Improving Low-resource RRG Parsing with Cross-lingual Self-training

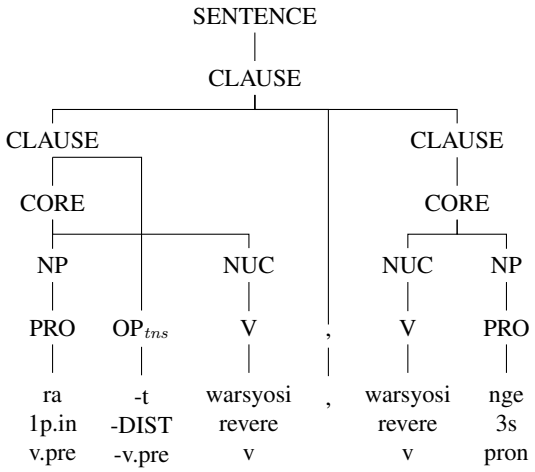
Kilian Evang Laura Kallmeyer Jakub Waszczuk
Kilu von Prince Tatiana Bladier Simon Petitjean
Heinrich Heine University Düsseldorf, Germany
first.last@hhu.de

Abstract

This paper considers the task of parsing low-resource languages in a scenario where parallel English data and also a limited seed of annotated sentences in the target language are available, as for example in bootstrapping parallel treebanks. We focus on constituency parsing using Role and Reference Grammar (RRG), a theory that has so far been understudied in computational linguistics but that is widely used in typological research, i.e., in particular in the context of low-resource languages. Starting from an existing RRG parser, we propose two strategies for low-resource parsing: first, we extend the parsing model into a cross-lingual parser, exploiting the parallel data in the high-resource language and unsupervised word alignments by providing internal states of the source-language parser to the target-language parser. Second, we adopt self-training, thereby iteratively expanding the training data, starting from the seed, by including the most confident new parses in each round. Both in simulated scenarios and with a real low-resource language (Daakaka), we find substantial and complementary improvements from both self-training and cross-lingual parsing. Moreover, we also experimented with using gloss embeddings in addition to token embeddings in the target language, and this also improves results. Finally, starting from what we have for Daakaka, we also consider parsing a related language (Dalkalaen) where glosses and English translations are available but no annotated trees at all, i.e., a no-resource scenario wrt. syntactic annotations. We start with cross-lingual parser trained on Daakaka with glosses and use self-training to adapt it to Dalkalaen. The results are surprisingly good.[1]

1 Introduction

Treebanks play an increasingly important role in typological research, where linguists are oftentimes faced with (primarily) oral low-resource languages and where grammatical frameworks such as Role and Reference Grammar (RRG; Van Valin and Foley, 1980; Van Valin, 2005) are a common choice (Toratani and González Vergara, 2020). RRG is a non-transformational linguistic theory strongly inspired by typological concerns. Its development was guided by the question of what a linguistic theory would "look like if it were based on the analysis of languages with diverse structures such as Lakhota, Tagalog and Dyirbal [...]?" (Van Valin, 2005, p. 1). RRG assumes constituency structures to be organized in layers, *viz.* nucleus (containing the predicate), core (containing the nucleus and the arguments of the predicate) and clause (the core and extracted arguments). Furthermore, each layer can have modifiers (termed periphery elements) and operators. An example from Daakaka (an Oceanic language, von Prince, 2015) is given in Fig. 1, using the annotation scheme from Bladier et al.



"We revered, revered him."

Figure 1: RRG annotation of a Daakaka sentence with glosses, parts of speech, and translation. Glosses: 1p.in– first person plural inclusive; DIST–distal TAM marker (for past and counterfactual contexts); 3s: third person singular

[1] Our experimental code is available at https://gitlab.com/treegrasp/rrgproj2.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4360–4371
October 12–17, 2022.

(2022).

With respect to complex constructions, i.e., combinations of more than one CLAUSE, CORE or NUC, RRG distinguishes not just coordination and subordination but, in addition, cosubordination. The latter has the general form [[ ]$_X$ [ ]$_X$]$_X$, an example is the combination of the two CLAUSEs in Figure 1 into a larger CLAUSE. In such a construction, an operator that applies to categories X takes scope over both X daughters while being realized only once. The TAM marker in Figure 1 for instance assigns tense to both CLAUSEs. The various ways of combining two NUC, CORE or CLAUSE constituents differ with respect to whether one of the two depends on the other, how tight the two units are concerning time and location of the two events, and to what extent the two units share operators such as tense, aspect, modality etc., all of which can be explained by the respective combination of layer level (NUC, CORE or CLAUSE) and construction type (coordination, subordination or cosubordination).

In this paper, we show a way to train RRG parsers on only a little amount of annotated data in the target language or, in the case of Dalkalaen, even no annotated data, yielding parse trees that can presumably reduce annotation effort considerably when used as a starting point for treebanking. Furthermore, the resulting parse trees might be sufficiently good to be the basis for (semi-)automatic investigations of syntactic properties of low-resource languages.

The scenario underlying our work is that we have a limited amount of data (translated to English and possibly glossed), and that a small subset of it (a few hundred or at most a few thousand sentences) is annotated with RRG trees. This is a realistic scenario for the result from typological fieldwork and grammar description. Consequently, and in contrast to most other proposals for parsing low-resource languages, (i) we are not aiming at dependency parsing but, instead, using a constituency scheme often used in typological research, and (ii) we cannot use a large language model trained on the target language, but (iii) we can make use of additional data typically included in fieldwork output, namely glosses and translations.

