Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging

Ayyoob Imani, Silvia Severini, Masoud Jalili Sabet, François Yvon, Hinrich Schütze

Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging

Ayyoob Imani [*1], **Silvia Severini**[*1],
Masoud Jalili Sabet[1], **François Yvon**[2], **Hinrich Schütze**[1]
[1]Center for Information and Language Processing (CIS), LMU Munich, Germany
[2]Université Paris-Saclay, CNRS, LISN, France
{ayyoob, silvia, masoud}@cis.lmu.de, francois.yvon@limsi.fr

Abstract

Part-of-Speech (POS) tagging is an important component of the NLP pipeline, but many low-resource languages lack labeled data for training. An established method for training a POS tagger in such a scenario is to create a labeled training set by transferring from high-resource languages. In this paper, we propose a novel method for transferring labels from multiple high-resource source to low-resource target languages. We formalize POS tag projection as graph-based label propagation. Given translations of a sentence in multiple languages, we create a graph with words as nodes and alignment links as edges by aligning words for all language pairs. We then propagate node labels from source to target using a Graph Neural Network augmented with transformer layers. We show that our propagation creates training sets that allow us to train POS taggers for a diverse set of languages. When combined with enhanced contextualized embeddings, our method achieves a new state-of-the-art for unsupervised POS tagging of low resource languages.

1 Introduction

In many applications, Part-of-Speech (POS) tagging is an important part of the NLP pipeline. In recent years, high-accuracy POS taggers have been developed owing to advances in machine learning methods that combine pretraining on large unlabeled corpora and supervised fine-tuning on well-curated annotated datasets. This methodology only applies to a handful of high-resource (HR) languages for which the necessary training data exists, leaving behind the majority of low-resource (LR) languages. When training resources are scarce, an established method for training POS taggers is to automatically generate the training data via cross-lingual transfer (Yarowsky and Ngai, 2001; Fossum and Abney, 2005; Agić et al., 2016; Eskander et al., 2020). Typically, POS annotations are projected through alignment links from the HR source to the LR target of a word aligned parallel corpus.

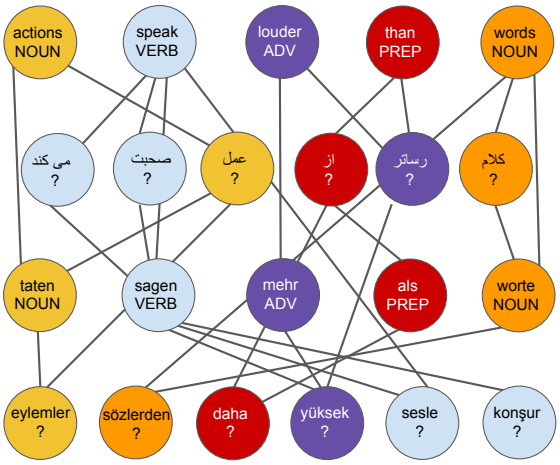

Figure 1: The sentence "Actions speak louder than words" in English and its translations in Persian, German, and Turkish, aligned at the word level. The POS tags for high-resource English and German are known. We use a GNN to exploit this graph structure and compute POS tags for low-resource Persian and Turkish.

In this paper, we propose GLP (Graph Label Propagation), a novel method for transferring labels simultaneously *from multiple high-resource source languages to multiple low-resource target languages*. We formalize POS tag projection as graph-based label propagation. Given translations of a sentence in multiple languages, we create a graph with words as nodes and alignment links as edges by aligning words for all language pairs. We then propagate POS labels associated with source language nodes to target language nodes, using a label propagation model that is formalized as a Graph Neural Network (GNN) (Scarselli et al., 2008). Nodes are represented by a diverse set of features that describe both linguistic properties and graph structural information. In a second step, we additionally employ self-learning to obtain reliable

[*]Equal contribution.

training instances in the target languages.

Our approach is based on *multiparallel corpora*, meaning that the translation of each sentence is available in more than two languages. We exploit the Parallel Bible Corpus (PBC) of Mayer and Cysouw (2014),[1] a multiparallel corpus that covers more than 1000 languages, many of which are extremely low-resource, by which we mean that only a tiny amount of unlabeled data is available or that no language technologies exist for them at all (Joshi et al., 2020).

We evaluate our method on a diverse set of low-resource languages from multiple language families, including four languages not covered by pretrained language models (PLMs). We train POS tagging models for these languages and evaluate them against references from the Universal Dependencies corpus (Zeman et al., 2019). We compare the results of our method against multiple state-of-the-art (SOTA) cross-lingual unsupervised and semisupervised POS taggers employing different approaches like annotation projection and zero-shot transfer. Our experiments highlight the benefits of our new transfer and self-learning methods; crucially, they show that reasonably accurate POS taggers can be bootstrapped without any annotated data for a diverse set of low-resource languages, establishing a new SOTA for high-resource-to-low-resource cross-lingual POS transfer. We also assess the quality of the projected annotations with respect to "silver" references and perform an ablation study. To summarize, our contributions are:[2]

- We formalize annotation projection as graph-based label propagation and introduce two new POS annotation projection models, GLP-B (GLP-Base) and GLP-SL (GLP-SelfLearning).
- We evaluate GLP-B and GLP-SL on 17 low-resource languages, including 4 languages not covered by large PLMs.
- By comparing our method with various supervised, semisupervised, and PLM-based approaches for POS tagging of low-resource languages, we establish a new SOTA for unsupervised POS tagging.

[1] We do not use PBC-specific features. Thus, our work is in principle applicable to any multiparallel corpus.

