Multilingual Projection for Parsing Truly Low-Resource Languages

Željko Agić[♡] **Anders Johannsen**[♡] **Barbara Plank**[♡♣]
Héctor Martínez Alonso[♡♠] **Natalie Schluter**[♡♢] **Anders Søgaard**[♡]
[♡] Center for Language Technology, University of Copenhagen, Denmark
[♣] Center for Language and Cognition, University of Groningen, The Netherlands
[♠] Univ. Paris Diderot, Sorbonne Paris Cité – Alpage, INRIA, France
[♢] MobilePay, Copenhagen, Denmark
{zeljko.agic,soegaard}@hum.ku.dk

Abstract

We propose a novel approach to cross-lingual part-of-speech tagging and dependency parsing for *truly* low-resource languages. Our annotation projection-based approach yields tagging and parsing models for over 100 languages. All that is needed are freely available parallel texts, and taggers and parsers for resource-rich languages. The empirical evaluation across 30 test languages shows that our method consistently provides top-level accuracies, close to established upper bounds, and outperforms several competitive baselines.

1 Introduction

State-of-the-art approaches to inducing part-of-speech (POS) taggers and dependency parsers only scale to a small fraction of the world's ∼6,900 languages. The major bottleneck is the lack of manually annotated resources for the vast majority of these languages, including languages spoken by millions, such as Marathi (73m), Hausa (50m), and Kurdish (30m). Cross-lingual transfer learning—or simply *cross-lingual learning*—refers to work on using annotated resources in other (source) languages to induce models for such low-resource (target) languages. Even simple cross-lingual learning techniques outperform unsupervised grammar induction by a large margin.

Most work in cross-lingual learning, however, makes assumptions about the availability of linguistic resources that do not hold for the majority of low-resource languages. The best cross-lingual dependency parsing results reported to date were presented by Rasooli and Collins (2015). They use the intersection of languages covered in the Google dependency treebanks project and those contained in the Europarl corpus. Consequently, they only consider closely related Indo-European languages for which high-quality tokenization can be obtained with simple heuristics.

In other words, we argue that recent approaches to cross-lingual POS tagging and dependency parsing are biased toward Indo-European languages, in particular the Germanic and Romance families. The bias is not hard to explain: treebanks, as well as large volumes of parallel data, are readily available for many Germanic and Romance languages. Several factors make cross-lingual learning between these languages easier: (i) We have large volumes of relatively representative, translated texts available for all language pairs; (ii) It is relatively easy to segment and tokenize Germanic and Romance texts; (iii) These languages all have very similar word order, making the alignments much more reliable. Therefore, it is more straightforward to train and evaluate cross-lingual transfer models for these languages.

However, this bias means that we possibly overestimate the potential of cross-lingual learning for truly low-resource languages, i.e., languages with no supporting tools or resources for segmentation, POS tagging, or dependency parsing.

The aim of this work is to experiment with cross-lingual learning via annotation projection, making minimal assumptions about the available linguistic resources. We only want to assume what we can in fact assume for truly low-resource languages. Thus, for the target languages, we do not assume the avail-

Transactions of the Association for Computational Linguistics, vol. 4, pp. 301–312, 2016. Action Editor: Chris Quirk.
Submission batch: 1/2016; Revision batch: 4/2016; 5/2016; 7/2016; Published 7/2016.

ability of any labeled data, tag dictionaries, typological information, etc. For annotation projection, we need a parallel corpus, and we therefore have to rely on resources such as the Bible (parts of which are available in 1,646 languages), and publications from the Watchtower Society (up to 583 languages). These texts have the advantage of being translated both conservatively and into hundreds of languages (massively multi-parallel). However, the Bible and the Watchtower are religious texts and are more biased than the corpora that have been assumed to be available in most previous work.

In order to induce high-quality cross-lingual transfer models from noisy and very limited data, we exploit the fact that the available resources are massively multi-parallel. We also present a novel multilingual approach to the projection of dependency structures, projecting edge *weights* (rather than edges) via word alignments from multiple sources (rather than a single source). Our approach enables us to project more information than previous approaches: (i) by postponing dependency tree decoding to after the projection, and (ii) by exploiting multiple information sources.

Our contributions are as follows:

(i) We present the first results on cross-lingual learning of POS taggers and dependency parsers, assuming only linguistic resources that are available for most of the world's written languages, specifically, Bible excerpts and translations of the Watchtower.

(ii) We extend annotation projection of syntactic dependencies across parallel text to the multi-source scenario, introducing a new, heuristics-free projection algorithm that projects weight matrices from multiple sources, rather than dependency trees or individual dependencies from a single source.

(iii) We show that our approach performs significantly better than commonly used heuristics for annotation projection, as well as than delexicalized transfer baselines. Moreover, in comparison to these systems, our approach performs particularly well on truly low-resource non-Indo-European languages.

All code and data are made freely available for general use.[1]

2 Weighted annotation projection

Motivation Our approach is based on the general idea of *annotation projection* (Yarowsky et al., 2001) using parallel sentences. The goal is to augment an unannotated target sentence with syntactic annotations projected from one or more source sentences through word alignments. The principle is illustrated in Figure 1, where the source languages are German and Croatian, and the target is English.

The simplest case is projecting POS labels, which are observed in the source sentences but unknown in the target language. In order to induce the grammatical category of the target word *beginning*, we project POS from the aligned words *Anfang* and *početku*, both of which are correctly annotated as NOUN. Projected POS labels from several sources might disagree for various reasons, e.g., erroneous source annotations, incorrect word alignments, or legitimate differences in POS between translation equivalents. We resolve such cases by taking a majority vote, weighted by the alignment confidences. By letting several languages vote on the correct tag of each word, our projections become more robust, less sensitive to the noise in our source-side predictions and word alignments.