We propose to improve parsing in this scenario by cross-lingually injecting information from English translations (and glosses, if available) into the target-language parser, combined with self-training. The main research question we address in this paper is to what extent this method improves performance, and in what situations, i.e., at how many annotated trees a language stops being "low resource enough" to be helped by this method. To answer this, we test our methods on various simulated degrees of "low resource-ness", from 100 training trees to over 4 000. To get more robust data, we test our method not only on a real low-resource language (Daakaka), but also on four other languages for which parallel RRG treebanks are available (German, French, Russian, and Farsi). In all cases, we use English as the source language. Finally, we conduct experiments on a language related to Daakaka, namely Dalkalaen, where we have glosses and English translations but no syntactic annotations (except for test data). The aim is to test whether extending a cross-lingual parser (with English as source language) to a related language while keeping the source language leads to useful results. The hypothesis is that parsing a no-resource language (with glosses and English translations, i.e., no-resource concerning syntactic annotations) benefits from knowledge about the English translation and from knowledge learned from a related language.

In the remainder of the paper, we will first discuss related work (Sec. 2), then explain our parsing architecture, including grammar extraction, RRG parsing, cross-lingual transfer, and self-training (Sec. 3). Then, in Sec. 4, we will describe our experimental setup, and Sec. 5 will discuss the results of the experiments. We conclude in Sec. 6.

2 Related Work

The problem of parsing low-resource languages has been addressed both for dependency and for constituency parsing for different scenarios concerning available data on the target side and available high resource parallel data (Zeman and Resnik, 2008; Vania et al., 2019). Many approaches assume that there is enough unlabeled data for the target language to train a language model (i.e., the term 'low resource' refers only to syntactically annotated data). Schuster et al. (2019) for instance use monolingual language models for both source and target language and use a mapping between decontextualized variants of these vectors to guide the cross-lingual transfer. Mulcaire et al. (2019); Kitaev et al. (2019) use polyglot language models trained on source and target data as input to

crosslingual parsing. In contrast to this, we assume a scenario where there is not enough target data to train a language model.

There is also variation concerning the way the information from the source parse is injected into the target parsing process. Many approaches project the source parse onto the target sentence via alignments, sometimes even using multiple source languages (Agić et al., 2016). Instead of projecting a parse of a source sentence to a target sentence, McDonald et al. (2011) use parallel English data to select among the k best parses of a target language sentence by comparing to the parallel English parse. In our case, we use only English parallel source data, and we project supertag information along word alignments.

Some approaches use delexicalized parsers (McDonald et al., 2011; Das et al., 2017), but it has also been shown that lexicalization, in particular when covering aspects shared between source and target language helps (Falenska and Çetinoğlu, 2017). In this vein, we experiment with including glosses in the target language, which means that the target language tokens contain information that is similar to the information captured in the aligned English tokens.

Self-training as a means of training data augmentation for parsing has been proposed in a number of papers (McClosky et al., 2006; Reichart and Rappoport, 2007; Rehbein, 2011; Rotman and Reichart, 2019), though not in the context of the above-mentioned approaches to cross-lingual transfer in low resource parsing.

3 Method

RRG Parsing Following earlier work on RRG parsing (Bladier et al., 2020b), we adopt a formalization of RRG as a Tree Wrapping Grammar (TWG; Kallmeyer et al., 2013), a tree rewriting grammar formalism in the spirit of Tree-Adjoining Grammar (TAG; Joshi and Schabes, 1997). In their parsing architecture, training trees are first decomposed by a rule-based algorithm into TWG elementary trees (supertags) and bilexical dependencies.[2] A neural model is then trained to predict supertag and dependency head probability distributions for each word in a sentence. At test time, an A* parsing algorithm takes these distributions as input and computes the optimal TWG derivation and derived tree. Because RRG trees contain crossing branches, a decrossing step before supertag extraction and a recrossing step between parsing and evaluation on the gold data is required. Figure 2 shows an example.

Note that decrossing is rather local, since crossing branches usually occur within a single group of layers CLAUSE – CORE – NUC (resp. XP – CORE_X – NUC_X), as in Fig. 1 and 2. The decrossing algorithm of Bladier et al. (2020b), which we use, differs from the graph decrossing method proposed by Boyd (2007) in that the tree structures undergo only minimal changes to largely preserve the original tree structure and that it follows handwritten rules to decross the nodes uniformly, e.g. the discontinuous OP_{tns} node under the CLAUSE is always re-attached to the closest lower CORE node. The co-anchoring for PPs without a clear meaning contribution (e.g. *made for the stairs*) and particle verbs (e.g. *pick up*) is simulated by including the corresponding internal structure of the dependent supertag into the head supertag (see Fig. 2). The supertags are extracted with features that indicate the original parent node, which facilitates the rule-based recrossing step after parsing.

For the steps of decrossing, supertag extraction, A* parsing, and recrossing, we use the system developed by (Bladier et al., 2020a,b). Our cross-lingual extension is in the neural supertag and arc scoring module. The idea is illustrated in Figure 3: we train a monolingual system for the source language and then use its internal representations as an additional input to a system for the target language. The representations are fed through a cross-lingual attention mechanism to take into account word alignment information. The resulting cross-lingual system takes a source-language sentence and an aligned target-language sentence as input (both are available in the parallel treebanking scenario) and produces a parse for the latter. We now describe the monolingual scoring module and our cross-lingual extension in detail.