[2] Our code, data, and trained models are available at https://github.com/ayyoobimani/GLP-POS

2 Related work

POS tagging Part of Speech tagging aims to assign each word the proper syntactic tag in its context (Manning and Schütze, 1999). For high-resource languages, for which large labeled training sets are available, high-accuracy POS tagging is achieved through supervised learning (Kondratyuk and Straka, 2019; Tsai et al., 2019).

Zero-shot transfer In low-resource settings, one approach is to use cross-lingual transfer thanks to pretrained multilingual representations, thereby enabling zero-shot POS tagging. Kondratyuk and Straka (2019) analyze the few-shot and zero-shot performance of mBERT (Devlin et al., 2019) fine-tuning on POS tagging. We include this approach in our set of baselines below. Ebrahimi and Kann (2021) and Wang et al. (2022a) analyze zero-shot POS tagging performance of XLM-RoBERTa (Conneau et al., 2020) and propose complementary methods such as continued pretraining, vocabulary expansion and adapter modules for better performance. We show that combining GLP with Wang et al. (2022a)'s embeddings further improves our base performance.

Annotation projection is another approach for annotation of low-resource languages. Yarowsky and Ngai (2001) first proposed the idea of projecting annotation labels across languages exploiting parallel corpora and word alignment. To reduce systematic transfer errors, Fossum and Abney (2005) extended this by projecting from multiple source languages. Agić et al. (2015a) and Agić et al. (2016) exploit multilingual transfer setups to bootstrap POS taggers for low-resource languages starting from a parallel corpus and taggers and parsers for high-resource languages. Other works project labels by leveraging token and type-level constraints (Täckström et al., 2013; Buys and Botha, 2016a; Eskander et al., 2020). The latter study notably proposes an unsupervised method for selecting training instances via cross-lingual projection and trains POS taggers exploiting contextualized word embeddings, affix embeddings and hierarchical Brown clusters (Brown et al., 1992). This approach is also used as a baseline below.

Semi-supervised approaches have been proposed to mitigate the noise of projecting between languages. This can be achieved with auxiliary lexical resources (Täckström et al., 2013; Ganchev and Das, 2013; Wisniewski et al., 2014; Li et al.,

2012) that guide unsupervised learning or act as an additional training signal (Plank and Agić, 2018). Other works combine manual and projected annotations (Garrette and Baldridge, 2013; Fang and Cohn, 2016). We outperform prior works without the use of additional resources (such as dictionaries and annotations).

Graph Neural Networks Many natural and real-life structures like physical systems, social networks & interactions, and molecular fingerprints have a graphical structure (Liu and Zhou, 2020). Graph neural networks have been successfully used to model them. Applications include social spammer detection (Wu et al., 2020), learning molecular fingerprints (Duvenaud et al., 2015) and human motion prediction (Li et al., 2020). Recently, GNNs have been adopted for NLP tasks such as text classification (Peng et al., 2018), sequence labeling (Zhang et al., 2018; Marcheggiani and Titov, 2017), neural machine translation (Bastings et al., 2017; Beck et al., 2018), and alignment link prediction (Imani et al., 2022). As far as we know, our work is the first to form the annotation projection problem as graph-based label propagation.

Multiparallel corpora A multiparallel corpus provides the translations of a source text in more than two languages. A few such corpora (Agić and Vulić, 2019; Mayer and Cysouw, 2014; Tiedemann, 2012) provide sentence-level aligned text for hundreds or thousands of languages; for many of these languages only a tiny amount of digitized content is available (Joshi et al., 2020). Although the amount of text found in existing multiparallel corpora is far less than in monolingual corpora, we believe that they can serve as cross-lingual bridges, with which effective representation for low-resource languages can be derived. Highly multiparallel corpora have been used for expanding pretrained models to more languages (Ebrahimi and Kann, 2021; Wang et al., 2022b), word alignment improvement and visualization (ImaniGooghari et al., 2021; Imani et al., 2022), embedding learning (Dufter et al., 2018), and annotation projection (Agić et al., 2015b; Severini et al., 2022).

3 Method

We now introduce our *Graph Label Propagation* (GLP) method, which formalizes the problem of annotation projection as graph-based label propagation. We first describe the graph structure, then the features associated with each node, and finally the architecture of our model.

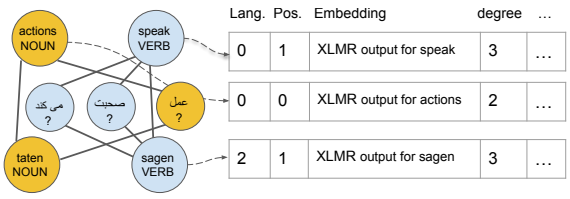

Figure 2: An example of how we represent nodes of an alignment graph using features for a part of the graph in Figure 1.

3.1 Problem formalization

The *multilingual alignment graph* (MAG) of a sentence is formalized as follows. Each sentence σ in our multiparallel corpus exists in a set L of languages.[3] L contains both high-resource source languages (in L_s) and low-resource target languages (in L_t) with $L_s \cup L_t = L$. Each word in these $|L|$ versions of σ will constitute a node in our graph. We first automatically annotate the text in all the source languages using pre-existing taggers: these POS tags are node labels; they are only known for languages in L_s, unknown otherwise. We then use Eflomal (Östling and Tiedemann, 2016), an unsupervised word alignment tool to compute alignment links for all $\frac{|L|*(|L|-1)}{2}$ language pairs: these links define the edges of our MAG. Figure 1 displays an example MAG for four languages, with English and German as sources and Turkish and Persian as targets. Note that both the word alignments and the node labels are noisy since we do not use gold data but statistical methods to generate them.