We can also project syntactic dependencies across word alignments. If (u_s, v_s) is a dependency edge in a source sentence, say the ingoing dependency from *das* to *Wort*, u_s *(Wort)* is aligned to u_t *(word)*, and v_s *(das)* is aligned to v_t *(the)*, we can project the dependency such that (u_t, v_t) becomes a dependency edge in the target sentence, making *the* a dependent of *word*. Obviously, dependency annotation projection is more challenging than projecting POS, as there is a structural constraint: the projected edges must form a dependency tree on the target side.

Hwa et al. (2005) were the first to consider this problem, applying heuristics to ensure well-formed trees on the target side. The heuristics were not perfect, as they have been shown to result in excessive non-projectivity and the introduction of spurious relations and tokens (Tiedemann et al., 2014; Tiedemann, 2014). These design choices all lead to di-

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

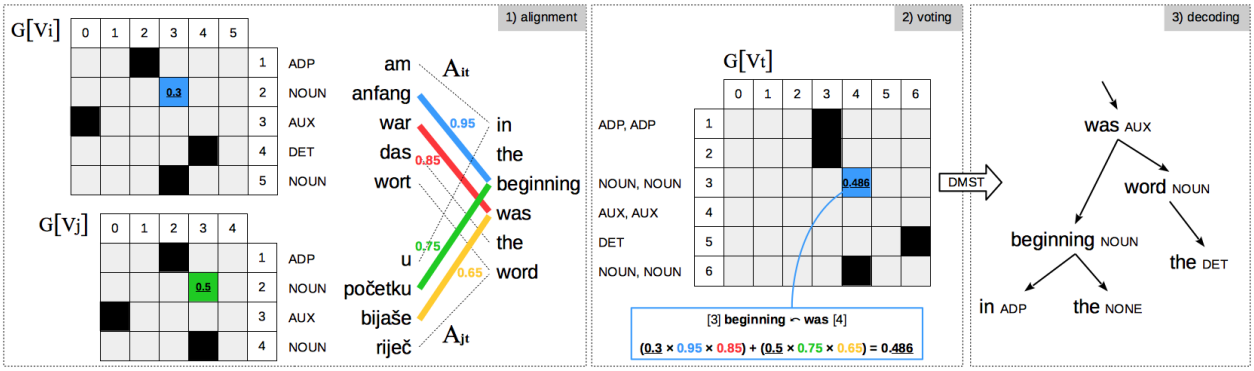

Figure 1: An outline of dependency annotation projection, voting, and decoding in our method, using two sources i (German) and j (Croatian) and a target t (English). Part 1 represents the multi-parallel corpus preprocessing, while parts 2 and 3 relate to our projection method. The graphs are represented as adjacency matrices with column indices encoding dependency heads. We highlight how the weight of target edge $(u_t = was, v_t = beginning)$ is computed from the two contributing sources.

minished parsing quality.

We introduce a heuristics-free projection algorithm. The key difference from most previous work is that we project the whole set of potential syntactic relations with associated weights—rather than binary dependency edges—from a large number of multiple sources. Instead of decoding the best tree on the source side—or for a single source-target sentence pair—we project weights prior to decoding, only decoding the aggregated multi-source weight matrix after the individual projections are done. This means that we do not lose potentially relevant information, but rather project dense information about all candidate edges.

2.1 Multi-source sentence graph

We assume the existence of n source languages and a target language t. For each tuple of translations in our multi-parallel corpus, our algorithm projects syntactic annotations from the n source sentences to the target sentence.

Projection happens at the sentence-level, taking a tuple of n annotated sentences and an unannotated sentence as input. We formalize the projection step as label propagation in a graph structure where the words of the target and source sentences are vertices, while edges represent dependency edge candidates between words within a sentence (a parse), as well as similarity relations between words of sentences in different languages (word alignments).

Formally, a *projection graph* is a graph $G = (V, E)$. All edges are weighted by the function $w_e \colon E \to \mathbb{R}$. The vertices can be decomposed into sets $V = V_0 \cup \cdots \cup V_n$, where V_i is the set of words in sentence i.

We often need to identify the target sentence $V_t = V_0$ and the source sentences $V_s = V_1 \cup \cdots \cup V_n$ separately. Edges between V_s and V_t are the result of word alignments. The alignment subgraph is the bipartite graph $A = (V_s, V_t, E_A)$, i.e., the subgraph of G induced by all (alignment) edges, E_A, connecting V_s and V_t.

The subgraph induced by the set of vertices V_i, written as $G[V_i]$, represents the dependency edge candidates between the words of the sentence i. In general these subgraphs are dense, i.e., they encode weight matrices of edge scores and not just the single best parse. For the source sentences, we assume that the weights are provided by a parser, while the weights for the syntactic relations of the target sentence are unknown.

With the above definitions, the dependency projection problem amounts to assigning weights to the edges of $G[V_t]$ by transferring the syntactic parse graphs $G[V_1], \ldots, G[V_n]$ from the source languages through the alignments A.