Monolingual Scoring Module The scoring module of Bladier et al. (2020b) takes a sequence of word embeddings $(x_i)_{i=1}^N$ as input, to which a 2-layer BiLSTM transducer is applied to provide contextualized word representations $(h_i)_{i=1}^N$. To these, two additional 2-layer BiLSTMs are applied to obtain supertag-specific ($h^{(sp)}$) and dependency-

[2]Note that the term 'dependencies' is used in a formal sense here, i.e., denoting directed edges between tokens. These edges mark combinations of the respective supertags via (wrapping) substitution or adjunction. They correspond only to a certain extent to dependencies in the linguistic sense.

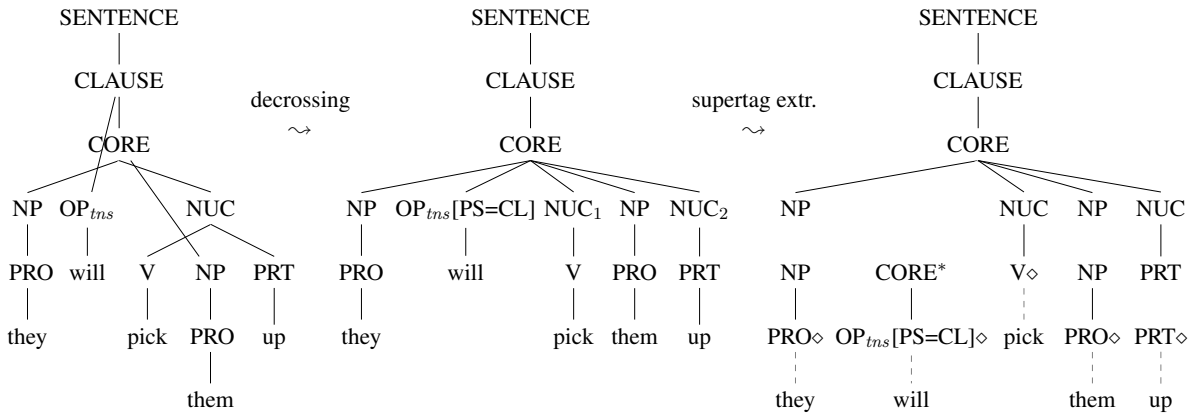


Figure 2: Example for decrossing and subsequent supertag extraction. [PS=CL] indicates that the OP$_{tns}$ node was originally immediately below CLAUSE. ◇ indicates the position of the lexical anchor.

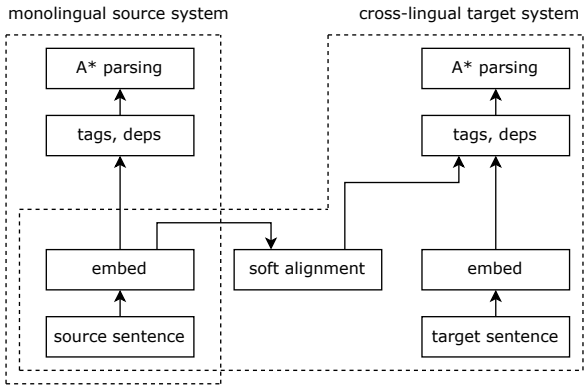


Figure 3: Architecture overview

specific ($h^{(dp)}$) word representations, respectively:

$$(h_1^{(sp)}, \ldots, h_N^{(sp)}) = \texttt{BiLSTM}_s(h_1, \ldots, h_N) \quad (1)$$

$$(h_1^{(dp)}, \ldots, h_N^{(dp)}) = \texttt{BiLSTM}_d(h_1, \ldots, h_N) \quad (2)$$

$(h_1^{(sp)}, \ldots, h_n^{(sp)})$ are used to predict supertags:

$$\Pr(sup(i)) = \texttt{softmax}(\texttt{Linear}_s(h_i^{(sp)})) \quad (3)$$

Finally, the dependency parsing component is based on biaffine scoring (Dozat and Manning, 2017), in which the head and dependent representations are obtained by applying two feed-forward networks to $(h_1^{(dp)}, \ldots, h_N^{(dp)})$, $\texttt{hd}_i = \texttt{FF}_{hd}(h_i^{(dp)})$ and $\texttt{dp}_i = \texttt{FF}_{dp}(h_i^{(dp)})$. The score of word j becoming the head of word i is then defined as follows (M is a matrix and b is a bias vector):

$$\phi(i,j) = \texttt{dp}_i^T M\ \texttt{hd}_j + b^T \texttt{hd}_j \quad (4)$$

Cross-lingual Embeddings via Soft Alignment Our cross-lingual system requires on input (i) a target sentence $(y_i)_{i=1}^M$ of length M, (ii) a corresponding source sentence $(x_i)_{i=1}^N$ of length N, and (iii) a soft word alignment function $a(j,i)$, obtained using standard unsupervised word alignment. $a(j,i)$ provides the probability of aligning the j-th target word with the i-th source word (0 in the source sentence is used for unaligned target words), and it holds that for each $j \in 1..M$ the sum of all $a(j,i)$ ($i \in 0..N$) is 1. The alignment enables an attention mechanism that projects the source supertag/dependency representations via the word alignments onto the corresponding target words. The source-side representations are provided by a monolingual parser trained on a large dataset available for the source language. Formally, let $(h_i^{(src\_sp)})_{i=1}^N$ and $(h_i^{(src\_dp)})_{i=1}^N$ be the source supertag and dependency representations (calculated as in Eq. 1 and Eq. 2). They are projected along the alignment function to obtain the target supertag/dependency projections:

