Multi-Source Cross-Lingual Model Transfer: Learning What to Share

Xilun Chen[†*] **Ahmed Hassan Awadallah**[‡] **Hany Hassan**[‡]

Wei Wang[‡] **Claire Cardie**[†]

[†]Cornell University
Ithaca, NY
{xlchen,cardie}@cs.cornell.edu

[‡]Microsoft Research
Redmond, WA
{hassanam,hanyh,Wei.Wang}@microsoft.com

Abstract

Modern NLP applications have enjoyed a great boost utilizing neural networks models. Such deep neural models, however, are not applicable to most human languages due to the lack of annotated training data for various NLP tasks. Cross-lingual transfer learning (CLTL) is a viable method for building NLP models for a low-resource target language by leveraging labeled data from other (source) languages. In this work, we focus on the multilingual transfer setting where training data in multiple source languages is leveraged to further boost target language performance.

Unlike most existing methods that rely only on language-invariant features for CLTL, our approach coherently utilizes both language-invariant and language-specific features *at instance level*. Our model leverages adversarial networks to learn language-invariant features, and mixture-of-experts models to dynamically exploit the similarity between the target language and each individual source language[1]. This enables our model to learn effectively what to share between various languages in the multilingual setup. Moreover, when coupled with unsupervised multilingual embeddings, our model can operate in a **zero-resource** setting where neither *target language training data* nor *cross-lingual resources* are available. Our model achieves significant performance gains over prior art, as shown in an extensive set of experiments over multiple text classification and sequence tagging tasks including a large-scale industry dataset.

1 Introduction

Recent advances in deep learning enabled a wide variety of NLP models to achieve impressive performance, thanks in part to the availability of large-scale annotated datasets. However, such an advantage is not available to most of the world languages since many of them lack the the labeled data necessary for training deep neural nets for a variety of NLP tasks. As it is prohibitive to obtain training data for all languages of interest, *cross-lingual transfer learning* (CLTL) offers the possibility of learning models for a *target language* using annotated data from other languages (*source languages*) (Yarowsky et al., 2001). In this paper, we concentrate on the more challenging *unsupervised* CLTL setting, where *no* target language labeled data is used for training.[2]

Traditionally, most research on CLTL has been devoted to the standard **bilingual transfer** (BLTL) case where training data comes from a single source language. In practice, however, it is often the case that we have labeled data in a few languages, and would like to be able to utilize all of the data when transferring to other languages. Previous work (McDonald et al., 2011) indeed showed that transferring from multiple source languages could result in significant performance improvement. Therefore, in this work, we focus on the multi-source CLTL scenario, also known as **multilingual transfer learning** (MLTL), to further boost the target language performance.

One straightforward method employed in CLTL is weight sharing, namely directly applying the model trained on the source language to the target after mapping both languages to a common embedding space. As shown in previous work (Chen et al., 2016), however, the distributions of the hidden feature vectors of samples from different languages extracted by the same neural net remain divergent, and hence weight sharing is not sufficient for learning a language-invariant feature space that generalizes well across languages. As such, previ-

[*]Most work was done while the first author was an intern at Microsoft Research.

[1]The code is available at https://github.com/microsoft/Multilingual-Model-Transfer.

[2]In contrast, supervised CLTL assumes the availability of annotations in the target language.

ous work has explored using *language-adversarial training* (Chen et al., 2016; Kim et al., 2017) to extract features that are invariant with respect to the shift in language, using only (non-parallel) unlabeled texts from each language.

On the other hand, in the MLTL setting, where multiple source languages exist, language-adversarial training will only use, for model transfer, the features that are common among all source languages and the target, which may be too restrictive in many cases. For example, when transferring from English, Spanish and *Chinese* to German, language-adversarial training will retain only features that are invariant across all four languages, which can be too sparse to be informative. Furthermore, the fact that German is more similar to English than to Chinese is neglected because the transferred model is unable to utilize features that are shared only between English and German.

To address these shortcomings, we propose a new MLTL model that not only exploits language-invariant features, but also allows the target language to dynamically and selectively leverage language-specific features through a probabilistic attention-style mixture of experts mechanism (see §3). This allows our model to learn effectively what to share between various languages. Another contribution of this paper is that, when combined with the recent unsupervised cross-lingual word embeddings (Lample et al., 2018; Chen and Cardie, 2018b), our model is able to operate in a **zero-resource** setting where neither *task-specific target language annotations* nor *general-purpose cross-lingual resources* (e.g. parallel corpora or machine translation (MT) systems) are available. This is an advantage over many existing CLTL works, making our model more widely applicable to many lower-resource languages.