2.2 Part-of-speech projection

Our annotation projection for POS tagging is similar to the one proposed by Agić et al. (2015). The algorithm is presented in Algorithm 1. We first introduce a conditional probability distribution $p(l|v)$

Algorithm 1: Project POS tags

Data: A projection graph $G = (V_s \cup V_t, E)$; a set of POS labels L; a function $p(l|v)$ assigning probabilities to labels l for word vertices v.
Result: A labeling of V_t
$\tilde{p} \leftarrow$ empty probability table
label $\leftarrow$ empty label-to-vertex mapping
for $v_t \in V_t$ **do**
  for $l \in L$ **do**
    $\tilde{p}(l|v_t) \leftarrow \sum_{v_s \in V_s} p(l|v_s) w_a(v_s, v_t)$
  label$(v_t) \leftarrow \arg\max_l \tilde{p}(l|v_t)$
return *label*

Algorithm 2: Project dependencies

Data: A projection graph $G = (V_s \cup V_t, E)$.
Result: A dependency tree covering the target vertices V_t.
if *project from trees* **then**
  for *i=1* **to** *n* **do**
    $G[V_i] \leftarrow \text{DMST}(G[V_i])$
for $(u_t, v_t) \in G[V_t]$ **do**
  $w_e(u_t, v_t) \leftarrow -\infty$
  if $(\cdot, u_t)$ *or* $(\cdot, v_t) \notin E_A$ **then**
    continue
  $w_e(u_t, v_t) \leftarrow$
  $\sum_{i=1}^{n} \max_{u_s, v_s \in V_i} w_e(u_s, v_s)\ w_a(u_s, u_t)\ w_a(v_s, v_t)$
$G[V_t] \leftarrow \text{normalize}(G[V_t])$
return *DMST*$(G[V_t])$

over POS tags $l \in L$ for each vertex v in the graph. For all source vertices, the probability distributions are obtained by tagging the corresponding sentences in our multilingual corpus with POS taggers, assigning a probability of one to the best tag for each word, and zero for all other tags. For each target token, i.e., each vertex v, the projection works by gathering evidence for each tag from all source tokens aligned to v, weighted by the alignment score:

$$p(l|v_t) \quad \propto \sum_{v_s \in V_s} p(l|v_s)\ w_a(v_s, v_t)$$

The projected tag for a target vertex v_t is then $\arg\max_l p(l|v_t)$. When both the alignment weights and the source tag probabilities are in $\{0, 1\}$, this reduces to a simple voting scheme that assigns the most frequent POS tag among the aligned words to each target word.

2.3 Dependency projection

While in POS projection, we project vertex labels, in dependency projection we project edge scores. Our procedure for dependency annotation projection is given in Algorithm 2. For each source language, we parse the corresponding side of our multi-parallel corpus using a dependency parser trained on the source language treebank. However, instead of decoding to dependency trees, we extract the weights for all potential syntactic relations, importing them into G as edge weights.

The parser we use in our experiments assigns scores $w_e \in \mathbb{R}$ to possible edges. Since the ranges and values of these scores are dependent on the training set size and the number of model updates, we *standardize* the scores to make them comparable across languages. Standardization centers the scores around zero with a standard deviation of one by subtracting the mean and dividing by the standard deviation. We apply this normalization per sentence.

Scores are then projected from source edges to target edges via word alignments $w_a \in [0, 1]$. Instead of voting among the incoming projections from multiple sources, we sum the projected edge scores. Because alignments vary in quality, we scale the score of the projected source edge by the corresponding alignment probability.

A target edge $(u_t, v_t) \in G[V_t]$ can originate from multiple source edges even from a single source sentence, due to $m : n$ alignments. In such cases, we only project the source edge $(u_s, v_s) \in G[V_{i>0}]$ with the maximum score, provided the words are aligned, i.e., (u_s, u_t) and $(v_s, v_t) \in E_A$.

In the case of a single source sentence pair, the target edge scores are set as follows:

$$w_e(u_t, v_t) \leftarrow \max_{u_s, v_s \in V_i} \overbrace{w_e(u_s, v_s)}^{\text{edge}} \overbrace{w_a(u_s, u_t)\, w_a(v_s, v_t)}^{\text{alignment}}$$

We note the distinction between edge weights w_e and alignment weights w_a. With multiple sources,

the target edge scores $w_e(u_t, v_t)$ are computed as a sum over the individual sources:

$$\sum_{i=1}^{n} \max_{u_s, v_s \in V_i} w_e(u_s, v_s)\ w_a(u_s, u_t)\ w_a(v_s, v_t)$$

After projection we have a dense set of weighted edges in the target sentence representing possible syntactic relations. This structure is equivalent to the $n \times n$ edge matrix used in ordinary first-order graph-based dependency parsing.

Before decoding, the weights are softmax-normalized to form a distribution over each possible head decision. The normalization balances out the contributions of the individual head decisions; and in our development setup, we found that omitting this step resulted in a substantial ($\sim$10%) decrease in parsing performance.

We then follow McDonald et al. (2005) in using directed maximum spanning tree (DMST) decoding to identify the best dependency tree in the matrix. We note that DMST decoding on summed projected weight matrices is similar to the idea of re-parsing with DMST decoding of the output on an ensemble of parsers (Sagae and Lavie, 2006), which we use as a baseline in our experiments.

3 Data

3.1 Training and test sets

We use source treebanks from the Universal Dependencies (UD) project, version 1.2 (Nivre et al., 2015).[2] They are harmonized in terms of POS tag inventory (17 tags) and dependency annotation scheme. In our experiments, we use the canonical data splits, and disregard lemmas, morphological features and alternative POS from all treebanks.