$$h_j^{(prj\_sp)} = \sum\nolimits_i a(j,i) h_i^{(src\_sp)} \quad (5)$$

$$h_j^{(prj\_dp)} = \sum\nolimits_i a(j,i) h_i^{(src\_dp)} \quad (6)$$

These projections are then concatenated with the target language supertag and dependency representations ($h_j^{(trg\_sp)}$ and $h_j^{(trg\_dp)}$, resp.), calculated as in Eq. 1, 2 but without pre-trained embeddings:

$$h_j^{(hbr\_sp)} = [h_j^{(prj\_sp)}; h_j^{(trg\_sp)}] \quad (7)$$

$$h_j^{(hbr\_dp)} = [h_j^{(prj\_dp)}; h_j^{(trg\_dp)}] \quad (8)$$

where for two given vectors v and w, $[v;w]$ denotes their concatenation. The resulting "hybrid" representations, $h_j^{(hbr\_sp)}$ and $h_j^{(hbr\_dp)}$, are from

this point on used as in the monolingual model in order to determine the supertag distribution (Eq. 3) and dependency head scores (Eq. 4).

Gloss Embeddings Language documentation data seldom comes in parsed form, but often in glossed form. For example, in the Daakaka and Dalkalaen data we use (cf. Section 4), each word is annotated with parts of speech as well as morpheme-by-morpheme glosses, where content morphemes are represented by English translations, and function morphemes by acronyms indicating their function (see bottom of Figure 1 for an example). These glosses contain valuable information for parsing, as they put words into morphosyntactic classes and contain specific lexical information as well. We experiment with making this information available to the parser by concatenating the character-based word embeddings with character-based part-of-speech and morpheme-by-morpheme gloss embeddings.

Self-training Self-training is a technique used to improve learning on limited amounts of training data. Applied to parsing, the idea is to train on what little data is available first and use the resulting model to parse unannotated data. Some of the resulting parses are then selected and added to the training data. The expanded training data is used to train a new model. This process can be repeated for multiple "rounds" of self-training (McClosky et al., 2006; Reichart and Rappoport, 2007; Huang and Harper, 2009; Kurniawan et al., 2021). Selection of the added data is crucial; generally the idea is to add those instances where the parser is especially confident and that are thus likely to be correct.

4 Experimental Setup

Data Our choice of data is mainly driven by the availability of RRG-annotated resources. RRGbank (Bladier et al., 2018) contains a subset of the English Wall Street Journal Corpus (Marcus et al., 1993) annotated with RRG trees. RRGparbank (Evang et al., 2021; Bladier et al., 2022) contains George Orwell's novel *1984* in the original English as well as translations to French, German, Russian, and Farsi, along with sentence alignments between English and every other language. The English data as well as parts in other languages are annotated. In addition, we use 6 499 sentences in Daakaka, first published as von Prince (2013a) and described in von Prince (2015), which come with glosses, part-of-speech tags, and English translations, and 1 871 of which have been annotated with RRG trees following the RRGparbank annotation guidelines. Furthermore, we use 3 393 sentences in Dalkalaen, first published as von Prince (2013b), also including glosses and English translations, of which 102 sentences have been annotated with RRG trees.

We use all English trees in RRGbank and RRGparbank to train the English source model. For German, French, Russian, Farsi, Daakaka, and Dalkalaen, we focus on sentences with 25 tokens or less that are 1:1-aligned with an English sentence (for Daakaka and Dalkalaen, this is almost all sentences). Of these, we use 80% for training, 10% for development, and 10% for testing. Note that low resource language corpora are often oral corpora, and they therefore tend to contain shorter sentences. This is the reason why only very few of the Daakaka and Dalkalaen sentences are longer than 25, and this is also why a sentence length limit of 25 simulates this low resource scenario adequately. For Dalkalaen, we use the unannotated sentences for self-training, while the 101 annotated sentences are used for testing. In the case of Daakaka, we ran experiments with and without gloss embeddings (in addition to the token embeddings), while for Dalkalaen we always used gloss and token embeddings.

We randomly downsample the training data according to the degree of "low resource-ness" we are simulating in each experiment.

English Source Model The English source model uses FastText word embeddings (Bojanowski et al., 2017). We tuned its hyperparameters using 80% of the data for training and 10% for validation. We then trained it on the entire English dataset.

Word Alignments For all sentence pairs (a sentence in the target language and the aligned English sentence), we computed a soft word alignment matrix using `efmaral` (Östling, 2015) with default settings (code modified to output matrices). For the experiments using gloss embeddings, we align to the glosses instead of the target-language words.

Target Models After a phase of hyperparameter tuning on the development data (cf. Appendix A), we trained 1) monolingual models and 2) cross-lingual models on the (decrossed and decomposed) training data for all target languages, except for Dalkalaen.

Source	Language	Total # Sentences	Total # Trees	≤ 25 tokens, 1:1-aligned # Sentences	≤ 25 tokens, 1:1-aligned ∅ Length	≤ 25 tokens, 1:1-aligned # Trees	≤ 25 tokens, 1:1-aligned # Supertags
RRGbank	English	49 208	3 765	n/a	n/a	n/a	n/a
RRGparbank	English	6 737	6 737	n/a	n/a	n/a	n/a
	German	6 661	5 822	5 026	13.0	4 482	3 128
	French	7 261	3 243	4 550	12.9	2 339	1 815
	Russian	6 669	6 001	5 638	11.2	5 315	3 293
	Farsi	6 604	1 253	4 726	12.3	1 040	946
von Prince (2013a)	Daakaka	6 499	1 871	6 279	10.3	1 845	1 170
von Prince (2013b)	Dalkalaen	3 393	102	3 272	10.2	101	229

Table 1: Data overview. The versions of RRGbank and RRGparbank used are the 2022-03-17 snapshots.