We evaluate our model on multiple MLTL tasks ranging from text classification to named entity recognition and semantic slot filling, including a real-world industry dataset. Our model beats all baseline models trained, like ours, without cross-lingual resources. More strikingly, in many cases, it can match or outperform state-of-the-art models that have access to strong cross-lingual supervision (e.g. commercial MT systems).

2 Related Work

The diversity of human languages is a critical challenge for natural language processing. In order to alleviate the need for obtaining annotated data for each task in each language, cross-lingual transfer learning (CLTL) has long been studied (Yarowsky et al., 2001; Bel et al., 2003, *inter alia*).

For *unsupervised* CLTL in particular, where no target language training data is available, most prior research investigates the **bilingual transfer** setting. Traditionally, research focuses on *resource-based* methods, where general-purpose cross-lingual resources such as MT systems or parallel corpora are utilized to replace task-specific annotated data (Wan, 2009; Prettenhofer and Stein, 2010). With the advent of deep learning, especially adversarial neural networks (Goodfellow et al., 2014; Ganin et al., 2016), progress has been made towards *model-based* CLTL methods. Chen et al. (2016) propose language-adversarial training that does not directly depend on parallel corpora, but instead only requires a set of bilingual word embeddings (BWEs).

On the other hand, the **multilingual transfer** setting, although less explored, has also been studied (McDonald et al., 2011; Naseem et al., 2012; Täckström et al., 2013; Hajmohammadi et al., 2014; Zhang and Barzilay, 2015; Guo et al., 2016), showing improved performance compared to using labeled data from one source language as in bilingual transfer.

Another important direction for CLTL is to learn cross-lingual word representations (Klementiev et al., 2012; Zou et al., 2013; Mikolov et al., 2013). Recently, there have been several notable work for learning fully unsupervised cross-lingual word embeddings, both for the bilingual (Zhang et al., 2017; Lample et al., 2018; Artetxe et al., 2018) and multilingual case (Chen and Cardie, 2018b). These efforts pave the road for performing CLTL without cross-lingual resources.

Finally, a related field to MLTL is multi-source domain adaptation (Mansour et al., 2009), where most prior work relies on the learning of domain-invariant features (Zhao et al., 2018; Chen and Cardie, 2018a). Ruder et al. (2019) propose a general framework for selective sharing between domains, but their method learns static weights at the *task level*, while our model can dynamically select what to share at the instance level. A very recent work (Guo et al., 2018) attempts to model the relation between the target domain and each source domain. Our model combines the strengths of these methods and is able to simul-

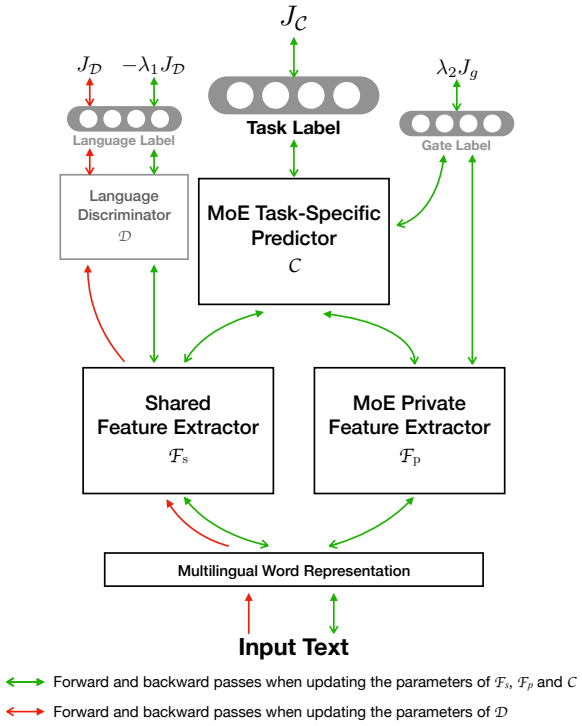

Figure 1: An overview of the MAN-MoE model.

taneously utilize both the domain-invariant and domain-specific features in a coherent way.