Out of the 33 languages currently in UD1.2, we drop languages for which the treebank does not distribute word forms (Japanese), and languages for which we have no parallel unlabeled data (Latin, Ancient Greek, Old Church Slavonic, Irish, Gothic). Languages with more than 60k tokens (in the training data) are considered source languages, the remaining 6 smaller treebanks (Estonian, Greek, Hungarian, Latin, Romanian, Tamil) are strictly considered targets. This results in 22 treebanks for training source taggers and parsers. We use two additional test sets: Quechua and Serbian. The first one does not entirely adhere to UD, but we provide a POS tagset mapping and a few modifications and include it as a test language to deepen the robustness assessment for our approach across language families. The Serbian test set fully conforms to UD, as a fork of the closely related Croatian UD dataset.[3] This results in a total of 30 target languages.

3.2 Multi-parallel corpora

We use two sources of massively parallel text. The first is the Edinburgh Bible Corpus (EBC) collected by Christodouloupoulos and Steedman (2014) containing 100 languages. EBC has either 30k or 10k sentences for each language, depending on whether they are made up of full Bibles or just translations of the New Testament, respectively. We also crawled and scraped the Watchtower Online Library website to collect what we will refer to as the Watchtower Corpus (WTC).[4] The data is from 2002-2016 and the final corpus contains 135 languages with sentences in the range of 26k-145k. While some EBC Bibles are written in dated language, we do not make any modifications to the corpus if the language is also present in WTC. However, as Basque is not represented in WTC, we replace the Basque Bible from 1571 with a contemporary version from 2004, to enable the use of Basque in the parsing experiments.[5]

EBC and WTC both consist of religious texts, but they are very different in terms of style and content. If we examine Table 1 that shows the most frequent words per corpus, we observe that the English Bible—the King James Version from 1611—contains many Old English verb forms ("hath", "giveth"). In contrast, the English Watchtower is written in contemporary English, both in terms of verb inflection ("does", "says") and vocabulary ("today", "human"). WTC also deals with contemporary topics such as blood "transfusion" (36 mentions) and "computer" (42 mentions).

The other languages also show differences in terms of language modernity and dialectal difference between EBC and WTC. While each Bible translation has its individual history, Watchtower transla-

[2] http://hdl.handle.net/11234/1-1548

[3] https://github.com/ffnlp/sethr

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

[5] http://www.biblija.net/biblija.cgi?l=eu

EBC: *hath, saith, hast, spake, yea, cometh, iniquity, wilt, smote, shew, begat, doth, lo, hearken, thence, verily, neighbour, goeth, shewed, giveth, smite, didst, wherewith, knoweth, night*
WTC: *bible, does, however, says, today, during, show, human, later, important, really, humans, meetings, personal, states, future, fact, relationship, result, attention, someone, century, attitude, article, different*

Table 1: The 25 most frequent words exclusive to the English Bible or Watchtower.

tions are commissioned by the same publisher, following established editorial criteria. Thus, we not only expect Watchtower to yield projected treebanks that are closer to contemporary language, but also more reliable alignments. We expect these properties to make WTC a more suitable parallel corpus for our experiments and for bootstrapping treebanks for new languages.

3.3 Preprocessing

Segmentation For the multi-parallel corpora, we apply naive sentence splitting using full-stops, question marks and exclamation points of the alphabets from our corpora. We have collected these trigger symbols from the corpora, provided that they appeared as individual tokens at the ends of lines, and belonged to the "Punctuation, Other" Unicode category. After sentence splitting, we use naive whitespace tokenization.[6] We also remove short-vowel diacritics from all corpora written in Arabic script.

We use the same sentence splitting and tokenization for EBC and WTC. This is done regardless of Bibles being distributed in a verse-per-line format, which means verses can be split in more than one sentence. The average sentence length across languages is 18.5 tokens in EBC and 16.7 in WTC.

The UD treebank tokenization differs from the tokenization used for the multi-parallel corpora. The UD dependency annotation is based on *syntactic words*, and the tokenization guidelines recommend, for example, splitting clitics from verbs, and undoing contractions (Spanish "del" becomes "de el"). These tokens made up of several syntactic words are called *multiword tokens* in the UD convention, and are included in the treebanks but are not integrated in the dependency trees, i.e., only their forming subtokens are assigned a syntactic head.[7] In order to harmonize the tokenization, we eliminate subtokens from the dependency trees, and incorporate the original multiword tokens—which are more likely to be naive raw tokens—in the trees instead. For each multiword token, we provide it with POS and dependency label from the highest subtoken, namely the subtoken that is closest to root. For example, in the case of a verb and its clitics, the chosen subtoken is the verb, and the multiword token is interpreted as a verb. If there are more candidates, we select one through POS ranking.[8]

Alignment We sentence- and word-align all language pairs in both our multi-parallel corpora. We use `hunalign` (Varga et al., 2005) to perform conservative sentence alignment.[9] The selected sentence pairs then enter word alignment. Here, we use two different aligners. The first one is IBM2 `fastalign` by Dyer et al. (2013), where we adopt the setup of Agić et al. (2015) who observe a major advantage in using reverse-mode alignment for POS projection (4-5 accuracy points absolute).[10] In addition, we use the IBM1 aligner `efmaral`[11] by Östling (2015). The intuition behind using IBM1 is that IBM2 introduces a bias toward more closely related languages, and we confirm this intuition through our experiments. We modify both aligners so that they output the alignment probability for each aligned token pair.

Tagging and parsing The source-sides of the two multi-parallel corpora, EBC and WTC, are POS-tagged by taggers trained on the respective source languages, using TnT (Brants, 2000). We parse the corpora using TurboParser (Martins et al., 2013). The parser is used in simple arc-factored mode with pruning.[12] We alter it to output per-sentence arc

[6] `https://github.com/bplank/multilingualtokenizer`

[7] `http://universaldependencies.org/format.html`

[8] `https://github.com/coastalcph/ud-conversion-tools`.