Self-training We experiment with up to 5 rounds of self-training. In each round, we use the current parsing model to parse all sentences that are not yet part of the training data. We then add the 500 output trees with the lowest weights as reported by the A* parser to the training data. With poorly performing initial models, it sometimes happens that many parse failures occur and less than 500 parses are found in total. In these cases, we add all parses.

For Daakaka and for the RRGparbank languages, self-training starts from a parser trained on a set of annotated trees, as described in the previous section. For Dalkalaen, the starting point was the parser trained on all annotated Daakaka data (train, dev, and test), with glosses.

Evaluation We applied the neural scoring model to the annotated part of the development/test data, then computed RRG trees using the A* parser. Failed parses were replaced by dummy trees consisting of a SENTENCE root that directly dominates POS tags taken from the highest-scoring supertags. The resulting trees were recrossed and compared to the manually annotated dev/test trees using the EVALB metric (Collins, 1997), ignoring function tags such as -PERI or -TNS. We report the F1 score.

5 Results and Discussion

Impact of Training Data Size We are interested in low-resource scenarios, so we first look at how the number of available training trees impacts parser accuracy for different languages. This provides the baseline against which we will later evaluate our cross-lingual embedding and self-training methods. We look at various degrees of "low resource-ness", simulated by randomly downsampling the training set: 100 training sentences, 500, 1000, 2000, 3000, 4000, and finally the scenario where we use as many training trees as possible (varies by language). The blue dots in Fig. 4 show the results. In all cases, we see a considerable improvement in each step up to 2000 training sentences, and after that (for those languages where we have data), results improve only slightly. Note that the f-scores in general are rather low (the best ones around 80%) compared to state of the art constituency parsing. This is due to the fact that, since we are simulating a low resource scenario, we do not use any pretrained word embeddings, even for the languages where these would be available.

Impact of Cross-lingual Embeddings Fig. 4 (orange crosses) also shows the effect of including cross-lingual embeddings in the model. In general, the less training data we have in the target language, the more the cross-lingual information from the aligned English sentences is helpful. Furthermore, in none of the cases, the f-score of the cross-lingual model is below the one of the monolingual model, except for one slight outlier for German at training data size 500. (Note however that on the test data, cross-lingual transfer, with 500 or 1 000 training sentences, is helpful for German, see Table 2.) The only language where cross-lingual embeddings improve parsing performance only very little, even when using only 100 sentences of training data, is Daakaka when using glosses. We think that the reason for this is that the helpful information coming from English words, in particular the implicit information on English supertags, is already to a certain extent provided by the glosses, therefore the cross-lingual model does not contribute a lot of useful additional features.

It is striking that monolingual parsing for German with only little training data (500 or more sentences) is better than for instance for French, even though German has more syntactic variation and therefore more supertags. We suspect that the reason is that French has more ambiguities in high frequency lexical items, in particular ambiguities

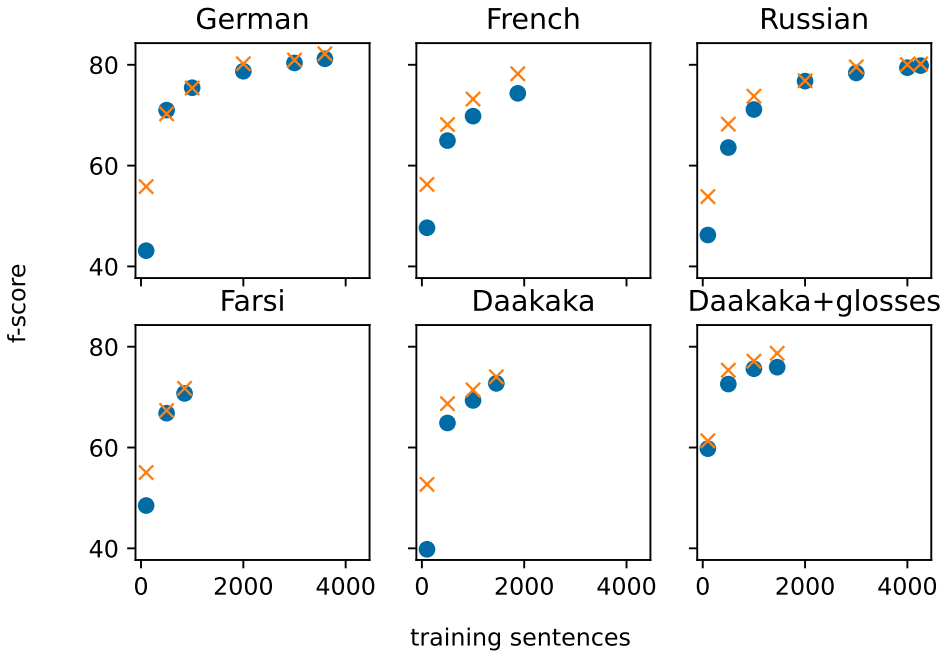


Figure 4: Parser accuracies on the development sets at 100, 500, 1000, 2000, 3000, 4000, and all available training sentences. Dots represent monolingual models, X's represent cross-lingual ones.

that come with different syntactic constructions, for instance the use of *de* and *à* as prepositions ('of' and 'to' respectively in English, 'von' and 'an' in German) or as clause linkage markers preceding an infinitive ('to' in English, 'zu' in German). With more training data and pretrained embeddings as input, this difference is reversed; Bladier et al. (2022) report better parsing results on the RRGparbank data for French and Russian than for German.

