PD3: Better Low-Resource Cross-Lingual Transfer By Combining Direct Transfer and Annotation Projection

Steffen Eger, Andreas Rücklé, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science
Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

We consider unsupervised cross-lingual transfer on two tasks, viz., sentence-level argumentation mining and standard POS tagging. We combine direct transfer using bilingual embeddings with annotation projection, which projects labels across unlabeled parallel data. We do so by either merging respective source and target language datasets or alternatively by using multi-task learning. Our combination strategy considerably improves upon both direct transfer and projection with few available parallel sentences, the most realistic scenario for many low-resource target languages.

1 Introduction

In recent years, interest in multi- and cross-lingual natural language processing (NLP) has steadily increased. This has not only to do with the recognition that performances of newly introduced systems should be robust across several tasks (in several languages), but more fundamentally with the idea of truly 'universal' NLP methods which should not only suit English, an arguably particularly simple exemplar of the world's roughly 7,000 languages.

A further motivation for cross-lingual approaches is the fact that many labeled datasets are to this date only available in English and labeled data is generally costly to obtain—be it via expert annotators or through crowd-sourcing. Therefore, methods which are capable of training on labeled data in a resource-rich language such as English and which can then be applied to typically resource-poor other languages are highly desirable.

Two standard cross-lingual approaches are projection (Yarowsky et al., 2001; Das and Petrov, 2011; Täckström et al., 2013; Agic et al., 2016) and direct transfer (McDonald et al., 2011). Direct transfer trains, in the source language L1, on language-independent or shared features and then directly applies the trained system to the target language of interest L2. In contrast, projection trains and evaluates on L2 itself. To do so, it uses parallel data, applies a system trained on L1 to its source side and then projects the inferred labels to the parallel L2 side. This projection step may involve word alignment information. After projection, an annotated L2 dataset is available on which L2 systems can be trained.

Projection and direct transfer each ignore important information, however. For example, standard projection ignores the available data in L1 once the L2 dataset has been created and standard direct transfer does not use any L2 information.

In this work, we investigate whether the inclusion of both L1 and L2 data outperforms transfer approaches that exploit only one type of such information, and if so, under what conditions. More precisely, we first train a system on shared features as in standard direct transfer on labeled L1 data. Then, we make use of two further datasets. One is based on the source side of parallel unlabeled data; it is derived similarly as in self-training (Yarowsky, 1995) by applying the trained system to unlabeled data, from which a pseudo-labeled dataset is derived. The other is based on its target side—using annotation projections—as in standard projection. Thus, we explore the effects of combining *P*rojection and *D*irect transfer using three datasets (PD3). Our approach is detailed in §2.

We report results for two L2 languages (French, German) on one sentence-level problem (argumentation mining) and one token-level problem (POS tagging). We find that our suggested approach PD3 substantially outperforms both direct transfer and projection when little parallel data is available, the most realistic scenario for many L2 languages.

While our approach is general, our focus is particularly on argumentation mining (ArgMin), a rapidly growing research field in NLP. Cross-

Proceedings of the 5th Workshop on Argument Mining, pages 131–143
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

lingual transfer is majorly important for ArgMin because it is inherently costly to get high-quality annotations for ArgMin due to: (i) subjectivity of argumentation as well as divergent and competing ArgMin theories (Daxenberger et al., 2017; Schulz et al., 2018), leading to disagreement among crowd-workers as well as expert annotators (Habernal and Gurevych, 2017), (ii) dependence of argument annotations on background knowledge and parsing of complex pragmatic relations (Moens, 2017). Thus, in order not to reproduce the same annotation costs for new languages, cross-lingual ArgMin methods are required. These techniques should both perform well with little available parallel data, to address many languages, and with general (non-argumentative) parallel data, because this is much more likely to be available. Our experiments address both of these requirements.[1]

2 PD3

Let $\mathcal{L}_S = \{(x^S, y^S)\}$ denote a set of L1 data points in which each x^S is an instance and y^S its corresponding label. We assume that x^S is either a sentence or a sequence of tokens, and y^S is either a single label or contains one label for each token in the sequence. We assume access to a set $\mathcal{U}^{S,T} = \{(x^S, x^T)\}$ of unlabeled L1 and L2 data points in which target and source instances x^T and x^S are translations of each other. We let $\mathcal{U}_S = \mathcal{U}_S^{S,T}$ stand for the L1 part of $\mathcal{U}^{S,T}$, i.e., $\mathcal{U}_S$ consists of the data points x^S only: $\mathcal{U}_S = \{x^S \,|\, (x^S, x^T) \in \mathcal{U}^{S,T}\}$. We let $\mathcal{U}_T = \mathcal{U}_T^{S,T}$ be analogously defined. Finally, we assume that our instances x^S and x^T have a shared representation, e.g., that their words have a bilingual vector space representation in which mono- and cross-lingually similar words are close to each other. Table 1 (a),(b) illustrates our resource assumptions.