[9] Parameters used: `utf`, `bisent`, `cautious`, `realign`.

[10] Parameters used: `d`, `o`, `v`, `r`.

[11] Also reverse mode, with default settings, see `https://github.com/robertostling/efmaral`.

[12] Parameters used: `basic`.

weight matrices.[13]

4 Experiments

Outline For each sentence in a target language corpus, we retrieve the aligned sentences in the source corpora. Then, for each of these source-target sentence pairs, we project POS tags and dependency edge scores via word alignments, aggregating the contributions of individual sources. Once all contributions are collected, we perform a per-token majority vote on POS tags and DMST decoding on the summed edge scores. This results in a POS-tagged and dependency parsed target sentence ready to contribute in training a tagger and parser.

We remove target language sentences that contain word tokens without POS labels. This may happen due to unaligned sentences and words. We then proceed to train models.

4.1 Setup

Each of the experiment steps involves a number of choices that we outline in this section. We also describe the baseline systems and upper bounds.

POS tagging Below, we present results with POS taggers based on annotation projection with both IBM1 and IBM2; cf. Table 3. We train TnT with default settings on the projected annotations. Note that we use the resulting POS taggers in our dependency parsing experiments in order not to have our parsers assume the existence of POS-annotated corpora.

For a more extensive assessment, we refer to the work by Agić et al. (2015) who report baseline and upper bounds. In contrast to their work, we consider two different alignment models and use the UD POS tagset (17 tags), in contrast to the 12 tags of Petrov et al. (2012). This makes our POS tagging problem slightly more challenging, but our parsing models potentially benefit from the extended tagset.[14]

Dependency parsing We use arc-factored TurboParser for all parsing models, applying the same setup as in preprocessing. There are three sets of models: our systems, baselines, and upper bounds. Our systems are trained on the projected EBC and WTC texts, while the rest—except system: DCA-PROJ (see below)—are trained on the (delexicalized) source-language treebanks.

To avoid a bias toward languages with big treebanks and to make our experiments tractable, we randomly subsample *all* training sets to a maximum of 20k sentences. In the multi-source systems, this means a uniform sample from all sources up to 20k sentences. This means our comparison is fair, and that our systems do not have the advantage of more training data over our baselines.

Our systems We report on four different cross-lingual systems, alternating the use of word aligners (IBM1, IBM2) and the structures we project, as they can be either (i) arc-factored weight matrices from the parser (GRAPHS) or (ii) the single-best trees provided by the parser after decoding (TREES). See the **if**-clause in Algorithm 2.

We tune two parameters for these four systems using English as development set, confidence estimation and normalization, and we report the best setups only. For the IBM1-based systems, we use the word alignment probabilities in the arc projection, but we use unit votes in POS voting. The opposite yields the best IBM2 scores: binarizing the alignment scores in dependency projection, while weight-voting the POS tags. We also evaluated a number of different normalization techniques in projection, only to arrive at standardization and softmax as by far the best choices.

Baselines and upper bounds We compare our systems to three competitive baselines, as well as three informed upper bounds or oracles. First, we list our baselines.

DELEX-MS: This is the multi-source direct delexicalized parser transfer baseline of McDonald et al. (2011).[15]

DCA-PROJ: This is the direct correspondence assumption (DCA)-based approach to projection, i.e., the *de facto* standard for projecting dependencies. First introduced by Hwa et al. (2005), it was recently elucidated by Tiedemann (2014), whose implementation we follow here. In contrast to our approach,

[13] Our fork of TurboParser is available from `https://github.com/andersjo/TurboParser`.

[14] For example, the AUX vs. VERB distinction from UD POS does not exist the tagset of Petrov et al. (2012), and neither does NOUN vs. PROPN (proper noun).

[15] Referred to as *multi-dir* in the original paper.

DCA projects trees on a source-target sentence pair basis, relying on heuristics and spurious nodes or edges to maintain the tree structure. In the setup, we basically plug DCA into our projection-voting pipeline instead of our own method.

REPARSE: For this baseline, we parse a target sentence using multiple single-source delexicalized parsers. Then, we collect the output trees in a graph, unit-voting the individual edge weights, and finally using DMST to compute the best dependency tree (Sagae and Lavie, 2006).

Now, we explain the three upper bounds:

DELEX-SB: This result is using the best single-source delexicalized system for a given target language following McDonald et al. (2013). We parse a target with multiple single-source delexicalized parsers, and select the best-performing one.

SELF-TRAIN: For this result we parse the target-language EBC and WTC data, train parsers on the output predictions, and evaluate the resulting parsers on the evaluation data. Note this result is available only for the source languages. Also, note that while we refer to this as self-training, we do *not* concatenate the EBC/WTC training data with the source treebank data. This upper bound tells us something about the usefulness of the parallel corpus texts.

FULL: Direct in-language supervision, only available for the source languages. We train parsers on the source treebanks, and use them to parse the source test sets.

Evaluation All our datasets—projected, training, and test sets—contain only the following CoNLL-X features: ID, FORM, CPOSTAG, and HEAD.[16] For simplicity, we do not predict dependency labels (DEPREL), and we only report unlabeled attachment scores (UAS). The POS taggers are evaluated for accuracy. We use our IBM1 taggers for all the baselines and upper bounds.

4.2 Results

Our average results are presented in Figure 2, including broken down by language family, the languages for which we had training data (Sources) and those for which we only had test data (Targets).