3 Model

One commonly adopted paradigm for neural cross-lingual transfer is the *shared-private* model (Bousmalis et al., 2016), where the features are divided into two parts: *shared* (language-invariant) features and *private* (language-specific) features. As mentioned before, the shared features are enforced to be language-invariant via language-adversarial training, by attempting to fool a language discriminator. Furthermore, Chen and Cardie (2018a) propose a generalized shared-private model for the multi-source setting, where a *multinomial adversarial network* (MAN) is adopted to extract common features shared by all source languages as well as the target. On the other hand, the private features are learned by separate feature extractors, one for each source language, capturing the remaining features outside the shared ones. During training, the labeled samples from a certain source language go through the corresponding private feature extractor for that particular language. At test time, there is no private feature extractor for the target language; only the shared features are used for cross-lingual transfer.

As mentioned in §1, using only the shared features for MLTL imposes an overly strong constraint and many useful features may be wiped out by adversarial training if they are shared only between the target language and a subset of source languages. Therefore, we propose to use a mixture-of-experts (MoE) model (Shazeer et al., 2017; Gu et al., 2018) to learn the private features. The idea is to have a set of language expert networks, one per source language, each responsible for learning language-specific features for that source language during training. However, instead of hard-switching between the experts, each sample uses a convex combination of all experts, dictated by an *expert gate*. Thus, at test time, the trained expert gate can decide the optimal expert weights for the unseen target language based on its similarity to the source languages.

Figure 1 shows an overview of our MAN-MoE model for multilingual model transfer. The boxes illustrate various components of the MAN-MoE model (§3.1), while the arrows depict the training flow (§3.2).

3.1 Model Architecture

Figure 1 portrays an abstract view of the MAN-MoE model with four components: the Multilingual Word Representation, the MAN Shared Feature Extractor $\mathcal{F}_s$ (together with the Language Discriminator $\mathcal{D}$), the MoE Private Feature Extractor $\mathcal{F}_p$, and finally the MoE Predictor $\mathcal{C}$. Based on the actual task (e.g. sequence tagging, text classification, sequence to sequence, etc.), different architectures may be adopted, as explained below.

Multilingual Word Representation embeds words from all languages into a single semantic space so that words with similar meanings are close to each other regardless of language. In this work, we mainly rely on the MUSE embeddings (Lample et al., 2018), which are trained in a fully unsupervised manner. We map all other languages into English to obtain a multilingual embedding space. However, in certain experiments, MUSE yields 0 accuracy on one or more language pairs (Søgaard et al., 2018), in which case the VecMap embeddings (Artetxe et al., 2017) are used. It uses *identical strings* as supervision, which does not require parallel corpus or human annotations. We further experiment with the recent unsupervised multilingual word embeddings (Chen and Cardie, 2018b), which gives improved performance (§4.2).

In addition, for tasks where morphological fea-

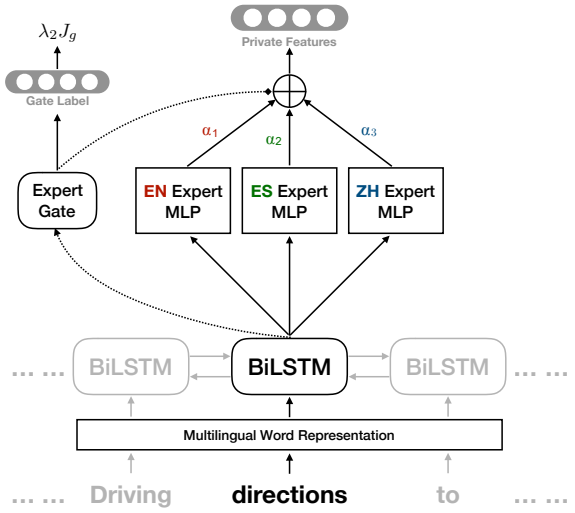

Figure 2: The MoE Private Feature Extractor $\mathcal{F}_p$ with three source languages: English (EN), Spanish (ES), and Chinese (ZH).

tures are important, one can add character-level word embeddings (Dos Santos and Zadrozny, 2014) that captures sub-word information. When character embeddings are used, we add a single CharCNN that is shared across all languages, and the final word representation is the concatenation of the word embedding and the char-level embedding. The CharCNN can then be trained end to end with the rest of the model.