PD3 is described in Algorithm 1. We first train a classifier C (e.g., a neural network) on our labeled L1 data $\mathcal{L}_S$. Then we apply the trained model on the unlabeled x^S instances from $\mathcal{U}_S$, yielding pseudo-labeled dataset $\hat{\mathcal{D}}_S$. Next, we create another pseudo-labeled L2 data set $\hat{\hat{\mathcal{D}}}_T$ by projecting the label $\hat{y}^S$ of x^S in a pair $(x^S, x^T) \in \mathcal{U}^{S,T}$ to the instance x^T. We note that projection is trivial and 'loss-less' for sentence classification tasks because there is exactly one label for the whole sentence.

```
Algorithm 1: PD3
  Input: $\mathcal{L}_S, \mathcal{U}^{S,T}, C$: labeled L1 data and unlabeled
         L1-L2 translations, and a classifier $C$
  Output: $M_{S\circledast\hat{S}\circledast\hat{\hat{T}}}$: a model trained (using $C$) on
          $\mathcal{L}_S$ as well as pseudo-labeled data derived
          from $\mathcal{U}^{S,T}$
  $M_S \leftarrow \text{train}_C(\mathcal{L}_S)$;
  $\hat{Y}_S \leftarrow \text{predict}_{M_S}(\mathcal{U}_S)$ ; // $\hat{\mathcal{D}}_S = \{(x^S, \hat{y}^S)\}$
  $\hat{\hat{\mathcal{D}}}_T \leftarrow \{(x^T, \hat{y}^S) \,|\, (x^S, x^T) \in$
    $\mathcal{U}^{S,T}, (x^S, \hat{y}^S) \in \hat{\mathcal{D}}_S\}$;
  $M_{S\circledast\hat{S}\circledast\hat{\hat{T}}} \leftarrow \text{train}_C(\mathcal{L}_S \circledast \hat{\mathcal{D}}_S \circledast \hat{\hat{\mathcal{D}}}_T)$;
```

In contrast, for sequence tagging problems, projection typically requires word alignment information, which is an error prone process. This is the reason why we use a 'double hat' for $\hat{\hat{D}}_T$ to indicate that there may be two sources of noise: one from prediction and one from projection.

Finally, we combine our original dataset $\mathcal{L}_S$ with the two pseudo-labeled dataset $\hat{\mathcal{D}}_S$ and $\hat{\hat{\mathcal{D}}}_T$ and train our classifier C on it; after training, our goal in cross-lingual transfer is to apply the trained classifiers to L2 data.

We denote this combination operation by $\circledast$. A simple approach is to let $\circledast$ be the "merging" (or, concatenation) of both datasets (**PD3-merge**). In this variant of PD3, $\mathcal{L}_S$, $\hat{\mathcal{D}}_S$ and $\hat{\hat{\mathcal{D}}}_T$ are merged into one big dataset on which training takes place.

A more sophisticated approach is to let $\circledast$ represent a multi-task learning (MTL) scenario (Caruana, 1993; Søgaard and Goldberg, 2016) in which L1 and L2 instances represent one task each (**PD3-MTL**). Here, rather than merging $\mathcal{L}_S$, $\hat{\mathcal{D}}_S$ and $\hat{\hat{\mathcal{D}}}_T$, we treat source language datasets ($\mathcal{L}_S$ and $\hat{\mathcal{D}}_S$) as one task and target language datasets ($\hat{\hat{\mathcal{D}}}_T$) as another task, each having a dedicated output layer. This leads to a different network architecture than in PD3-merge, in which we now have two separate output layers (i.e., one for each language); this distinction is also illustrated in Figure 1 below. Thus, for each input instance, we predict two outputs (e.g., two ArgMin labels), one in the source language and one in the target language.[2]

The general idea behind MTL is to learn several

[1] Data and code to reproduce our experiments are available from https://github.com/UKPLab/emnlp2018-argmin-workshop-pd3.

[2] During training, we update parameters for the 'correct' task as well as for all shared weights. At test time, we only pick the output corresponding to the target language task, if we focus on cross-lingual transfer, or corresponding to the source language, if we focus on in-language evaluation.

Not cooking [...]	1
To sum up [...]	0
For example [...]	2
I will [...]	3
⋮	⋮

(a) Labeled L1 data $\mathcal{L}_S$

He said [...]	Er sagte [...]
A blue [...]	Ein blauer [...]
Very good!	Sehr gut!
How [...]	Wie [...]
⋮	⋮

(b) Unlabeled parallel data $\mathcal{U}^{S,T}$

	$\mathcal{L}_S$	$\hat{\mathcal{D}}_S$	$\hat{\hat{\mathcal{D}}}_T$
Direct Transfer	✓		
Projection			✓
PD3	✓	✓	✓
. . .			

(c) Resources used by approaches

Table 1: Illustration of resources used for PD3: (a) labeled source language data; (b) unlabeled parallel data; (c) comparison with Direct Transfer and annotation projection. Arrows indicate the information flow: we use $\mathcal{L}_S$ to label the source side of parallel data and then project to its target side. Note that both variants of PD3 (PD3-merge and PD3-MTL) use the same resources but utilize/combine them differently, as described in the text.

tasks jointly, in one architecture with shared parameters, so that generalized representations can be learned (in the hidden layers of a neural network) that benefit multiple tasks. In our case, the two tasks solve the same problem (e.g., ArgMin), but in different languages. A general advantage of MTL over merging arises when tasks have different output spaces, in which case merging may confuse a learner due to heterogeneous labels across the two tasks. We do not face this situation. However, in our context, an advantage of MTL over merge may still be that the MTL paradigm has more capacity because it has connecting weights between the task-specific output layers and the network's last (common) hidden layer. Further, MTL can accommodate task-specific losses, which can be used to, e.g., down-weight one of the two tasks, besides further conceptual differences (Caruana, 1993). In our situation, splitting original and pseudo-labeled datasets by languages, in MTL, may also better account for syntactic and semantic idiosyncrasies of individual languages than merge, where such distinctions are blurred.