	Languages				
Baselines	All	Sources	Targets	IE	Non-IE
DELEX-MS	45.43⋆	45.64⋆	44.59⋆	49.53⋆	34.88†
DCA-PROJ	47.87†	47.05⋆	47.19⋆	51.33†	40.66†
REPARSE	47.79⋆	47.87⋆	47.47⋆	51.34⋆	38.67⋆
Our systems					
IBM1 GRAPHS	52.82⋆	53.01⋆	52.07⋆	55.44⋆	46.08⋆
TREES	**53.47**⋆	**53.49**⋆	**53.38**⋆	**55.91**⋆	**47.19**⋆
IBM2 GRAPHS	46.44†	46.14⋆	44.39⋆	49.54†	38.47†
TREES	46.48†	46.67⋆	45.54⋆	49.58†	38.93⋆
Upper bounds					
DELEX-SB	48.52⋆	48.64⋆	48.02⋆	50.91⋆	42.35⋆
SELF-TRAIN	—	58.38⋆	—	—	—
FULL	—	72.55⋆	—	—	—

Table 2: Overview of the parsing experiment results for the 25 languages in EBC ∩ WTC. We report the best average UAS score per system and language subset. IE: Indo-European languages, †: EBC, ⋆: WTC.

We see that our systems are substantially better than both multi-source delexicalized transfer, DCA, and reparsing based on delexicalized transfer models. Focusing on our system results, we see that projection with IBM1 leads to better models than projection with IBM2. We also note that our improvements are biggest with non-Indo-European languages. Our IBM1-based parsers top the ones using IBM2 alignment by 6 points UAS on Indo-European languages, while the difference amounts to almost 10 points UAS on non-Indo-European languages (cf. Table 2). This difference in scores exposes a systematic bias towards more closely related languages in work using even more advanced word alignment (Tiedemann and Agić, 2016).

The detailed results using the Watchtower Corpus are listed in Table 3, where we also list the POS tagging accuracies. Note that these are not directly comparable to Agić et al. (2015), since they use a more coarse-grained tagset, and the results listed here are using WTC. We list the detailed results with the Bible Corpus online.[17] The tendencies are the same, but the results are slightly lower almost consistently across the board.

Finally, we observe that our results are also better than those that can be obtained using a predictive model to select the best source language for delexi-

[16] http://ilk.uvt.nl/conll/#dataformat

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

	POS		MULTI-PROJ				*Baselines*			*Upper bounds*			
Sources	IBM1	IBM2	IBM1: GRAPHS	TREES	IBM2: GRAPHS	TREES	DELEX-MS	DCA-PROJ	REPARSE	DELEX-SB		SELF-TRAIN	FULL
Arabic	54.40	44.48	39.55	39.24	28.74	29.58	21.15	32.14	26.24	32.59	pl	49.47	70.79
Bulgarian	70.02	56.02	49.79	49.02	39.15	38.54	48.37	37.18	52.09	49.32	da	53.46	74.01
Croatian	76.56	74.21	55.96	55.33	49.20	50.34	45.49	50.56	48.69	46.68	cs	54.77	68.69
Czech	79.67	72.01	52.42	53.09	42.80	43.33	47.99	44.36	49.65	47.80	sl	57.77	70.36
Danish	86.20	84.66	61.27	62.26	54.82	56.41	55.96	58.64	57.13	56.21	no	64.69	71.23
Dutch	69.51	70.10	57.75	58.93	54.82	55.28	54.35	55.03	56.46	56.64	pt	61.64	72.23
English	78.92	77.22	60.00	61.21	56.46	56.72	53.87	57.12	55.13	52.62	no	66.53	76.18
Farsi	33.66	32.86	26.98	24.49	19.27	18.79	19.48	12.26	20.83	24.65	ar	22.62	64.86
Finnish	69.63	58.29	42.00	43.19	31.91	32.04	41.52	35.60	44.91	43.20	no	51.23	59.35
French	80.36	75.67	56.64	57.79	49.71	48.76	51.53	51.47	51.85	53.49	it	59.91	75.36
German	69.97	62.48	45.73	46.54	38.73	37.88	45.79	36.70	47.21	45.12	no	50.62	67.36
Hebrew	63.01	51.78	45.40	45.59	34.02	35.46	25.02	36.34	27.68	41.71	id	51.37	60.26
Hindi	50.52	42.11	16.84	17.05	15.34	14.36	21.04	10.77	20.90	25.06	fi	44.17	82.63
Indonesian	75.49	70.75	58.18	59.58	48.05	49.76	39.67	52.29	44.80	48.43	he	65.91	73.74
Italian	85.93	82.35	65.84	66.29	60.81	61.25	58.06	63.57	60.03	61.91	es	69.37	79.21
Norwegian	85.84	83.57	66.80	67.30	63.56	64.60	60.11	64.37	62.21	61.12	da	72.42	79.08
Polish	73.84	69.08	62.62	63.46	53.51	56.55	54.87	55.40	56.37	54.31	cs	63.80	76.37
Portuguese	84.22	82.33	63.94	64.91	61.01	61.59	56.99	63.16	58.27	57.79	es	68.80	77.66
Slovene	78.36	74.40	60.69	61.51	53.61	54.15	52.53	54.80	54.15	53.43	cs	63.76	73.63
Spanish	86.39	84.05	64.24	65.39	61.34	61.09	55.87	61.90	59.30	56.25	it	70.16	75.73
Swedish	86.28	84.43	65.28	66.52	60.74	62.16	57.48	62.45	59.94	61.12	no	66.80	74.86
Targets													
Estonian	75.76	68.76	63.43	65.94	51.31	57.58	48.48	58.41	54.34	58.62	no	—	—
Greek	75.04	63.57	60.86	61.69	50.19	50.06	54.90	52.95	59.29	55.27	no	—	—
Hungarian	73.70	69.52	47.80	50.84	44.83	45.38	46.66	42.33	49.85	47.62	fi	—	—
Quechua	19.49	15.19	26.17	25.93	23.15	22.74	21.67	27.48	22.87	24.30	pl	—	—
Romanian	78.08	74.67	62.08	62.52	52.46	51.95	51.23	54.78	51.01	54.27	id	—	—
Tamil	44.27	35.23	22.41	22.16	21.61	17.34	34.07	15.98	34.67	37.99	hi	—	—
Averages													
All	70.56	65.18	51.88	52.51	45.23	45.69	45.34	46.22	47.62	48.43		—	—
Sources	73.28	68.23	53.2	53.75	46.55	47.08	46.05	47.43	48.28	49.02		58.54	72.55
Targets	61.06	54.49	47.13	48.18	40.59	40.84	42.84	41.99	45.34	46.35		—	—