Impact of Self-training Now let us look at whether self-training improves the parsing performance, again for monolingual as well as cross-lingual parsing. Figure 5 plots the f-scores for the different languages and for different sizes of training data. Overall, we oftentimes see a positive effect of self-training, and this effect is stronger in the *extremely low* and *very low* resource scenarios (100 and 500 training sentences resp.). The effect is less visible with Daakaka with glosses, but the experiments on the test data (see below) will show that also for this scenario, self-training is actually helpful. It is surprising that, even when starting with only 100 sentences, which means that many constructions in the target language are not present in the training data, self-training improves results, at least for approximately the first two rounds.

In the cases where we start with 1000 training sentences or even with all available training data (note that these have different sizes), the effect of self training is less clear. In some cases, parsing performance even decreases slightly. This means that the trees added during self-training do not contain new knowledge about possible supertag combinations while containing probably errors that decrease parsing performance when used in training. Note however that these numbers are from just one run on the dev data. On the test data, when averaging over several runs, self-training helps for all languages when starting with 1000 training sentences (see below).

Zero-shot parsing with self-training We now investigate the scenario where we want to parse language data that has glosses and translations, but no syntactic annotations at all, not even a seed. We focus on the scenario where we do have access to a seed training set for a related language that also has translations, and glosses following a similar schema. This is also a realistic scenario in linguistic fieldwork, where linguists often investigate multiple related languages within a region. We take the example of Daakaka, for which we have a moderately sized training set, and the closely related language Dalkalaen, for which we have only created a small annotated test set (102 sentences). We apply our cross-lingual model to this scenario by first training our model on all Daakaka data

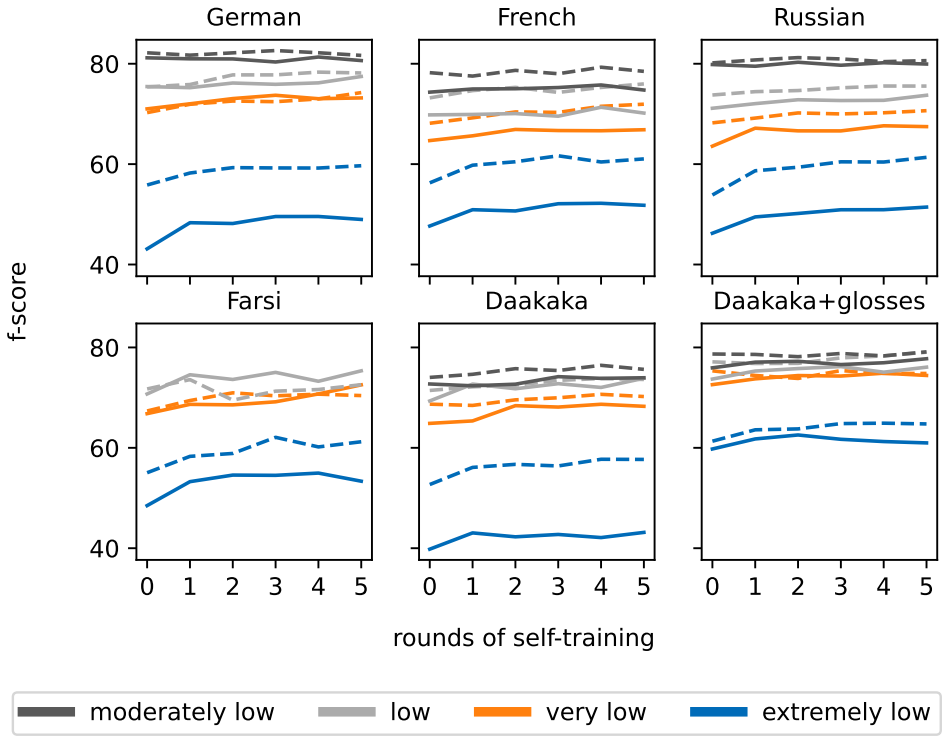


Figure 5: Effect of self-training in the *extremely low* (100 annotated sentences), *very low* (500), *low* (1 000; 851 for Farsi), and *moderately low* (3 594 for German, 1 875 for French, 4 259 for Russian, 1 455 for Daakaka) resource scenarios after 0, 1, 2, 3, 4, and 5 rounds of self-training, adding up to 500 sentences in each round (it can be less due to parse failures). Solid curves represent monolingual and dashed curves cross-lingual models.

(train, dev, and test), and then using the unannotated Dalkalaen data for self-training. The results are shown at the bottom of Table 2. It can be seen that even in this "no-resource" scenario, a performance comparable with the low-resource scenario can be reached.