Table 1 (c) compares the different resource assumptions of direct transfer, annotation projection, and PD3. Note that other selections of resources might be possible (e.g., 'PD2', using only $\mathcal{L}_S$ and $\hat{\hat{\mathcal{D}}}_T$, or even differently annotated L2 data). We discuss some of these in the supplementary material.

3 Data

Table 2 gives dataset statistics for our two tasks, which we describe in the following.

ArgMin Our focus task is ArgMin on the sentence-level: the task is to determine whether a sentence contains one of the argumentative constructs major claim, claim, premise, or else is non-argumentative (Peldszus and Stede, 2013; Stab and Gurevych, 2014). We use the latest version of an English student essay corpus (Stab and Gurevych, 2017), which has recently also been translated to German by student crowd-workers (Eger et al., 2018). We give four examples from the English ArgMin dataset in Table 3. The majority of all instances is labeled as premise (47%). We use 3,000 sentences of the original training split as our parallel corpus and only train on the remaining 2,086 sentences (this is the set $\mathcal{L}_S$). We additionally evaluate our approaches with parallel data from TED (Hermann and Blunsom, 2014), where we train on the full 5,086 sentences from the ArgMin training split. TED contains a collection of talks on science, education, and related fields, transcribed into written English and translated by crowd-workers into different languages. We take two sources of parallel data here because the domain of the parallel data intuitively has an influence on results in tasks such as argumentation mining. That is, while standard NLP tasks such as POS tagging are relatively stable across different domains, arguments may be very differently realized across different datasets (Daxenberger et al., 2017). Frequency aspects also play a role, since argumentation may be prominent in domains such as student essays or debate portal, but much less ubiquitous in, e.g., news articles.

POS Tagging We also include a standard NLP task, namely, POS tagging. We use subsets of the Universal Dependency Treebanks (Nivre et al., 2016) with English as L1 and German and French as L2s. For English, we select 800 random sentences from the corresponding English treebank as training data and 200 sentences as development

Task	Task type	$\lvert\mathcal{Y}\rvert$	Train-EN		Dev-EN		Test-DE		Test-FR	
			Sent.	Tokens	Sent.	Tokens	Sent.	Tokens	Sent.	Tokens
POS	Token-Level	18	800	13,292	200	3,174	799	12,512	1,478	35,766
AM	Sentence-Level	4	5,086	105,990	607	12,658	1,448	29,234	-	-

Table 2: Statistics for datasets used in this work. $\lvert\mathcal{Y}\rvert$ denotes the size of the label space.

data.[3] We evaluate the system that transfers from English to German or French on the original development data provided in the corresponding treebank splits. As our unlabeled parallel data, we use subsets of various sizes from the TED parallel corpus for English-French and English-German.

4 Experimental Setup

Sentence level network architecture: In our **sentence-level ArgMin** experiments, we use a convolutional neural network (CNN) with 1-max pooling to learn a representation of the input sentence and feed this representation into a softmax regression classifier.[4] We use 800 CNN filters with a window size of 3. For optimization, we use Adam with a learning rate of 0.001. Training sentences are processed in minibatches of size 16. We do not apply dropout or ℓ_2 regularization.

We report average macro F1 scores over 20 runs with different random initializations. For PD3-merge, we shuffle the merged data before training—i.e., mini-batches can contain $\mathcal{L}_S$, $\hat{\mathcal{D}}_S$, and $\hat{\hat{\mathcal{D}}}_T$ data. For PD3-MTL, we shuffle L1 and L2 data individually and during training we sample each mini-batch from either task according to its size. In the MTL setup, we share the CNN layer across tasks and use task-specific softmax regression layers.

Sequence tagging network architecture: For **token-level POS tagging**, we implement a bidirectional LSTM as in Ma and Hovy (2016) and Lample et al. (2016) with a CRF output layer. This is a state-of-the-art system for sequence tagging tasks such as POS and NER. Our model uses pretrained word embeddings and optionally concatenates these with a learned character-level representation. For all experiments, we use the same network topology: we use two hidden layers with 100 hidden units each, applying dropout on the hidden units and on the word embeddings. We use Adam as optimizer. Our network uses a CRF output layer rather than a softmax classifier to account for dependencies between successive labels.

In the MTL setup, we use the same architecture, but connect the last hidden layer to individual output layers, one for each task. Our MTL architecture extends the architecture of Søgaard and Goldberg (2016) by replacing the softmax output layer with a CRF output layer, and by including character-level word representations. The difference between MTL and single-task learning (STL) is illustrated in Figure 1. STL is a network with only one task, as in PD3-merge, direct transfer and standard projection.

We report average accuracy over five (or 10, in case of very little data) random weight matrix initializations. In the MTL setup, we choose a mini-batch randomly in each iteration (containing instances from only one of the tasks as in our sentence-level ArgMin experiments).