3.2 Features

To train graph neural networks, we represent each node using a set of features (Duong et al., 2019). In Figure 2 you see a simple illustration of how nodes are represented using a feature vector. The graph in this figure is part of the original graph in Figure 1. Two types of features are considered: features that represent the inherent meaning of a node/word (word representation features) and features that describe the position of a node within the graph (graph structural features). Word representation features consist of: XLM-R (Conneau et al., 2020) embeddings, the node's language and its position within the sentence. Since XLM-R embeddings are not available for all languages, we alternatively

[3] $|L|$ might be different for different sentences.

experiment with static word embeddings created using Levy et al. (2017)'s sentence-ID method, which we train on PBC. Our graph structural features are similar to Imani et al. (2022)'s work on link prediction. They include five centrality features: *degree, closeness* (Freeman, 1978), *betweenness* (Brandes, 2001), *load* (Newman, 2001), and *harmonic centrality* (Boldi and Vigna, 2014). Each of these features describes the node's position within the graph from a different perspective. For example, *degree* is the number of neighbors of the node and *harmonic centrality* measures how important/influential a node is. They also include two community features corresponding to the ID of the node's communities computed respectively with the greedy modularity community detection method of Clauset et al. (2004) and the label propagation algorithm of Cordasco and Gargano (2010). These two methods detect communities of nodes such that there are many links within the communities and only a few between them.

3.3 GLP architecture

Figure 3 displays the architecture of our GLP model; white nodes are for the source (= training) languages and green nodes for the target languages. The model has two parts: the GNN-based *encoder* turns the alignment graph into node representations and the *classifier* learns to label nodes based on these representations. The network is trained to reproduce POS tags for each source node; it is then used to predict the unknown tags for target nodes.

The encoder has two GATConv layers (Veličković et al., 2018): given a graph with M nodes represented as $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M$, with respective neighborhoods $\mathcal{N}(1), \mathcal{N}(2), ..., \mathcal{N}(M)$, a GATConv layer computes a new representation $\mathbf{x}'_i$ for each node as:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{i,j} \mathbf{W} \mathbf{x}_j, \qquad (1)$$

with $\mathbf{W}$ a learnable weight matrix. $\alpha_{i,j}$ measures how much node i "attends" to node j as follows:

$$\alpha_{i,j} = \frac{\exp\left(g\left(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp\left(g\left(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_i \,\|\, \mathbf{W}\mathbf{x}_k]\right)\right)}$$

where $\|$ stands for concatenation, g is the LeakyReLU (Maas et al.) and $\mathbf{a}$ is a weight vector. As neighborhoods only use alignment links, the representation of a node is only influenced by nodes in other languages. Also note that both source and target nodes are fed to the encoder.

We train two GLP models: *GLP-Base* (GLP-B) and *GLP-SelfLearning* (GLP-SL). The first one is the basic GNN architecture. It tags a token based on the other languages only, i.e., it makes no use of the sequence information of the token in its own language. The second additionally employs self-learning and is given access to the local context of each token in its own language.

GLP-B uses a multi-layer perceptron as classifier. We feed the node representations to the classifier and train the model end-to-end. We can only do this for source nodes since we have no training data for the target languages.

GLP-SL additionally employs self-learning and a better classifier. Self-learning takes advantage of node labels predicted by GLP-B in the first step: when the prediction confidence exceeds a threshold γ, these labels are deemed correct and the corresponding nodes are considered when training the classifier. GLP-SL uses a Transformer architecture to predict POS tags. The Transformer input consists of all translations of a sentence, where words are represented as GNN node embeddings. Each embedding is the concatenation of input (x_i) and output representations (x'_i) of the corresponding node in the GNN. In addition to the information available from neighbor nodes in *other* languages, the Transformer can attend to other words of the sentence in the *same* language, some of which may already be (automatically) labeled. This is very different from the training of GLP-B, where the POS of words of the same language were either all known (for source languages) or all unknown (for target languages), and explains why we resorted to a simpler classifier in the first stage.

Similar to (Eskander et al., 2020; Agić et al., 2016), GLP-SL uses type-level information: for each word type, we create a tag distribution by accumulating counts of the number of times each tag was assigned. For source words, we use the training data to estimate the distribution. For target words, we use the predictions of GLP-B on PBC.

3.4 Neural POS tagger

We use the noisy labeled data, generated by GLP-B or GLP-SL, to train monolingual neural POS taggers. Each model is a Bi-LSTM (Bidirectional Long Short-Term Memory, (Hochreiter and

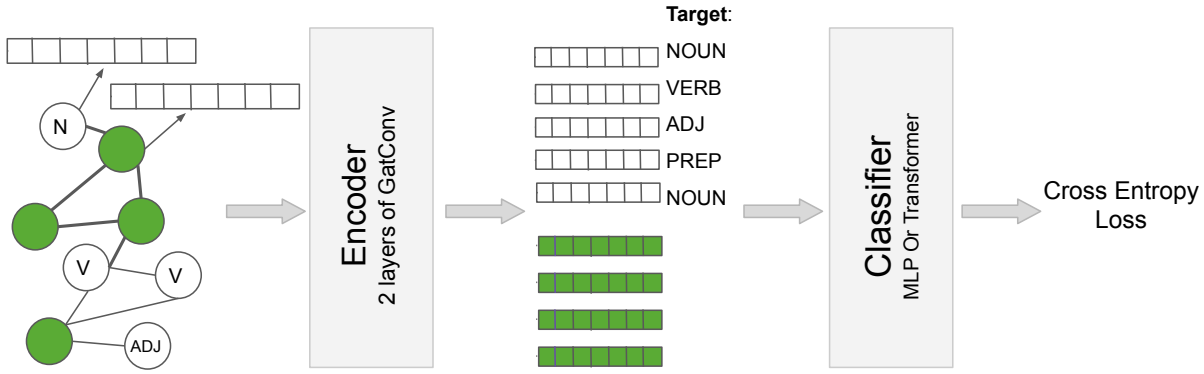

Figure 3: The architecture of GLP (Graph Label Projector). Source nodes are in white, target nodes in green. For training, we first feed the alignment graph of a sentence to the encoder to compute a representation for each node. Next we feed the representations of the source nodes to the classifier. The training objective is cross entropy on prediction of POS tags. Note that we know the POS tags of the source nodes. After training, the model can generalize the POS tag prediction to target nodes.