Table 3: POS tagging accuracies and UAS parsing scores for the models built using WTC data. The results are split for source and target languages. All baselines and upper bounds use IBM1 POS taggers, while our MULTI-PROJ systems use their respective IBM1 or IBM2 taggers.

calized transfer (Rosa and Žabokrtský, 2015); and better than what can be obtained using an oracle (DELEX-SB) to select the source language.

Direct supervision (FULL) upper bound unsurprisingly records the highest scores in the experiment, as it uses biased in-language and in-domain training data. We also experiment with learning curves for direct supervision, with a goal of establishing the amount of manually annotated sentences needed to beat our cross-lingual systems. We find that for most languages this number falls within the range of 100-400 in-domain sentences.

5 Discussion

Function words In UD, a subset of function words—tags: ADP, AUX, CONJ, SCONJ, DET, PUNCT—have to be leaves in the dependency trees, unless, e.g., they participate in multiword expressions. Our predictions show some violations of this constraint (less than 1% of all words with these POS), but this ratio is similar to the amount of violations found in the test data.

Projectivity The UD treebanks are in general largely projective. Our UD test languages have an average of 89% fully projective sentences. However, with IBM1 for example, we only predict 55% of all sentences to be projective. Regardless of the differences in UAS, we observe a corpus effect in the difference of projectivity of the predictions between using EBC (65%) and WTC (55%). We attribute the higher level of projectivity of EBC-projected treebanks to Bible sentences being shorter.

The least projective predictions are Farsi (17%) and Hindi (19%), for which we also obtain the lowest UASs. This may be a consequence of our naive tokenization, yielding unreliable alignments. However, projectivity correlates more with UAS ($\rho = 0.56$) than with POS prediction accuracy ($\rho = 0.34$).

Dependency length We observe that the average edge length on IBM1 and WTC is of 2.95, while for EBC it is 2.67. The average gold edge length is

3.6—which is significantly higher at $p < 0.05$ (Student's t-test). However, the variance in gold edge length is about 1.2 times the deviation of predicted edge length. In other words, gold edges are often longer and more far-reaching. This difference indicates our predictions have worse recall for longer dependencies such as subordinate clauses, while being more accurate in local, phrasal contexts.

POS errors Unlike most previous work on cross-lingual dependency parsing, and following the notable exception of McDonald et al. (2011), we rely on POS predictions from cross-lingual transfer models. One may hypothesize that there is a significant error propagation from erroneous POS projection. We observe, however, that about 40% of wrong POS predictions are nevertheless assigned the right syntactic head. We argue that the fairly uniform noise on the POS labels helps the parsers regularize over the POS-dependency relations.

Possible improvements We treat POS and syntactic dependencies as two separate annotation layers and project them independently in our approach. Moreover, we project edge scores for dependencies, in contrast to only the single-best source POS tags. Johannsen et al. (2016) introduce an approach to joint projection of POS and dependencies, showing that exploiting the interactions between the two layers yields even better cross-lingual parsers. Their approach also accounts for transferring tag distributions instead of single-best POS tags.

All the parsers in our experiments are restricted to 20k training sentences. EBC and WTC texts offer up to 120k training instances per language. We observe limited benefits of going beyond our training set cap, indicating a more elaborate instance selection-based approach would be more beneficial than just adding more training data.

In our dependency graph projection, we normalize the weights per sentence. For future development, we note that corpus-level normalization might achieve the same balancing effect while still preserving possibly important language-specific signals regarding structural disambiguations.

EBC and WTC constitute a (hopefully small) subset of the publicly available multilingual parallel corpora. The outdated EBC texts can be replaced by newer ones, and the EBC itself replaced or augmented by other online sources of Bible translations. Other sources include the UN Declaration of Human Rights, translated to 467 languages,[18] and repositories of movie subtitles, software localization files, and various other parallel resources, such as OPUS (Tiedemann, 2012).[19] Our approach is language-independent and would benefit from extension to datasets beyond EBC and WTC.

6 Related work

POS tagging While projection annotation of POS labels goes back to Yarowsky's seminal work, Das and Petrov (2011) recently renewed interest in this problem. Das and Petrov (2011) go beyond our approach to POS annotation by combining annotation projection and unsupervised learning techniques, but they restrict themselves to Indo-European languages and a coarser tagset. Li et al. (2012) introduce an approach that leverages potentially noisy, but sizeable POS tag dictionaries in the form of Wiktionaries for 9 resource-rich languages. Garrette et al. (2013) also consider the problem of learning POS taggers for *truly* low-resource languages, but suggest crowdsourcing such POS tag dictionaries.