Cross-lingual Embeddings: For token-level experiments, we initially train 100-d BIVCD embeddings (Vulić and Moens, 2015) from Europarl (Koehn, 2005) (for EN-DE) and the UN corpus (Ziemski et al., 2016) (for EN-FR), respectively. For sentence-level experiments, we use 300-d BIVCD embeddings. This means that we initially assume that high-quality bilingual word embeddings are readily available for the two languages involved. At first sight, this appears a realistic assumption since high-quality bilingual embeddings can already be obtained with very little available *bilingual* data (Zhang et al., 2016; Artetxe et al., 2017). In low-resource settings, however, even little *monolingual* data is typically available for L2 and we address this setup subsequently.

Upper bound: For both ArgMin and POS, we report the in-language upper bound, i.e., when the model is trained and evaluated on L2. For this, we choose random L2 train sets of size $\lvert\mathcal{L}_S\rvert$.

Projection strategy for sequence tagging: We first word-align parallel data using fast-align (Dyer et al., 2013). When an L2 word is uniquely aligned to an L1 word, we assign it the L1 word's unique

[3] We choose only 800 sentences in order to keep overall computational costs of our experiments smaller. Note that 800 sentences yield an in-language performance of roughly 90%.

[4] An alternative would have been to directly work on sentence-level representations using cross-lingual sentence embeddings (Rücklé et al., 2018).

Not cooking fresh food will lead to lack of nutrition	Claim
To sum up, [...] the merits of animal experiments still outweigh the demerits	Major claim
For example, tourism makes up one third of Czech's economy	Premise
I will mention some basic reasoning as follows	O

Table 3: Simplified examples (EN) from our AM corpus, one for each of the four classes.

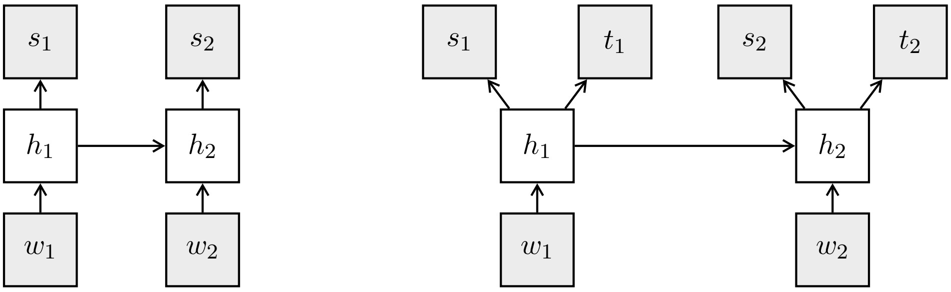

Figure 1: Sequence tagging STL vs. MTL with two tasks. For readability, character-level representations and CRF connections in the output layers are omitted. Bidirectional connections in the hidden layers are also missing. Here, w are the input words and s and t denote different tasks; h are the hidden layers.

label. When an L2 word is aligned to several L1 words, we randomly draw one of the aligned source labels. When an L2 word is not aligned to any L1 word, we draw a label randomly from its unique labels in the remainder of the corpus. Our projection strategy is standard, cf. Agic et al. (2016).

5 Experiments

5.1 Results

Detailed results for PD3-merge, PD3-MTL, standard projection, and direct transfer as a function of the available parallel data are given in Table 4 in the appendix. Condensed and averaged results (over DE and FR) are shown in Figure 2.

ArgMin Results are shown in Figure 2 (right), ranging over $\{50, 100, 500, 1000, 2000, 3000\}$ parallel sentences. PD3 is consistently more effective than projection and outperforms direct transfer with at least 100 parallel sentences. In particular, PD3-merge outperforms direct transfer already with 50 parallel sentences ($\sim$44% for PD3-merge vs. $\sim$39% for direct transfer) and quickly closes the gap towards the in-language upper-bound ($\sim$54% vs. $\sim$59% with 500 parallel sentences). PD3-MTL on the other hand only slightly (but consistently) improves upon projection. With an increased number of parallel sentences, we observe that all methods reach performances very close to the in-language upper bound.

POS Tagging Figure 2 (left) shows POS results, averaged across DE and FR, when transferring from English. Tagging accuracies are given as a function of the size of the available parallel data, ranging over $\{50, 100, 500, 1000, 5000\}$ parallel sentences. As for ArgMin, PD3 is consistently better than projection and improves upon direct transfer with more than 50 parallel sentences. As the number of parallel sentences increases, PD3-MTL, PD3-merge and standard projection become indistinguishable, indicating that it does not pay out anymore to use the more resource-intensive approach PD3. However, most importantly, with little parallel data, gains of PD3 over standard projection are substantial: for 50 parallel sentences performance values are roughly doubled ($\sim$30% accuracy for projection vs. >55% for PD3). For little available parallel data, PD3-MTL can also considerably improve upon PD3-merge. For example, with 100 parallel sentences, PD3-MTL achieves an accuracy of $\sim$65%, whereas PD3-merge achieves $\sim$60% and direct transfer achieves $\sim$61%.

5.2 Analysis

We now analyze several aspects of our approach, such as the errors it commits and the differences between PD3-MTL and PD3-merge, as well as whether we observe the same trends for high- and low-quality bilingual embeddings.

