.57	0.86	**1.19**	-1.70	1.29
Polish (pl)	79.07	0.39	62.95	hr	-11.84	-1.70	**-1.41**	1.85	**2.81**	2.59	-1.70	3.15
Portuguese (pt)	85.98	-0.88	67.50	it	-0.38	-0.43	**0.00**	0.10	**1.23**	0.77	0.00	1.09
Romanian (ro)	75.77	-0.10	53.25	es	-4.71	-5.14	**-0.42**	1.96	2.45	**2.50**	-0.24	2.62
Slovene (sl)	76.53	-2.66	53.72	cs	-3.98	**-0.64**	-3.97	-1.67	**-0.81**	-1.07	-4.11	-0.84
Swedish (sv)	88.19	-3.15	66.01	no	**-0.04**	-0.52	-3.50	-2.05	-1.36	**0.30**	-1.54	1.18
Tamil (ta)	43.49	-8.49	29.86	hu	-6.23	**-3.97**	-6.23	-9.45	**-6.73**	-8.39	-3.97	-1.30
Mean	71.65	-2.48	51.96		-3.81	-2.12	-3.09	-1.05	-0.41	-0.42	-1.76	0.36
Best sample size g	–	–	–	–	5	100	3	–	4	2	50	50
Best single #	–	8	–	–	6	5	7	–	–	–	10	–
Absolute best #	–	0	–	–	2	1	0	0	5	3	1	15

Table 2: Parsing target languages using source language weighting. We report changes in UAS over the SINGLE-BEST delexicalized parsers. POS tags are provided by cross-lingual taggers. **Bold**: the best system for a given language, separate for source selection and reparsing, excluding COMBINED. Underlined: COMBINED systems that match or beat the other respective weighting methods. Gray: Best overall average score.

source selection and reparsing, for all the weighting approaches.

Figure 1A shows the changes in UAS for best **source prediction** with varying input sizes. LANG-ID and WALS converge on their predictions early on, so their UAS scores remain nearly constant. Yet, KL-POS largely benefits from more POS data: it starts at around -1.5 UAS from WALS and even below LANG-ID, but steadily rises up to $\sim$1 point over WALS at its peak UAS for $g = 100$ sentences.

The B part of Figure 1 reveals a different pattern for KL-POS in **reparsing**. While WALS once again stays expectedly constant, KL-POS peaks with +0.01 UAS at $g = 4$, only to decrease with growing input sizes. Since WALS rarely updates its initial predictions, its source distributions $\hat{P}(S|T)$ mostly remain unchanged with g, implying the same invariance for the reparsing scores. However, KL-POS converges much later, which means that its $\hat{P}(S|T)$ decreases in variation as g increases: all but the best source start contributing less weight to the edges in G^t for reparsing. Moreover, while the differences between WALS vectors do not update with g, the KL divergences update towards the predicted best source, and away from the other contributing sources.

The COMBINED method manages to integrate the advantages of KL-POS and WALS. In source selection, it improves its predictions with larger samples, while maintaining the robustness over very small samples (+0.5 UAS over WALS, +1.9 over KL-POS for $g = 1, 2$). In reparsing, the combination significantly outperforms the two systems it integrates. Even more notably, where KL-POS deteriorates and WALS flatlines, the COMBINED reparsing scores steadily improve with g. We suggest that integrating i) the invariance of WALS language vector distances with ii) the variation of KL-POS towards the predicted best source with increasing granularity causes this positive effect.

Source rankings are implicit in the distributions

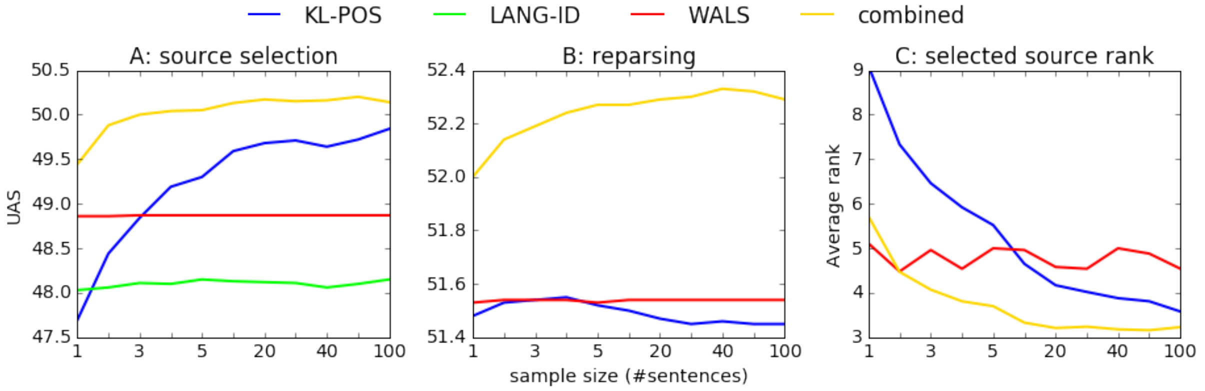

Figure 1: Sample size (granularity) impact on source selection and reparsing. A, B: Changes in UAS over different sample sizes for the four approaches in best single source prediction, and three approaches in reparsing. C: Average true rank of the predicted best source in relation to granularity.