MAN Shared Feature Extractor $\mathcal{F}_s$ is a multinomial adversarial network (Chen and Cardie, 2018a), which is an adversarial pair of a feature extractor (e.g. LSTM or CNN) and a *language discriminator* $\mathcal{D}$. $\mathcal{D}$ is a text classifier (Kim, 2014) that takes the shared features (extracted by $\mathcal{F}_s$) of an input sequence and predicts which language it comes from. On the other hand, $\mathcal{F}_s$ strives to fool $\mathcal{D}$ so that it cannot identify the language of a sample. The hypothesis is that if $\mathcal{D}$ cannot recognize the language of the input, the shared features then do not contain language information and are hence language-invariant. Note that $\mathcal{D}$ is trained only using unlabeled texts, and can therefore be trained on all languages including the target language.

MoE Private Feature Extractor $\mathcal{F}_p$ is a key difference from previous work, shown in Figure 2. The figure shows the Mixture-of-Experts (Shazeer et al., 2017) model with three source languages, English, Spanish, and Chinese. $\mathcal{F}_p$ has a shared BiLSTM at the bottom that extracts contextualized word representations for each token w in the input sentence. The LSTM hidden representation $\boldsymbol{h}_w$ is then fed into the MoE module, where each source language has a separate expert network (a MLP). In addition, the *expert gate* $\mathcal{G}$ is a linear transformation that takes $\boldsymbol{h}_w$ as input and outputs a softmax score α_i for each expert. The final private feature vector is a mixture of all expert outputs, dictated by the expert gate weights α.

During training, the expert gate is trained to predict the language of a sample using the gate loss J_g, where the expert gate output α is treated as the softmax probability of the predicted languages. In other words, the more accurate the language prediction is, the more the correct expert gets used. Therefore, J_g is used to encourage samples from a certain source language to use the correct expert, and each expert is hence learning language-specific features for that language. As the BiLSTM is exposed to all source languages during training, the trained expert gate will be able to examine the hidden representation of a token to predict the optimal expert weights α, even for **unseen** target languages at test time. For instance, if a German test sample is similar to the English training samples, the trained expert gate will predict a higher α for the English expert, resulting in a heavier use of it in the final feature vector. Therefore, even for the unforeseen target language (e.g. German), $\mathcal{F}_p$ is able to dynamically determine what knowledge to use from each individual source language at a token level.

MoE Task-Specific Predictor $\mathcal{C}$ is the final module that make predictions for the end task, and may take different forms depending on the task. For instance, for sequence tagging tasks, the shared and private features are first concatenated for each token, and then past through a MoE module similar to $\mathcal{F}_p$ (as shown in Figure 6 in the Appendix). It is straightforward to adapt $\mathcal{C}$ to work for other tasks. For example, for text classification, a pooling layer such as dot-product attention (Luong et al., 2015) is added at the bottom to fuse token-level features into a single sentence feature vector.

$\mathcal{C}$ first concatenates the shared and private features to form a single feature vector for each token. It then has another MoE module that outputs a softmax probability over all labels for each token. The idea is that it may be favorable to put different weights between the language-invariant and language-specific features for different target languages. Again consider the example of English, German, Spanish and Chinese. When transferring to Chinese from the other three, the source lan-

Algorithm 1 MAN-MoE Training

Require: labeled corpus $\mathbb{X}$; unlabeled corpus $\mathbb{U}$; Hyperpamameter $\lambda_1, \lambda_2 > 0$, $k \in \mathbb{N}$

repeat
  ▷ $\mathcal{D}$ iterations
  for $diter = 1$ to k **do**
    $l_{\mathcal{D}} = 0$
    for all $l \in \Delta$ **do** ▷ For all languages
      Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_l$
      $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$ ▷ Shared features
      $l_{\mathcal{D}} \mathrel{+}= L_{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); l)$ ▷ $\mathcal{D}$ loss
    Update $\mathcal{D}$ parameters using $\nabla l_{\mathcal{D}}$
  ▷ Main iteration
  $loss = 0$
  for all $l \in \mathcal{S}$ **do** ▷ For all source languages
    Sample a mini-batch $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{X}_l$
    $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$ ▷ Shared features
    $\boldsymbol{f}_p, \boldsymbol{g}_1 = \mathcal{F}_p(\boldsymbol{x})$ ▷ Private feat. & gate outputs
    $\hat{\boldsymbol{y}}, \boldsymbol{g}_2 = \mathcal{C}(\boldsymbol{f}_s, \boldsymbol{f}_p)$
    $loss \mathrel{+}= L_{\mathcal{C}}(\hat{\boldsymbol{y}}; \boldsymbol{y}) + \lambda_2(L_g(\boldsymbol{g}_1; l) + L_g(\boldsymbol{g}_2; l))$
  for all $l \in \Delta$ **do** ▷ For all languages
    Sample a mini-batch $\boldsymbol{x} \sim \mathbb{U}_l$
    $\boldsymbol{f}_s = \mathcal{F}_s(\boldsymbol{x})$ ▷ Shared features
    $loss \mathrel{+}= -\lambda_1 \cdot L_{\mathcal{D}}(\mathcal{D}(\boldsymbol{f}_s); l)$ ▷ Confuse $\mathcal{D}$
  Update $\mathcal{F}_s, \mathcal{F}_p, \mathcal{C}$ parameters using $\nabla loss$
until convergence