PD3-MTL vs. PD3-merge For POS, the better performance of PD3-MTL in some cases compared to PD3-merge may be due PD3-MTL having more parameters due to independent connection weights between the CRF classifier and the last hidden layer. Moreover, some authors have also argued that MTL is “fundamentally different” from simply adding auxiliary data (Bollmann and Søgaard, 2016). In contrast, for ArgMin, we observed that PD3-merge substantially outperforms PD3-MTL in many cases. We hypothesize that the reason is the model selection for PD3-MTL, which chooses the model with best performance on the dev portion of $\hat{\hat{\mathcal{D}}}_T$. Since the model trained on the small $\mathcal{L}_S$ train set tends to overpredict the majority class here, the label distribution on the parallel data differs substantially from that of the test data. The effects in PD3-merge are not as pronounced since it also contains parts of data with the true label distribution.

Direct Transfer vs. PD3 Direct transfer sometimes outperforms PD3 for very few available parallel sentences because PD3 uses noisy data in the form of projected labels, which are particularly unreliable when parallel data is scarce (see our error analysis below). This is not true for ArgMin, however, because projection is loss-less here, as remarked above. Accordingly, direct transfer never outperforms PD3 for ArgMin.

Domain shift of parallel data Using TED as parallel corpus in ArgMin rather than a held-out portion of the ArgMin dataset itself, we observe the following, see Figure 3 (top) and Table 4: (i) PD3-merge still outperforms all other methods; (ii) PD3-MTL more strongly outperforms projection; (iii) the in-language upper-bound is harder to reach. Overall, however, our curves follow a very similar trend as they do when parallel data comes from ArgMin itself, even though argumentation in TED is certainly much less pronounced than it is in student essays. This means that our approach appears robust to changes in domain of the parallel data even for domain-specific problems such as ArgMin, and can still outperform direct transfer in these cases. This is important since parallel data is generally sparse and most likely there is a substantial domain gap to the original L1 train data. The TED results are also interesting insofar as PD3-merge using 1K parallel sentences performs similarly as standard projection does using 100K.

Error Analysis For **POS**, the projection system that uses only 50 parallel sentences suffers not only from a tiny L2 training corpus (50 sentences, 783 tokens). Because the parallel corpus is tiny, getting high-quality alignments from fast-align on it is also more difficult because the aligner lacks statistical evidence. We checked alignment quality on 11 randomly chosen short translation pairs (both pairs shorter than 10 tokens) and on 3 long pairs (both longer than 20 tokens) for EN-DE. On the short pairs, 26% of the alignment decisions of fast-align were wrong. On the long pairs, 46% were wrong. In contrast, with 5000 parallel sentences error rates were considerably lower: 11% and 16%, respectively. Hence, projection uses a tiny corpus with considerable noise in the case of very small amount of parallel data, causing it to commit all kinds of errors (e.g., tagging verbs as numbers, etc.). In contrast, PD3 uses a larger and much cleaner amount of L1 data besides the tiny and noisy L2 corpus, which causes it to perform substantially better.

Direct transfer systems suffer mostly from two sources of noise: “syntactic shift” due to the L2 language having a different word order than the L1 counterpart on which they have been trained; “semantic shift” due to the test words being all OOV (this is analogous to monolingually replacing words by OOV synonyms). The latter effect may be understood as a “blurring” of the input. Accordingly, direct transfer easily confuses similar classes: for example, the EN→DE direct transfer system has a low F1-score on AUX (confusing auxiliary verbs with actual verbs) of 35% and on NOUN (confusing nouns with proper nouns) of 37%. Adding L2 data to the train set, as in PD3, quickly alleviates this: the F1-score on AUX for 100 parallel sentences is 37% and it is 62% for NOUN for PD3-merge. For 5000 parallel sentences, corresponding numbers are 56% and 76% respectively.

For **ArgMin** and tiny amounts of parallel data, projection predicts all classes but has a very strong tendency to predict the majority class ‘premise’. The reason is not that projected labels are noisy—in contrast, they are very good, because projection is error-free, as stated above. The problem is rather that the amount of training data for standard projection is tiny in this case (size of $\hat{\hat{\mathcal{D}}}_T$). PD3 in contrast trains on much more data and mimics the true distribution much better. Common errors for PD3 and direct transfer are confusing claims with major claims; these often have very similar surface realizations.

Low-resource shared representations In our main experiments, we assumed access to high qual-

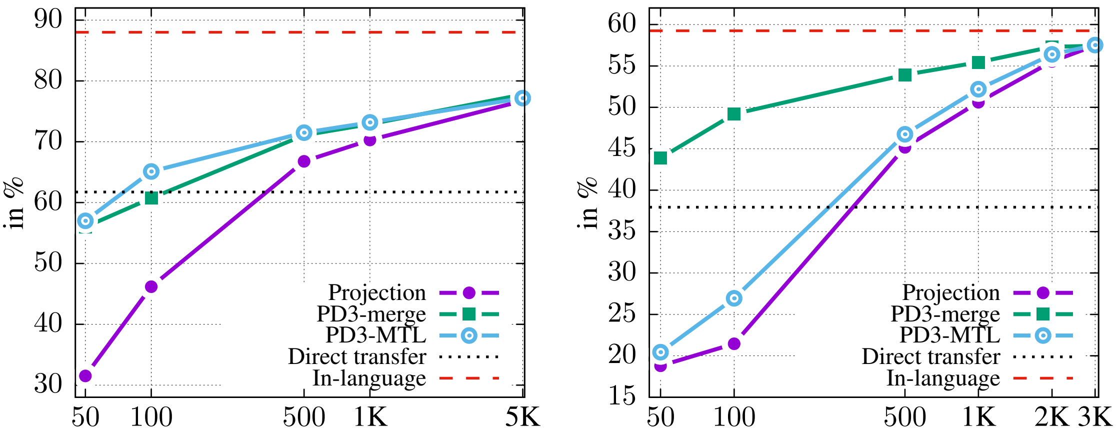

Figure 2: Left: POS accuracies in % as a function of available parallel sentences. Right: Sentence-level ArgMin F1 scores in % as a function of available parallel sentences.