Test Results After developing our models on the development data, we tested them on the test data. Here, we focus on the very low resource (500 annotated training sentences) and low resource (1000 sentences; 851 for Farsi) scenarios. We run each experiment five times with different random seeds (these affect the initial model parameters and the downsampling of the training data) and give the average f-scores in Table 2; the numbers in bold are the best results for a specific size of training data. The results confirm even more clearly that both cross-lingual transfer as well as self-training improve parsing results. In both cases (500 and 1 000 resp. 851 training sentences) the cross-lingual model systematically outperforms the monolingual model, sometimes by a large margin (see the column with 0 rounds of self-training). Furthermore, for all languages, self-training leads to further improvement, though not always with the same number of self-training rounds. For French in the case of 500 training sentences, after the first three rounds, the maximum score is already reached, and for Farsi and Daakaka with glosses in the very low resource scenario, the scores decrease after 4 rounds of self-training. For the other languages and training data sizes, continuing to 5 rounds brings further slight improvement. In most cases, the best score is reached with the combination of cross-lingual parsing and self-training, except for Farsi, where mono-lingual parsing benefits substantially from self-training while cross-lingual parsing improves only very little.

6 Conclusions

In this paper, we investigated constituency parsing, more precisely RRG parsing, for low-resource languages in a scenario where English translations, a limited set of annotated sentences in the target language and possibly also glosses are available. RRG is a theory that is widely used in typological research. We extended an existing RRG parser into a cross-lingual parser, exploiting the parallel

language	German						French					
self-training	0	1	2	3	4	5	0	1	2	3	4	5
very low, mono	68.9	70.1*	70.9*	71.5*	71.9*	72.2	62.6	64.9*	65.0	66.3*	66.3	66.4
very low, cross	69.0	70.2$^{*}_{\dagger}$	72.0$^{*}_{\dagger}$	73.3$^{*}_{\dagger}$	73.5$^{*}_{\dagger}$	**74.0**$_{\dagger}$	69.9$_{\dagger}$	70.4$_{\dagger}$	71.6$^{*}_{\dagger}$	71.9$_{\dagger}$	72.1$_{\dagger}$	**72.2**$_{\dagger}$
low, mono	74.3	74.7*	75.1	75.8*	75.8	76.0	69.4	70.6*	71.4*	71.5	72.0*	72.0
low, cross	74.7$_{\dagger}$	76.1$^{*}_{\dagger}$	77.5$^{*}_{\dagger}$	77.7$_{\dagger}$	78.2$^{*}_{\dagger}$	**78.3**$_{\dagger}$	74.1$_{\dagger}$	74.4$_{\dagger}$	75.3$_{\dagger}$	**75.9**$^{*}_{\dagger}$	75.7$_{\dagger}$	75.4$_{\dagger}$
language	Russian						Farsi					
self-training	0	1	2	3	4	5	0	1	2	3	4	5
very low, mono	65.8	67.3*	68.0*	68.6	69.2*	69.2	66.7	69.7*	70.2	71.1	**71.8**	71.0
very low, cross	69.9$_{\dagger}$	70.4$^{*}_{\dagger}$	71.4$^{*}_{\dagger}$	71.9$_{\dagger}$	72.0$^{*}_{\dagger}$	**72.2**$_{\dagger}$	69.2$_{\dagger}$	70.4	70.9	71.2	70.5	70.0
low, mono	71.7	73.1*	73.4	73.9*	74.3	74.4	71.3	74.1*	75.1*	74.8	74.9	**75.7**
low, cross	74.3$_{\dagger}$	75.1$^{*}_{\dagger}$	75.3$_{\dagger}$	75.9$^{*}_{\dagger}$	75.9$_{\dagger}$	**76.0**$_{\dagger}$	73.0$_{\dagger}$	73.2	73.4	74.2	73.9	73.9
language	Daakaka						Daakaka+glosses					
self-training	0	1	2	3	4	5	0	1	2	3	4	5
very low, mono	63.0	62.8	64.7*	64.9	65.1	65.1	67.9	69.5*	70.1*	70.7*	70.9	70.5
very low, cross	66.0$_{\dagger}$	67.0$^{*}_{\dagger}$	67.7$_{\dagger}$	68.3$_{\dagger}$	68.4$_{\dagger}$	**68.8**$_{\dagger}$	70.2$_{\dagger}$	70.7$_{\dagger}$	71.7$^{*}_{\dagger}$	72.2$^{*}_{\dagger}$	**72.4**$_{\dagger}$	72.2$_{\dagger}$
low, mono	67.6	68.1*	68.7	68.7	69.2	69.5	71.9	71.5	72.4*	72.9	73.4*	73.3
low, cross	70.4$_{\dagger}$	70.8$_{\dagger}$	70.8$_{\dagger}$	71.2$^{*}_{\dagger}$	71.1$_{\dagger}$	**71.3**$_{\dagger}$	73.1$_{\dagger}$	73.7$^{*}_{\dagger}$	74.3$_{\dagger}$	74.2$_{\dagger}$	74.5$_{\dagger}$	**74.7**$_{\dagger}$
language							Dalkalaen+glosses					
self-training							0	1	2	3	4	5
zero, cross							69.0	71.8*	72.4	73.0*	**73.6**	73.2

Table 2: Test results for the *very low* and *low* resource scenarios for German, French, Russian, Farsi, and Daakaka, as well as the *zero* resource scenario for Dalkalaen. All f-scores are averaged over 5 runs with different random initializations. *indicates that the result is significantly better ($p \leq .05$) than the corresponding result with one less round of self-training according to a permutation test (cf. Dror et al., 2018). $_{\dagger}$indicates the same for cross-lingual results compared to the corresponding monolingual model.

English data and unsupervised word alignments. Furthermore, we also adopted self-training, i.e., iteratively expanding the training data by adding the most confident new parses in each round. Our experiments showed that both self-training and cross-lingual parsing yield substantial and in almost all cases complementary improvements. Further experiments showed that even when only translations and glosses but no annotated sentences are available, self-training starting from a cross-lingual parser for a related language, based also on tokens and glosses, leads to considerable improvements and surprisingly good results.