Schmidhuber, 1997)) with XLM-RoBERTa embeddings (Conneau et al., 2020). The input is a sentence labeled by GLP-B or GLP-SL. A token is assigned the NULL tag in case of missing labels. It is then ignored (i.e., masked) when computing the cross-entropy loss. To avoid predicting NULL, we set the corresponding output cell in the softmax to $-\infty$, similar to (Eskander et al., 2020).

4 Experimental setup

Table 1 gives our split of languages into training (15), development (4) and test (17) sets. The training set contains the source languages used for the transfer, while the development set languages are used as targets for parameter tuning. Training and test languages represent diverse language families and diverse availability. Note that training and dev languages are high-resource while test languages are low-resource. For most of the test languages, there are fewer than 8000 verses available in the Parallel Bible Corpus;[4] for Manx, fewer than 4000. We evaluate POS tagging performance on Universal Dependencies (UD) (Zeman et al., 2019) test sets. As UD and PBC tokenizations differ, we further adopt the following rule: if a PBC token corresponds to a sequence of several UD tokens, we replace the sequence with the original word, tagged with the tag of the UD token in the sequence that is highest in the dependency tree (cf. (Agić et al., 2016)). To tag the high-resource training and dev languages, we use Stanza (Qi et al., 2020),[5] a state-of-the-art NLP Python library. We create word alignments using Eflomal (Östling and Tiedemann, 2016),[6] a high-quality statistical word aligner, with the "intersection" symmetrization heuristic. Other than parallel data, Eflomal does not need any supervision signal; we can thus use it for any language pair in PBC. Details on models' hyperparameters are in Appendix A.3. All tagging results reported below are averages over three runs of the neural

	Lang	ISO	Family	# verses
Training languages	Arabic	arb	Afro-Asiatic, Semitic	31173
	Chinese	zho	Sino-Tibetan, Sinitic	31157
	Danish	dan	Indo-European, Germanic	31173
	English	eng	Indo-European	31099
	Finnish	fin	Uralic, Finnic	30200
	French	fra	Indo-European, Romance	31173
	German	deu	Indo-European, Germanic	31173
	Irish	gle	Indo-European, Celtic	34957
	Italian	ita	Indo-European, Romance	35377
	Polish	Pol	Indo-European, Slavic	31157
	Russian	rus	Indo-European, Slavic	31173
	Spanish	spa	Indo-European, Romance	31157
	Swedish	swe	Indo-European, Germanic	31157
	Tamil	tam	Dravidian, Southern Dravidian	7942
	Urdu	urd	Indo-European, Indic	7046
Dev languages	Czech	ces	Indo-European, Slavic	31157
	Greek	ell	Indo-European, Greek	31173
	Hebrew	heb	Afro-Asiatic, Semitic	23174
	Hungarian	hun	Uralic, Ugric	31157
Test languages	Afrikans	afr	Indo-European, Germanic	31157
	Amharic	amh	Afro-Asiatic, Semitic	7942
	Basque	eus	Basque, Basque	7958
	Belarusian	bel	Indo-European, Slavic	7958
	Bulgarian	bul	Indo-European, Slavic	31173
	Hindi	hin	Indo-European, Indic	7952
	Indonesian	ind	Austronesian, Malayo-Sumbawan	31157
	Lithuanian	lit	Indo-European, Baltic	31149
	Marathi	mar	Indo-European, Indic	7947
	Persian	pes	Indo-European, Iranian	7931
	Portuguese	pos	Indo-European, Romance	31157
	Telugu	tel	Dravidian, South-Central Dravidian	31163
	Turkish	tur	Altaic, Turkic	31157
	Bambara	bam	Mande, Western Mande	7958
	Erzya	myv	Uralic, Mordvin	7958
	Manx	glv	Indo-European, Celtic	3994
	Yoruba	yor	Niger-Congo, Defoid	30819

Table 1: Language family and number of verses in PBC for training, dev, and test languages in our experiments.

[4] Bible versions are described in Appendix A.1.

[5] https://stanfordnlp.github.io/stanza/

[6] 7github.com/robert/eflomal

POS tagger model.

5 Results

We evaluate GLP on 17 test languages from different families, resource availability, and scripts, on Universal Dependencies v2.10, the latest version (see details in Appendix A.2). Our results are in Table 2. For the four languages not supported by XLM-R, we report results obtained with static embeddings (see §3.2) in the GNN part (GLP-SL) and XLM-R embeddings only on the neural POS tagger model.[7] The best performance, > 89, is obtained for Bulgarian and Portuguese. All scores with XLM-R are above 80, except for Basque. This is probably because no language from the same family appears in the training set. Similarly, Turkish has the lowest performance among the other test languages. Scores without XLM-R are overall lower, yet competitive, showing that our projection method also works for very low-resource languages. Prior work has used older versions of UD . We now compare against each baseline, evaluating on the relevant version of UD in each case.