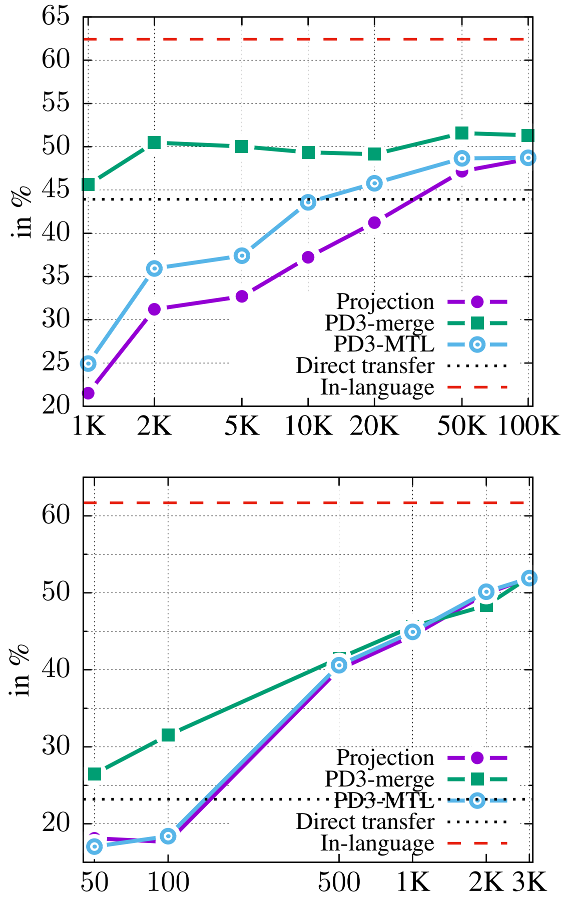

Figure 3: Top: Sentence-level ArgMin F1 scores in % as a function of available parallel sentences (sampled from the parallel TED corpus). Bottom: Sentence-level ArgMin F1 scores in % as a function of available parallel sentences (low-quality bilingual word embeddings).

ity bilingual word embeddings. This may not be justified when the L2 language is low-resource. Hence, we investigated performances when little monolingual data in L2 is available. That is, we limited monolingual data to only 30K sentences in L2. Given that we assumed 50-3000 parallel sentences for projections and given that monolingual data is typically much more plentiful than parallel data, we deemed 30K plausible. We report trends for the ArgMin sentence classification problem.

Hence, we trained a monolingual word2vec model on 30K DE sentences (randomly sampled from the German Wikipedia). For English, we trained a similar model on the whole of the English Wikipedia (since L1 is not low-resource). To induce a bilingual vector space, we then mapped English and German in a common space via the method of Artetxe et al. (2017). This approach iteratively expands a small seed lexicon of matched word pairs, thereby successively improving vector space alignment across two languages. It has been reported to induce good bilingual representations even when only common digits in the two languages or a few dictionary entries are available as initial seed lexicon. We induced a seed dictionary from our parallel sentences (ranging over $\{50, 100, 500, 1000, 2000, 3000\}$ pairs) using fast-align and then applied the technique of Artetxe et al. (2017). We subsequently re-ran PD3 and all other models with the resulting low-quality bilingual word embeddings. The results, for ArgMin, are shown in Figure 3 (bottom). As can be seen, direct transfer becomes considerably worse in this case, which is expected, since the embedding space is of much lower quality now. The performance drop is from 37% macro-F1 with high-quality embeddings to 23%. However, all trends stay the same, e.g., PD3-merge remains the top performer for the sentence-level experiments, followed by PD3-MTL and standard projection. A difference is that PD3-merge now becomes indistinguishable from standard projection for 1K parallel sentences already, rather than 2K as before.

In the extreme case when the bilingual vector space separates into two independent spaces, one for each language, then standard projection is at least as good as PD3, for all sizes of parallel data. This is because the L1 data cannot improve the L2 model since both operate on independent representations. However, it is likely that the added noise may then even confuse a PD3 system if it is not well-regularized.

We experimented with further reductions to 10K monolingual sentences in L2 and still saw a similar trend as in Figure 3 (bottom). Below 10K sentences, we found that, somewhat surprisingly, word2vec could not induce meaningful monolingual embedding spaces, though it is conceivable that other representation learning techniques, such as those based on co-occurrence matrices, would have performed better.

Comparison For POS, we note that our numbers are generally incomparable to other works because we use 800 monolingual sentences to train an English tagger from and (more importantly) treat the number of parallel sentences as a variable whose influence we investigate. Still, to give a reference: Täckström et al. (2013) report cross-lingual tagging accuracies of up to 90% for German and French as L2 using a constraint feature-based CRF. They use up to 5M parallel sentences and 500K size training data in L2, massively more than we use.