$\hat{P}(S|T)$. Here, we compare them to the gold rankings induced from the SINGLE-BEST scores. In Figure 1C, we observe how the **average true rank** of the selected source changes with granularity. KL-POS significantly improves with larger samples. Working with one-sentence inputs, it assigns the targets to only the 9th best source on average, while with 50-100 sentences, it assigns the 3rd or 4th best source parser. WALS is mostly constant at an average rank of 4.5. COMBINED once again provides the best of both worlds, as it assigns the 5th or 6th best source with $g = 1, 2$, and stably predicts the 3rd best source on average with inputs of +20 sentences.

6 Limitations

Contributions of sources. In our experiments, we used SINGLE-BEST parsers as upper bounds. There, the best source parser was selected for each target language by its overall performance on the respective test set. This system recorded an average UAS of 51.96±4.35. However, if we select the best single source for each *sentence* instead, the oracle score rises significantly: by +13.58 points UAS, to 65.54±4.88.

To substantiate, in Figure 2, we show that for 28.66% of the parsed sentences on average, the best parse does **not** come from the parser that was ranked best in test set-level evaluation. We view this 13-point gap in UAS as a margin for improving our source selection in future work, as it suggests that we have yet to exhaust the search space of predictive features for sentence-level source ranking. For example, we could use the UD development data to learn models that predict the rankings of source parsers from the target sequences of tokens and POS tags, possibly using WALS as an additional feature source.

Scalability. Our contribution is mainly focused on the link between delexicalized parsing and language identification. In that focus, we abstracted away from certain relevant low-level issues in realistic text processing.

Firstly, we used gold-standard tokenization and sentence splits. While accurate splitters exist for many languages, realistic segmentation would still incur a penalty. Secondly, the LANG-ID models we used are readily available for around 100 languages. Scaling up to +1000 languages would require scaling down on the available resources for building identifiers, which would likely result in a minor performance decrease downstream.

Finally, and most importantly, our models depend on the existing cross-lingual POS taggers by Agić et al. (2016), which in turn rely on parallel resources. While their models do scale up, we excluded POS tagging from language identification. A more realistic proposal would assume that taggers, too, have to be selected at runtime before any parsing takes place.

7 Related work

Research in cross-lingual POS tagging and dependency parsing is nowadays plentiful, but only a fraction of it focuses on *truly* low-resource languages and *realistic* proposals.

McDonald et al. (2011) were among the first notable exceptions to use real cross-lingual POS taggers in their multi-source parser transfer experiments. They employed the label propagation-based taggers from Das and Petrov (2011). Agić et al. (2015; 2016) used a simpler approach to projection, but they were the first to propose multilingual projection for building taggers and parsers for 100+ low-resource languages in one pass. Zeman and Resnik (2008) used perplexity per word as a metric to select the source training instances that relate to the target data. Søgaard (2011) extended their approach to sequences of POS tags, and to multiple sources. Their metrics in turn relate to LANG-ID and KL-POS, but their approach is based on test-set granularity and the selection of appropriate data for *training* the parsers, while ours deals with varying input sizes and source parser selection at runtime.

Ammar et al. (2016) noted a -6.3 points decrease in UAS for cross-lingual parsing accuracy when the language identifiers and POS tags are predicted at runtime. Their taggers are fully supervised with 93.3% average accuracy for the seven resource-rich languages from their experiment. They also simulated a low-resource scenario, where they used gold POS and omitted language guessing.

WALS data has been heavily exploited in NLP research. In that line of work, and partly related to our paper, Søgaard and Wulff (2012) proposed adapting delexicalized parsers through distance-based instance weighting over WALS data. Their work in turn relates to Naseem et al. (2012), who also use WALS features in a multilingual parser adaptation model. The research by Naseem et al. (2012) and Täckström et al. (2013) addresses the issues with multi-source delexicalized transfer by selectively sharing model parameters, also with typological motivation through WALS features. This line of work has seen subsequent improvements by Zhang and Barzilay (2015), who introduce a hierarchical tensor-based model for constraining the learned representations based on desired feature interactions. Georgi et al. (2010) and Rama and Kolachina (2012) used WALS to evaluate the concept of language similarity for facilitating cross-lingual NLP. Östling (2015) used WALS to evaluate word order typologies induced through word alignments. O'Horan et al. (2016) provide a comprehensive survey on the usage of typological information in NLP.

Plank and Van Noord (2011) applied similarity measures over cross-domain data for dependency parser adaptation. Prior to our contribution, only Rosa and Žabokrtský (2015a; 2015b) attempted to address source parser selection, by using KL divergence over gold POS tags.