5.1 Annotation projection-based baselines

In this section, we compare with the unsupervised SOTA in cross-lingual POS tagging via annotation projection: ESKANDER (Eskander et al., 2020), AGIC (Agić et al., 2016) and BUYS (Buys and Botha, 2016b) as well as EFLOMAL. We also compare with a semi-supervised SOTA method that uses rapid annotation in addition to cross-lingual projection: CTRL (Cotterell and Heigold, 2017).

5.1.1 Fully unsupervised baselines

EFLOMAL is a simple projection method using alignment links followed by majority voting, similar to early annotation projection methods (Agić et al., 2015b; Fossum and Abney, 2005). We first align all target sentences with the corresponding sentences in all training languages with Eflomal (Östling and Tiedemann, 2016). Each target word is then tagged with the most common tag in the aligned source words. The annotation projection method ESKANDER (Eskander et al., 2020) uses alignment links and token and type constraints to project tags from source to target. The neural POS tagger features include XLM-R embeddings, affix embeddings, and word clusters created on PBC and Wikipedia of the target languages. Table 3 compares EFLOMAL, ESKANDER and GLP. In this table -Eng stands for when only English is used as the source language in GLP and -All stands for when all training languages are used (see §6.1). GLP outperforms both baselines in all cases but Indonesian, where ESKANDER is 0.7 points better. However, they tune their hyperparameters on this language using dev data while we only tune them on dev languages. Compared to ESKANDER, we use a simpler neural POS tagger and less resources, as we do not use affix embeddings nor word clusters. Our initial experiments indicated that word clusters were not helping in our setup. The higher quality of the annotated data created by GLP may already contain the information provided by word clusters.

Table 4 compares AGIC, BUYS, CTRL, and GLP-SL. AGIC (Agić et al., 2016) is a cross-lingual POS tagger for low-resource languages based on PBC excerpts and translations of the Watchtower.[8] BUYS (Buys and Botha, 2016b) extends previous approaches for projecting POS tags using bitexts to infer constraints on the possible tags for a given word type or token.

Table 4 shows that GLP outperforms AGIC and BUYS, except for Portuguese (BUYS), where our results are slightly below. BUYS projects from Spanish, which is closely related to Portuguese. Eskander et al. (2020) showed that it can be advantageous to transfer only from one closely related language as opposed to a mix of close and distant languages. Note that BUYS performance for Portuguese drops down to 84.3 when transferring from English. BUYS also uses Europarl[9] with up to 2M tokens which is closer in domain to UD than PBC. Thus, compared to BUYS, the parallel data we use are smaller, and from a more distant domain.

5.1.2 Semisupervised baseline

CTRL (Cotterell and Heigold, 2017) is a character-level recurrent neural network for multi-task cross-lingual transfer of morphological taggers. Their experiments include small sets of 100 and 1000 annotated target tokens. The bottom part of Table 4 shows that GLP-SL outperforms CTRL despite being fully unsupervised.

[7] XLM-R embeddings are used even for languages unseen during its pretraining as they improve performance. This is probably due to the fact that some words (e.g., names) can be well represented even for an unseen language.

[8] Obtained by crawling http://wol.jw.org

[9] http://www.statmt.org/europarl/

					with XLM-R								without XLM-R			
afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	bel	mar	bam	myv	glv	yor
87.7	82.4	70.9	90.1	81.8	85.3	85.7	81.8	89.2	83.8	80.1	85.9	87.9	65.4	64.4	63.9	59.9

Table 2: Accuracy on UD v2.10 test for GLP-SL when transferring from all training source languages (i.e., GLP-SL-All). See the other tables for comparison with prior work, which uses older versions of UD.

	afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	AVG	bel	mar	AVG
EFLOMAL-Eng	73.7	74.9	60.4	78.9	58.1	72.4	80.3	59.2	74.1	77.5	67.6	70.6	76.2	73.2	71.3
EFLOMAL-All	83.9	79.3	64.5	85.0	68.1	78.4	82.8	68.6	83.8	77.1	74.8	76.9	79.6	77.8	77.2
ESKANDER-Eng	86.9	75.3	67.3	85.6	73.9	**84.1**	80.9	77.2	86.1	80.0	74.3	79.2			
ESKANDER-All	89.3	79.3	67.1	88.2	72.8	83.0	82.5	77.3	87.8	77.1	74.6	79.9			
GLP-B-Eng	86.6	81.9	67.5	85.7	76.8	82.7	81.1	76.2	87.6	82.5	76.4	80.4	80.0	82.3	80.6
GLP-SL-Eng	84.4	81.9	68.6	84.0	75.8	81.3	81.0	73.5	86.4	80.6	75.8	79.4	75.1	81.5	79.2
GLP-B-All	**89.7**	**83.6**	67.4	**89.7**	79.9	82.8	**85.9**	79.6	87.7	81.4	**80.3**	82.5	87.9	83.2	83.0
GLP-SL-All	87.5	82.9	**70.6**	**89.7**	**81.9**	83.4	85.8	**81.9**	**89.6**	**83.7**	78.4	**83.2**	**88.8**	**88.4**	**84.0**

Table 3: Accuracy on UD v2.5 test for EFLOMAL, ESKANDER (Eskander et al., 2020) and GLP. "-Eng": transfer from English only. "-All": transfer from all training languages (see Eskander et al. (2020) and Table 1). Bold: best score for each language.