For ArgMin, we also have no direct comparisons, because we are the first, to our knowledge, to explore the student essay corpus of Stab and Gurevych (2017) on sentence- rather than token-level. Sentence-level annotation may be preferable because it is sometimes both conventional as well as difficult to decide which exact tokens should be part of an argument component (Persing and Ng, 2016). In terms of cross-language drop, Eger et al. (2018) report a similar drop of roughly 20pp when training an argumentation mining system on English and applying it to similarly annotated German data, for direct transfer. They close this gap using machine translation, while we close it under much milder assumptions using small amounts of parallel data and a more sophisticated transfer approach.

6 Related Work

Our work connects to different strands of research.

Multi-Task Learning MTL was shown to be particularly beneficial when tasks stand in a natural hierarchy and when they are syntactic in nature (Søgaard and Goldberg, 2016). Moreover, it has been claimed that further main benefits for MTL are observed when data for the main task is sparse, in which case the auxiliary tasks may act as regularizers that prevent overfitting (Ruder et al., 2017). The latter is the case for PD3-MTL with little available parallel data.

MTL has also been made use of for *supervised cross-lingual transfer* techniques (Cotterell and Heigold, 2017; Yang et al., 2017; Kim et al., 2017; Dinh et al., 2018). These assume small training sets in L2, and a system trained on them is regularized by a larger amount of training data in L1. In contrast to these, we assume no gold labels in L2 (*unsupervised transfer*), which necessitates a projection step. Our approach could also be combined with these supervised ones, by adding this small gold data to the three different datasets that we use in PD3.

Argumentation Mining ArgMin is a fast-growing field in NLP with applications in decision making and the legal domain (Palau and Moens, 2009) and can be solved on sentence-level (Daxenberger et al., 2017; Niculae et al., 2017; Stab et al., 2018) or token-level (Eger et al., 2017; Schulz et al., 2018). Cross-lingual ArgMin has recently attracted interest (Aker and Zhang, 2017; Eger et al., 2018). The proposed approaches mostly used machine translation, which is unavailable for the vast majority of the world's languages.

Low-resource transfer Low-resource language transfer has recently become very popular, e.g., when relying on only very few translation pairs for bilingual embedding space induction (Artetxe et al., 2017; Zhang et al., 2016) or in unsupervised machine translation using no parallel sources at all (Artetxe et al., 2018; Lample et al., 2018). Low-resource transfer (on a level of domains rather than languages) has also been considered in ArgMin (Schulz et al., 2018), assuming little annotated data in a new target domain due to annotation costs of ArgMin as a subjective high-level task.

7 Concluding Remarks

We combined direct transfer with annotation projection, addressing short-comings of both methods and combining their strengths. We saw consistent gains over either of the two methods in isolation, particularly in the small dataset scenario with 50-500 parallel sentences. This is arguably the most realistic scenario for a good portion of the world's languages, for which several dozens of parallel sentences are readily available e.g. from Bible translations (Christodoulopoulos and Steedman, 2015). We also note that while translating 50 sentences by hand may be as easy as labeling 50 sentences in L2, provided the problem requires no expert knowledge, parallel data serves many NLP problems, while the cost of labeling multiplies by the number of problems.

We also analyzed our approach under changes to external factors such as the bilingual embeddings and the domain of the parallel data, and found it to perform stable under such shifts, consistently outperforming the two baselines it is built upon in the setting of little available parallel sentences. This is particularly important for tasks such as ArgMin, for which it is inherently difficult to get domain specific parallel data, let alone for many languages.

Future work should consider further extensions: E.g., for cross-lingual approaches, it is also possible to select predictions on the source side of parallel data into the train sets only if the classifier's confidence exceeds a certain threshold, or to apply this process iteratively (Täckström, 2012). This can be immediately applied and extended to the PD3 approach. Another extension is to perform self-training on L2 data, which we briefly discuss in the supplementary material. Moreover, PD3 should also be applied in scenarios where L2 is a more distant language to English than considered here, or to setups where L1 is another language than English, although it is unlikely that the general trends we detected here would not persist under L1 and L2 variations. Further, while we did not observe consistent gains of PD3-MTL (sometimes considerable losses) over PD3-merge, we note that there are refinements of the MTL paradigm (e.g., Liu et al. (2017)) which might yield better results in our situation.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1816B (CEDIFOR) and 03VP02540 (ArgumenText) and by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1).

References

Zeljko Agic, 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 of Computational Linguistics (TACL)*, 4:301–312.

Ahmet Aker and Huangpan Zhang. 2017. Projection of argumentative corpora from source to target languages. In *Proceedings of the 4th Workshop on Argument Mining, ArgMining@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 67–72.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 451–462.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. *ICLR*.

Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bidirectional lstms and multi-task learning. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 131–139.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the 10th International Conference on Machine Learning (ICML 1993)*, pages 41–48.

Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

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 (EMNLP 2017)*, pages 759–770.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 2011 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2011)*, pages 600–609.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the Essence of a Claim? Cross-Domain Claim Identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2045–2056.