Acknowledgments

We would like to thank the three anonymous reviewers for their valuable feedback. This work was carried out as a part of the research project TreeGraSP[3] funded by a Consolidator Grant of the European Research Council (ERC).

References

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

Tatiana Bladier, Kilian Evang, Valeria Generalova, Laura Kallmeyer, Robin Möllemann, Natalia Moors, Rainer Osswald, and Simon Petitjean. 2022. RRGparbank: A parallel role and reference grammar treebank. In *Proceedings of LREC*. To appear.

Tatiana Bladier, Laura Kallmeyer, Rainer Osswald, and Jakub Waszczuk. 2020a. Automatic extraction of tree-wrapping grammars for multiple languages. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 55–61, Düsseldorf, Germany. Association for Computational Linguistics.

Tatiana Bladier, Andreas van Cranenburgh, Kilian Evang, Laura Kallmeyer, Robin Möllemann, and Rainer Osswald. 2018. RRGbank: a role and reference grammar corpus of syntactic structures extracted from the Penn Treebank. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*. Linköping Electronic Conference Proceedings.

Tatiana Bladier, Jakub Waszczuk, and Laura Kallmeyer. 2020b. Statistical parsing of tree wrapping grammars. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6759–6766, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with

[3] https://treegrasp.phil.hhu.de

subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Adriane Boyd. 2007. Discontinuity revisited: An improved conversion to context-free representations. In *Proceedings of the Linguistic Annotation Workshop*, pages 41–44, Prague, Czech Republic. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Ayan Das, Affan Zaffar, and Sudeshna Sarkar. 2017. Delexicalized transfer parsing for low-resource languages using transformed and combined treebanks. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 182–190, Vancouver, Canada. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Kilian Evang, Tatiana Bladier, Laura Kallmeyer, and Simon Petitjean. 2021. Bootstrapping Role and Reference Grammar treebanks via Universal Dependencies. In *Proceedings of Universal Dependencies Workshop 2021 (UDW 2021)*.

Agnieszka Falenska and Özlem Çetinoğlu. 2017. Lexicalized vs. delexicalized parsing in low-resource scenarios. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 18–24, Pisa, Italy. Association for Computational Linguistics.

Zhongqiang Huang and Mary Harper. 2009. Self-training PCFG grammars with latent annotations across languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 832–841, Singapore. Association for Computational Linguistics.

Aravind K Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123. Springer.

Laura Kallmeyer, Rainer Osswald, and Robert D. Van Valin, Jr. 2013. Tree Wrapping for Role and Reference Grammar. In *Formal Grammar 2012/2013*, volume 8036 of *LNCS*, pages 175–190. Springer.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

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

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

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

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. 2019. Low-resource parsing with crosslingual contextualized representations. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 304–315, Hong Kong, China. Association for Computational Linguistics.

Robert Östling. 2015. *Bayesian Models for Multilingual Word Alignment*. Ph.D. thesis, Stockholm University.

Kilu von Prince. 2013a. *Daakaka, The Language Archive*. MPI for Psycholinguistics, Nijmegen.

Kilu von Prince. 2013b. *Dalkalaen, The Language Archive*. MPI for Psycholinguistics, Nijmegen.

Kilu von Prince. 2015. *A Grammar of Daakaka*. Mouton de Gruyter, Berlin, Boston.

Ines Rehbein. 2011. Data point selection for self-training. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 62–67, Dublin, Ireland. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Kiyoko Toratani and Carlos González Vergara. 2020. Role and reference grammar. https://rrg.caset.buffalo.edu/rrg/RRGBib_2020.pdf. Accessed: 2021-11-15.

Robert D. Van Valin, Jr. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge University Press.

Robert D. Van Valin, Jr. and William A. Foley. 1980. Role and reference grammar. In E. A. Moravcsik and J. R. Wirth, editors, *Current approaches to syntax*, volume 13 of *Syntax and semantics*, pages 329–352. Academic Press, New York.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1105–1116, Hong Kong, China. Association for Computational Linguistics.

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*.

A Hyperparamters

The hyperparameters of our models are given in Table 3.

	mono	cross
character-level LSTM embedding layers		
character size	25	25
depth	1	1
output size for words	300	300
output size: glosses	100 (0)	100 (0)
output size: POS	100 (0)	100 (0)
output dropout	0.1	0.1
common contextualization layer		
input size	500 (300)	500 (300)
output size	200	200
depth	2	2
dropout	0.1	0.1
output dropout	0.1	0.1
contextualization layer for taggers		
input size	400	800
output size	200	200
depth	2	2
dropout	0.1	0.1
output dropout	0.1	0.1
contextualization layer for biaffine layer		
input size	400	800
output size	200	200
depth	2	2
dropout	0.1	0.1
output dropout	0.1	0.1
auxiliary POS tagging layer		
input size	400	400
supertagging layer		
input size	400	400
biaffine dependency parsing layer		
input size	400	400
hidden size	100	100
output size	100	100
dropout	0.1	0.1
A* decoder (partage-twg)		
top-n tags given	15	15
top-n dep. heads	15	15
β probability cutoff factor (Clark and Curran, 2007)	0.01	0.01

Table 3: Hyperparameters in monolingual and cross-lingual models. Numbers in parentheses apply to models with no gloss embeddings. Contextualization layers for the cross-lingual model have twice as big inputs because they concatenate source-language and target-language representations.