	Target	AGIC		GLP-SL-All
v1.2	bul	70.0	mul	**86.1**
	hin	50.5	mul	**79.0**
	ind	75.5	mul	**79.5**
	pes	33.7	mul	**75.2**
	por	84.2	mul	**87.7**
	Target	BUYS		GLP-SL-All
v1.2	bul	81.8	eng	**86.1**
	por	**88.0**	esp	87.7
	Target	CTRL		GLP-SL-All
v2.0	Bul	68.8	rus-100	**89.3**
	Bul	83.1	rus-1000	**89.3**
	Por	81.8	esp-100	**90.1**
	Por	88.9	esp-1000	**90.1**

Table 4: Accuracy on UD test for AGIC (Agić et al., 2016), BUYS (Buys and Botha, 2016b), CTRL (Cotterell and Heigold, 2017) and GLP-SL. We also report the source language or "mul" for multilingual, and for CTRL, the number of the supervision tokens.

5.2 Zero-shot baselines

Cross-lingual projection is also possible thanks to multilingual pretrained language models (PLMs). A PLM is first fine-tuned to POS tagging on source languages and then used to infer tags for target languages. While this approach performs well for some languages without requiring any parallel data, its performance tends to be poor for low-resource languages (Hu et al., 2021). Joshi et al. (2020) clusters languages into six groups based on the amount of available unlabeled and labeled data that exists for them. Groups 1 and 2 consist of languages such as Manx and Yoruba with the least amount of available data, while group 5 contains languages like English and Spanish with the largest amount of available monolingual and labeled data. We compare our approach with three baselines using test languages from groups 1 and 2.

mBERT based baselines: Kondratyuk and Straka (2019) use the zero-shot approach with multilingual BERT (Devlin et al., 2019) as PLM. We train our POS taggers using mBERT (instead of XLM-R) embeddings for a fair comparison. Table 5 displays the results for the low-resource languages in group 1 and 2 also available in the compared work. GLP-SL outperforms zero-shot in all cases by at least 12 percentage points. This result suggests that annotation projection using GLP is more effective than using multilingual representations for truly low-resource languages (i.e., languages from the first two groups of (Joshi et al., 2020)). To create proper representations for a language, PLMs require a huge amount of monolingual data that is not available for many languages. As Table 5 suggests, due to poor representations, zero-shot transfer to these languages is also poor. However, we could successfully exploit the Bible's parallel data in GLP for the benefit of these languages.

XLM-R based baselines: Ebrahimi and Kann (2021) continue pretraining PLMs on PBC and show that this boosts performance for languages unseen during the initial pretraining. Wang et al. (2022a) adapt PLMs to languages with little monolingual data using various sources of data includ-

	bam	myv	yor
Kondratyuk and Straka (2019)	30.9	46.7	50.9
GLP-SL-ALL	**65.5**	**64.6**	**63.3**

Table 5: POS tagging accuracy on UD v2.3 test for zero-shot mBERT and GLP-SL using mBERT embeddings.

ing PanLex lexicons,[10] translations of English Wikipedia to target languages and the JHU Bible corpus (McCarthy et al., 2020). These approaches are in fact complementary to GLP: we can equip GLP with better multilingual representations to further improve our results based on standard XLM-R. This is reflected in Table 6, where we report results for zero-shot baselines and combinations based on Wang et al. (2022a)'s improved XLM-R embeddings (instead of standard XLM-R) to represent tokens for the POS tagger. We see that these combinations lead to large performance improvements, establishing new SOTA results.

	bam	myv	glv
Ebrahimi and Kann (2021)	60.5	66.6	59.7
Wang et al. (2022a)	69.4	74.3	68.8
GLP-SL-ALL + wang-before	**71.1**	78.9	70.1
GLP-SL-ALL + wang-after	70.2	**80.6**	**70.7**

Table 6: Accuracy on UD v2.5 test for two baselines and for our method combined with (Wang et al., 2022a)'s XLM-R models before and after finetuning on the POS tagging task. ("glv" accuracy is on v2.7.)

6 Analysis

6.1 Ablation study

We conduct an ablation study to better understand what benefits our model.

"Eng" vs "All" Previous works highlighted the importance of a diverse set of source languages for cross-lingual transfer (Lin et al., 2019; Turc et al., 2021). The last four lines of Table 3 report GLP-B and GLP-SL results when transferring from English (i.e., using English as the only source), and when transferring from the full set of source languages (see Table 1). The transfer from English has lower performance than from all languages (except for a decrease from 67.5 to 67.4 for Basque/GLP-B). This means that our projection method does benefit from more data and from the rich information present in the diversity of source languages.

[10] https://panlex.org/snapshot/

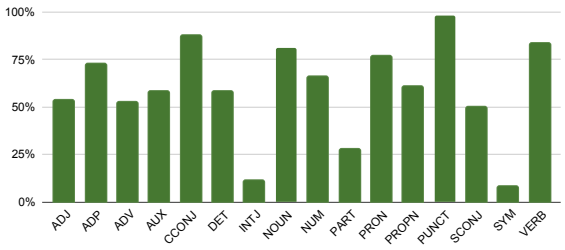

Figure 4: Average per tag accuracy of our GLP sets with respect to the "silver" reference.

GLP-B vs GLP-SL Table 3 reports results when training the neural POS tagger on GLP-B data and on GLP-SL data. GLP-B performs better than GLP-SL for four languages: Afrikaans, Lithuanian, Portuguese, and Turkish; but the performance difference is small (1.2 percentage points difference on average). In eight out of thirteen languages, GLP-SL gives better results (2.3 percentage points difference on average). This shows that the transformer architecture and the self-learning strategy are effective for most languages.

Contextualized vs. Static embeddings Our GLP models use XLM-R embeddings for languages for which they are available, otherwise static embeddings (see §3.2). In order to understand their usefulness in the transfer process, we compare with the performance obtained when static embeddings are used by GLP-SL. Results reported in Appendix B show an average improvement of 3 percentage points when XLM-R embeddings are used. The largest differences ($> 5\%$) are observed for Hindi, Persian, and Marathi. However, for the four languages not supported by XLM-R, the POS tagging accuracy is substantially lower when using contextualized embeddings compared to static embeddings (16.6 points drop on average).