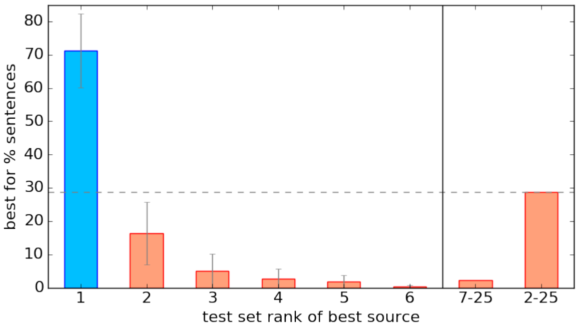

Figure 2: Distribution of per-sentence top-scoring source parsers over their test-set ranks. Blue: Percentage of sentences for which the best parser was ranked #1 in test set-based evaluation. Red: Sentences where the best parser was ranked #2-25, i.e., not ranked #1. The percentages are averaged over 26 languages.

8 Conclusions

We introduced an unbiased approach for cross-lingual transfer of delexicalized parsers. It is a robust and scalable source parser selection and reparsing system for low-resource languages. In a realistic experiment over cross-lingual POS tags and varying quantities of input text, our method remarkably outperformed even the informed upper bound delexicalized system. We emphasize the importance of acknowledging specifics of actual low-resource languages through realistic experiment design when proposing solutions aimed at addressing these languages.

Acknowledgements

We are thankful to Héctor Martínez Alonso, Barbara Plank, and Natalie Schluter for their valuable comments on an earlier version of the paper. We also thank the anonymous reviewers for their feedback. Finally, we acknowledge the NVIDIA Corporation for supporting our research.

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *ACL*, pages 268–272.

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. 2016. Multilingual Projection for Parsing Truly Low-Resource Languages. *TACL*, 4:301–312.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many Languages, One Parser. *TACL*, pages 431–444.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In *COLING*, pages 89–97.

Thorsten Brants. 2000. TnT: A Statistical Part-of-Speech Tagger. In *ANLP*, pages 224–231.

Dipanjan Das and Slav Petrov. 2011. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *ACL*, pages 600–609.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Ryan Georgi, Fei Xia, and William Lewis. 2010. Comparing Language Similarity Across Genetic and Typologically-Based Groupings. In *COLING*, pages 385–393.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *CoNLL*, pages 1–18.

Anders Johannsen, Željko Agić, and Anders Søgaard. 2016. Joint Part-of-Speech and Dependency Projection from Multiple Sources. In *ACL*, pages 561–566.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-Shelf Language Identification Tool. In *ACL*, pages 25–30.

Xuezhe Ma and Fei Xia. 2014. Unsupervised Dependency Parsing with Transferring Distribution via Parallel Guidance and Entropy Regularization. In *ACL*, pages 1337–1348.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *EMNLP*, pages 62–72.

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 *ACL*, pages 92–97.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient Higher-Order CRFs for Morphological Tagging. In *EMNLP*, pages 322–332.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective Sharing for Multilingual Dependency Parsing. In *ACL*, pages 629–637.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *LREC 2016*, pages 1659–1666.

Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2016. Survey on the Use of Typological Information in Natural Language Processing. *arXiv preprint arXiv:1610.03349*.

Robert Östling. 2015. Word Order Typology Through Multilingual Word Alignment. In *ACL*, pages 205–211.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A Universal Part-of-Speech Tagset. *arXiv preprint arXiv:1104.2086*.

Barbara Plank and Gertjan Van Noord. 2011. Effective Measures of Domain Similarity for Parsing. In *ACL*, pages 1566–1576.

Taraka Rama and Prasanth Kolachina. 2012. How Good are Typological Distances for Determining Genealogical Relationships among Languages? In *COLING*, pages 975–984.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-Driven Cross-Lingual Transfer of Dependency Parsers. In *EMNLP*, pages 328–338.

Rudolf Rosa and Zdeněk Žabokrtský. 2015a. KLcpos3 - a Language Similarity Measure for Delexicalized Parser Transfer. In *ACL*, pages 243–249.

Rudolf Rosa and Zdeněk Žabokrtský. 2015b. MST-Parser Model Interpolation for Multi-source Delexicalized Transfer. In *IWPT*, pages 71–75.

Kenji Sagae and Alon Lavie. 2006. Parser Combination by Reparsing. In *NAACL*, pages 129–132.

Anders Søgaard and Julie Wulff. 2012. An Empirical Study of Non-Lexical Extensions to Delexicalized Transfer. In *COLING*, pages 1181–1190.

Anders Søgaard. 2011. Data Point Selection for Cross-Language Adaptation of Dependency Parsers. In *ACL*, pages 682–686.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target Language Adaptation of Discriminative Transfer Parsers. In *NAACL*, pages 1061–1071.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation between Related Languages. In *IJCNLP*, pages 35–42.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical Low-Rank Tensors for Multilingual Transfer Parsing. In *EMNLP*, pages 1857–1867.