Erik-Lân Do Dinh, Steffen Eger, and Iryna Gurevych. 2018. Killing four birds with two stones: Multi-task learning for non-literal language detection. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1558–1569.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 644–648.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2018. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual Models for Compositional Distributed Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 58–68.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2822–2828.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *In Proceedings of the tenth Machine Translation Summit*, pages 79–86. AAMT.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 260–270.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. *ICLR*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1–10.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1064–1074.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 62–72.

Marie-Francine Moens. 2017. Argumentation mining: How can a machine acquire common sense and world knowledge? *Argument & Computation*. Accepted.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured svms and rnns. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Andreas Peldszus and Manfred Stede. 2013. From Argument Diagrams to Argumentation Mining in Texts: A Survey. *International Journal of Cognitive Informatics and Natural Intelligence*, 7(1):1–31.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated power mean word embeddings as universal cross-lingual sentence representations. *CoRR*, abs/1803.01400.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. In *arXiv preprint*.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–41. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 231–235.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 21–25.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1501–1510.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Oscar Täckström. 2012. Nudging the envelope of direct transfer methods for multilingual named entity recognition. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 55–63.

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

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 1061–1071.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 719–725.

Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *5th International Conference on Learning Representations (ICLR 2017)*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL 1995)*, pages 189–196.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research (HLT 2001)*, pages 1–8.

Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. 2016. Ten pairs to tag – multilingual pos tagging via coarse mapping between embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016))*, pages 1307–1317.

Micha Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

A Supplemental Material

In Table 4, we show detailed results across languages and tasks, as well as different transfer strategies. Below, we discuss another transfer strategy named L2-ST.

Further approaches When systems are trained on shared features as in direct transfer, then another approach for unsupervised cross-lingual transfer is self-training on L2 data (Täckström, 2012; Täckström et al., 2013). The idea is to train a system on labeled source language data $\mathcal{L}_S$, and then directly apply this trained system to the parallel target language data (this is possible because of the shared feature representation), rather than its source side and merge this newly obtained "self-labeled" dataset with $\mathcal{L}_S$.

We found this strategy, named L2-ST in Table 4 in the appendix, to perform substantially below our considered transfer strategies when there is a sufficient amount of L2 data available. Only with very little target L2 data (50 parallel sentences) did we observe some gains over PD3 in POS tagging. The reason is that for very little parallel data, alignment links are very noisy, as discussed above, so that the projected labels are of low quality. In this case, however, the best strategy is then to combine PD3 with self-training in L2, and thus to combine four datasets: two of them in L1 and two of them in L2. This strategy, which we dub PD4 in Table 4, outperforms L2-ST, but is worse than PD3 for high- and medium-sized parallel corpora. The reason is that the system trained on $\mathcal{L}_S$ is typically much better when applied to L1 data than when applied to L2—see our discussion on direct transfer—and thus the L2 predictions resulting from labeling the source side of parallel data and then projecting to L2 are better than those from directly predicting on L2, provided the projection step is sufficiently good.

This is also the reason why PD4-merge always underperforms PD3-merge for ArgMin—since projection is error-free for sentence level classification.

Task	**Projection**	**PD3-merge**	**PD3-MTL**	**L2-ST**	**PD4-merge**	**Direct Transfer**	**In-Language**
Parallel Sentences							(upper bound)
Token-level POS tagging with TED as parallel corpus (EN→DE)							
50	37.86	53.63	55.21	53.56	56.09	55.63	86.29
100	45.37	60.84	61.27	55.81	60.14		
500	67.07	70.16	70.18	57.60	64.68		
1,000	70.74	72.30	72.24	57.27	66.18		
5,000	76.04	77.22	76.55	56.28	66.67		
Token-level POS tagging with TED as parallel corpus (EN→FR)							
50	25.16	58.36	58.78	67.21	67.55	67.87	92.67
100	46.97	60.64	68.96	70.02	71.42		
500	66.49	72.00	72.79	70.34	73.81		
1,000	69.86	73.51	74.14	70.07	73.23		
5,000	77.41	78.29	77.81	68.92	74.37		
Sentence-level AM with 3K sentences of AM as parallel corpus (EN→DE)							
50	18.80	43.89	20.45	39.95	41.13	37.94	59.25
100	21.46	49.20	26.95	36.55	45.89		
500	45.18	53.93	46.75	39.18	49.53		
1,000	50.62	55.45	52.20	38.24	49.87		
2,000	55.55	57.32	56.39	38.47	50.29		
3,000	57.47	57.42	57.52	38.35	51.41		
Sentence-level AM with TED as parallel corpus (EN→DE)							
1,000	21.51	45.61	24.93	42.32	46.59	43.93	62.42
2,000	31.21	50.48	35.93	41.63	45.87		
5,000	32.71	50.03	37.40	43.57	46.85		
10,000	37.22	49.35	43.57	44.42	47.24		
20,000	41.23	49.13	45.78	45.08	48.02		
50,000	47.16	51.57	48.66	43.18	50.20		
100,000	48.58	51.32	48.72	45.05	50.53		

Table 4: Individual results for all tasks, languages, and number of parallel sentences. We report the accuracy for our token-level POS tagging experiments and F1 scores for our sentence-level AM experiments. L2-ST denotes cross-lingual transfer with self-training using L2 data as in (Täckström, 2012; Täckström et al., 2013). PD4-merge combines PD3 with self-training in L2.