6.2 Quality of artificial training sets

In order to evaluate the quality of the training sets generated by GLP-SL ("GLP sets"), we create a "silver" reference and compute the accuracy of GLP sets with respect to it. To build the silver reference, we annotate the training sets with the Stanza POS tagger for the languages for which it is available (12 out of 17). We obtain an average accuracy of 78.7, with Belarusian being the best and Basque the worst. The best predicted tokens are punctuation marks, coordinating conjunctions, and verbs, while the worst ones are symbols, interjections, and particles (see Figure 4). The high accuracy of

78.7 illustrates the ability of GLP-SL to successfully project annotations from high to low-resource languages.

7 Conclusion and future work

We presented GLP, a novel method for transferring labels from high-resource source to low-resource target languages, based on a formalization of annotation projection as graph-based label propagation. We exploited the Parallel Bible Corpus and showed that reasonably accurate POS taggers can be bootstrapped from projected labels. Since we do not use PBC-specific or language-specific features, GLP is in principle applicable to the more than 1000 languages of PBC and to any other multiparallel corpus.

One direction for the future is to employ a similar model to transfer higher-level structures such as dependency trees. Since our method works with graphical structures, one might be able to project dependency trees effectively. We could also extend our projection method to other tagging tasks like named entity recognition – although this requires using other parallel corpora to mitigate the domain shift problem of such a task. Another line for future work is to study the best combinations of source languages to transfer to any target language.

Limitations

Our method is evaluated on 17 languages carefully chosen to be from different families and scripts. However, we don't consider the other languages (more than 1000) in PBC due to computational constraints and lack of test sets.

A limitation of the GLP is that training over a MAG (multilingual alignment graph) created for all PBC languages requires a prohibitively large amount of resources, and based on our experiments, if we use a larger number of target languages at the same time, the performance will likely drop. Therefore one has to process languages in smaller batches (in our case, 36 languages). Accordingly, to cover all PBC subcorpora, $1341/36 = 38$ GLP models should in principle be trained.

Ethic statement

Our work is based on the Parallel Bible Corpus of Mayer and Cysouw (2014) that consists of Bible verses and is tested on the Universal Dependency treebanks (Zeman et al., 2019), an ensemble of different data sources. We would like to clarify that we treat the data simply as a multiparallel corpus, and the content does not necessarily reflect the opinions of the authors nor of the institutions funding the authors.

Acknowledgements

We would like to thank Sebastian Ruder for his kind and thoughtful suggestions on this work.

This work was funded by the European Research Council (grant #740516) and the German Federal Ministry of Education and Research (BMBF, grant #01IS18036A).

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015a. If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272, Beijing, China. Association for Computational Linguistics.

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015b. If all you have is a bit of the bible: Learning pos taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 268–272.

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

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–283, Melbourne, Australia. Association for Computational Linguistics.

Paolo Boldi and Sebastiano Vigna. 2014. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262.

Ulrik Brandes. 2001. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Jan Buys and Jan A. Botha. 2016a. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1954–1964, Berlin, Germany. Association for Computational Linguistics.

Jan Buys and Jan A Botha. 2016b. Cross-lingual morphological tagging for low-resource languages. *arXiv preprint arXiv:1606.04279*.

Aaron Clauset, Mark EJ Newman, and Cristopher Moore. 2004. Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Gennaro Cordasco and Luisa Gargano. 2010. Community detection via semi-synchronous label propagation algorithms. In *2010 IEEE international workshop on: business applications of social network analysis (BASNA)*, pages 1–8. IEEE.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. *arXiv preprint arXiv:1801.06807*.

Chi Thang Duong, Thanh Dat Hoang, Ha The Hien Dang, Quoc Viet Hung Nguyen, and Karl Aberer. 2019. On node features for graph neural networks. *arXiv preprint arXiv:1911.08795*.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4555–4567, Online. Association for Computational Linguistics.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4820–4831, Online. Association for Computational Linguistics.

Meng Fang and Trevor Cohn. 2016. Learning when to trust distant supervision: An application to low-resource POS tagging using cross-lingual projection. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 178–186, Berlin, Germany. Association for Computational Linguistics.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

Linton C Freeman. 1978. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Kuzman Ganchev and Dipanjan Das. 2013. Cross-lingual discriminative learning of sequence models with posterior regularization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1996–2006, Seattle, Washington, USA. Association for Computational Linguistics.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3633–3643, Online. Association for Computational Linguistics.

Ayyoob Imani, Lütfi Kerem Senel, Masoud Jalili Sabet, François Yvon, and Hinrich Schuetze. 2022. Graph neural networks for multiparallel word alignment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1384–1396, Dublin, Ireland. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Masoud Jalili Sabet, Philipp Dufter, Michael Cysou, and Hinrich Schütze. 2021. ParCourE: A parallel corpus explorer for a massively multilingual corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 63–72, Online. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. 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 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Omer Levy, Anders Søgaard, and Yoav Goldberg. 2017. A strong baseline for learning cross-lingual word embeddings from sentence alignments. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 765–774, Valencia, Spain. Association for Computational Linguistics.

Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 214–223.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1389–1398, Jeju Island, Korea. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Zhiyuan Liu and Jie Zhou. 2020. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(2):1–127.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. Citeseer.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Mark EJ Newman. 2001. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E*, 64(1):016132.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.

Barbara Plank and Željko Agić. 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 614–620, Brussels, Belgium. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Silvia Severini, Ayyoob Imani, Philipp Dufter, and Hinrich Schütze. 2022. Towards a broad coverage named entity resource: A data-efficient approach for many diverse languages. *arXiv preprint arXiv:2201.12219*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. 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 3632–3636.