Finally, Agić et al. (2015) were the first to introduce the idea of learning models for more than a dozen truly low-resource languages *in one go*, and our contribution can be seen as a non-trivial extension of theirs.

Parsing With the exception of Zeman and Resnik (2008), initial work on cross-lingual dependency parsing focused on annotation projection (Hwa et al., 2005; Spreyer et al., 2010). McDonald et al. (2011) and Søgaard (2011) simultaneously took up the idea of delexicalized transfer after Zeman and Resnik (2008), but more importantly, they also introduced the idea of *multi-source* cross-lingual transfer in the context of dependency parsing. McDonald et al. (2011) were the first to combine annotation projection and multi-source transfer, the approach taken in this paper.

Annotation projection has been explored in the context of cross-lingual dependency parsing since Hwa et al. (2005). Notable approaches include the

[18] http://www.ohchr.org/EN/UDHR/Pages/SearchByLang.aspx

[19] http://opus.lingfil.uu.se/

soft projection of reliable dependencies by Li et al. (2014), and the work of Ma and Xia (2014), who make use of the source-side distributions through a training objective function.

Tiedemann and Agić (2016) provide a more detailed overview of model transfer and annotation projection, while introducing a competitive machine translation-based approach to synthesizing dependency treebanks. In their work, we note the IBM4 word alignments favor more closely related languages, and that building machine translation systems requires parallel data in quantities that far surpass EBC and WTC combined.

The best results reported to date were presented by Rasooli and Collins (2015). They use the intersection of languages represented in the Google dependency treebanks project and the languages represented in the Europarl corpus. Consequently, their approach—similar to all the other approaches listed in this section—is potentially biased toward closely related Indo-European languages.

7 Conclusions

We introduced a novel, yet simple and heuristics-free, method for inducing POS taggers and dependency parsers for *truly* low-resource languages. We only assume the availability of a translation of a set of documents that have been translated into *many* languages. The novelty of our dependency projection method consists in projecting edge scores rather than edges, and specifically in projecting these annotations from multiple sources rather than from only one source. While we built models for more than a hundred languages during our experiments, we evaluated our approach across 30 languages for which we had test data. The results show that our approach is superior to commonly used transfer methods.

Acknowledgements We thank the editors and the anonymous reviewers for their valuable comments. This research is funded by the ERC Starting Grant LOWLANDS (#313695).

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*.

Thorsten Brants. 2000. TnT: A Statistical Part-of-Speech Tagger. In *ANLP*.

Christos Christodouloupoulos and Mark Steedman. 2014. A Massively Parallel Corpus: The Bible in 100 Languages. *Language Resources and Evaluation*, 49(2).

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

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *ACL*.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-World Semi-Supervised Learning of POS-Taggers for Low-Resource Languages. In *ACL*.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping Parsers via Syntactic Projection Across Parallel Texts. *Natural Language Engineering*, 11(3).

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

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly Supervised Part-of-Speech Tagging. In *EMNLP*.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Soft Cross-lingual Syntax Projection for Dependency Parsing. In *COLING*.

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

André F. T. Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers. In *ACL*.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online Large-Margin Training of Dependency Parsers. In *ACL*.

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

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*.

Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Cristina Bosco, Sam Bowman, Giuseppe G. A. Celano, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Tomaž Erjavec, Richárd

Farkas, Jennifer Foster, Daniel Galbraith, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Berta Gonzales, Bruno Guillaume, Jan Hajič, Dag Haug, Radu Ion, Elena Irimia, Anders Johannsen, Hiroshi Kanayama, Jenna Kanerva, Simon Krek, Veronika Laippala, Alessandro Lenci, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Shunsuke Mori, Hanna Nurmi, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Prokopis Prokopidis, Sampo Pyysalo, Loganathan Ramasamy, Rudolf Rosa, Shadi Saleh, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Kiril Simov, Aaron Smith, Jan Štěpánek, Alane Suhr, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Sumire Uematsu, Larraitz Uria, Viktor Varga, Veronika Vincze, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2015. Universal Dependencies 1.2.

Robert Östling. 2015. Word Order Typology Through Multilingual Word Alignment. In *ACL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *LREC*.

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

Rudolf Rosa and Zdeněk Žabokrtský. 2015. KLcpos3: A Language Similarity Measure for Delexicalized Parser Transfer. In *ACL*.

Kenji Sagae and Alon Lavie. 2006. Parser Combination by Reparsing. In *NAACL*.

Kathrin Spreyer, Lilja Øvrelid, and Jonas Kuhn. 2010. Training Parsers on Partial Trees: A Cross-Language Comparison. In *LREC*.

Anders Søgaard. 2011. Data Point Selection for Cross-Language Adaptation of Dependency Parsers. In *ACL*.

Jörg Tiedemann and Željko Agić. 2016. Synthetic Treebanking for Cross-Lingual Dependency Parsing. *Journal of Artificial Intelligence Research*, 55.

Jörg Tiedemann, Željko Agić, and Joakim Nivre. 2014. Treebank Translation for Cross-Lingual Parser Induction. In *CoNLL*.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *LREC*.

Jörg Tiedemann. 2014. Rediscovering Annotation Projection for Cross-Lingual Parser Induction. In *COLING*.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel Corpora for Medium Density Languages. In *RANLP*.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection Across Aligned Corpora. In *NAACL*.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation Between Related Languages. In *IJCNLP Workshop on NLP for Less Privileged Languages*.