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2106.16171*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022a. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022b. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1779–1785.

Yongji Wu, Defu Lian, Yiheng Xu, Le Wu, and Enhong Chen. 2020. Graph convolutional networks with markov random field reasoning for social spammer detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1054–1061.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Daniel Zeman, Joakim Nivre, and Mitchell et al. Abrams. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia. Association for Computational Linguistics.

A Reproducibility details

A.1 Data editions

Table 7 lists the edition used for all the experiments in this paper.

A.2 Universal Dependency tests specification

Table 8 lists the Universal Dependency testsets used in our experiments.

Lang	Edition	Lang	Edition
Arabic	arb-x-bible	Hungarian	hun-x-bible-newworld
Chinese	zho-x-bible-newworld	Afrikansc	afr-x-bible-newworld
Danish	dan-x-bible-newworld	Amharic	amh-x-bible-newworld
English*	eng-x-bible-mixed	Basque	eus-x-bible-navarrolabourdin
Finnish*	fin-x-bible-helfi	Belarusian	bel-x-bible-bokun
French	fra-x-bible-louissegond	Bulgarian	bul-x-bible-newworld
German	deu-x-bible-bolsinger	Hindi	hin-x-bible-bsi
Irish	gle-x-bible	Indonesian	ind-x-bible-newworld
Italian	ita-x-bible-2009	Lithuanian	lit-x-bible-ecumenical
Polish	pol-x-bible-newworld	Marathi	mar-x-bible
Russian	rus-x-bible-newworld	Persian	pes-x-bible-newmillennium2011
Spanish	spa-x-bible-newworld	Portuguese	por-x-bible-newworld1996
Swedish	swe-x-bible-newworld	Telugu	tel-x-bible
Tamil	tam-x-bible-newworld	Turkish	tur-x-bible-newworld
Urdu	urd-x-bible-2007	Bambara	bam-x-bible
Czech	ces-x-bible-newworld	Erzya	myv-x-bible
Greek	ell-x-bible-newworld	Manx	glv-x-bible
Hebrew*	heb-x-bible-helfi	Yoruba	yor-x-bible-2010

Table 7: PBC editions for all used languages. *Edition from Imani et al. (2022).

Lang	Test
Afrikaans	af_afribooms-ud-test
Amharic	am_att-ud-test
Basque	eu_bdt-ud-test
Belarusian	be_hse-ud-test
Bulgarian	bg_btb-ud-test
Hindi	hi_hdtb-ud-test
Ind	id_gsd-ud-test
Lithuanian	lt_alksnis-ud-test
Marathi	mr_ufal-ud-test.
Persian	fa_seraji-ud-test
Portuguese	pt_bosque-ud-test
Telugu	te_mtg-ud-test
Turkish	tr_imst-ud-test
Bambara	bm_crb-ud-test
Erzya	myv_jr-ud-test
Manx	gv_cadhan-ud-test
Yoruba	yo_ytb-ud-test

Table 8: Universal Dependency test sets used in our experiments.

A.3 Models parameters

GLP The GLP is implemented using the PyTorch geometric library.[11] All hyperparameters are tuned on the dev set. GLP-B has 2 layers of MLP of size 2048 while GLP-SL uses four layers of transformer with hidden size 2048 and 16 attention heads. Although we didn't observe a difference between different sizes from 512 to 2048. We tuned the learning rate, batch size, and γ (the self-learning threshold) over the validation languages. In GLP-B learning rate and batch size are respectively 0.001, 8, and in GLP-SL 0.00001, and 32. In general, when using XLM-R embeddings, the model has higher confidence, so the γ parameter is set to 0.95 when not using XLM-R embeddings and 0.98 when using XLM-R embeddings. The whole model needs about $16GB$ of GPU memory. GLP-B takes about 2 hours to train and GLP-SL about 12 hours. We used early stopping with patience of 8 for both GLP-B and GLP-SL.

Neural POS tagger We run our method on up to 48 cores of Intel(R) Xeon(R) CPU E7-8857 v2 with 1TB memory and a single GeForce GTX 1080 GPU with 8GB memory. The POS tagger uses the Flair framework (Akbik et al., 2019) and SequenceTagger model with 128 hidden size, the "xlm-roberta-base" embeddings, and AdamW optimizer Loshchilov and Hutter (2018). The hyperparameters, including the fixed number of epochs (15) are tuned using the UD development sets of the development languages. Each Neural POS tagger was trained in less than 30 minutes.

B Contextualized vs. Static embeddings

Table 9 shows results obtained with our GLP-SL with and without using XLM-R embeddings for projection. Note that the final neural POS tagger models always use XLM-R embeddings, even for languages unseen during XLM-R pretraining.

[11] https://pytorch-geometric.readthedocs.io/en/latest/

	afr	amh	eus	bul	hin	ind	lit	pes	por	tel	tur	bel	mar	AVG	bam	myv	glv	yor	AVG
with XLM-R	87.7	82.4	70.9	90.1	81.8	85.3	85.7	81.8	89.2	83.8	80.1	85.9	87.9	**84.1**	43.0	55.2	50.0	39.0	46.8
without XLM-R	88.4	82.8	72.7	89.3	73.7	80.2	83.9	71.3	85.0	80.1	77.8	85.2	82.0	81.0	65.4	64.4	63.9	59.9	**63.4**

Table 9: Accuracy on UD v2.10 for GLP-SL when transferring from all training languages (i.e., GLP-SL-All) with and without using XLM-R for the transfer in GLP-SL.