guages are similar to each other while all being rather distant from Chinese. Therefore, the adversarially learned shared features might be more important in this case. On the other hand, when transferring to German, which is much more similar to English than to Chinese, we might want to pay more attention to the MoE private features. Therefore, we adopt a MoE module in $\mathcal{C}$, which provides more flexibility than using a single MLP[3].

3.2 Model Training

Denote the set of all N source languages as $\mathcal{S}$, where $|\mathcal{S}| = N$. Denote the target language as $\mathcal{T}$, and let $\Delta = \mathcal{S} \cup \mathcal{T}$ be the set of all languages. Denote the annotated corpus for a source language $l \in \mathcal{S}$ as $\mathbb{X}_l$, where $(x, y) \sim \mathbb{X}_l$ is a sample drawn from $\mathbb{X}_l$. In addition, unlabeled data is required for all languages to facilitate the MAN training. We hence denote as $\mathbb{U}_{l'}$ the unlabeled texts from a language $l' \in \Delta$.

The overall training flow of variant components is illustrated in Figure 1, while the training algorithm is depicted in Algorithm 1. Similar to MAN, there are two separate optimizers to train MAN-MoE, one updating the parameters of $\mathcal{D}$ (red arrows), while the other updating the parameters of all other modules (green arrows). In Algorithm 1, $L_{\mathcal{C}}$, $L_{\mathcal{D}}$ and L_g are the loss functions for the predictor $\mathcal{C}$, the language discriminator $\mathcal{D}$, and the expert gates in $\mathcal{F}_p$ and $\mathcal{C}$, respectively.

In practice, we adopt the NLL loss for $L_{\mathcal{C}}$ for text classification, and token-level NLL loss for sequence tagging:

$$L^{NLL}(\hat{y}; y) = -\log P(\hat{y} = y) \quad (1)$$

$$L^{T\text{-}NLL}(\hat{\boldsymbol{y}}; \boldsymbol{y}) = -\log P(\hat{\boldsymbol{y}} = \boldsymbol{y}) = -\sum_i \log P(\hat{y}_i = y_i) \quad (2)$$

where y is a scalar class label, and $\boldsymbol{y}$ is a vector of token labels. $L_{\mathcal{C}}$ is hence interpreted as the *negative log-likelihood* of predicting the correct task label. Similarly, $\mathcal{D}$ adopts the NLL loss in (1) for predicting the correct language of a sample. Finally, the expert gates $\mathcal{G}$ use token-level NLL loss in (2), which translates to the negative log-likelihood of using the correct language expert for each token in a sample.

Therefore, the objectives that $\mathcal{C}$, $\mathcal{D}$ and $\mathcal{G}$ minimize are, respectively:

$$J_{\mathcal{C}} = \sum_{l \in \mathcal{S}} \mathop{\mathbb{E}}_{(x,y) \in \mathbb{X}_l} \left[L_{\mathcal{C}} \left(\mathcal{C}(\mathcal{F}_s(x), \mathcal{F}_p(x)); y \right) \right] \quad (3)$$

$$J_{\mathcal{D}} = \sum_{l \in \Delta} \mathop{\mathbb{E}}_{x \in \mathbb{U}_l} \left[L_{\mathcal{D}}(\mathcal{D}(\mathcal{F}_s(x)); l) \right] \quad (4)$$

$$J_{\mathcal{G}} = \sum_{l \in \mathcal{S}} \mathop{\mathbb{E}}_{x \in \mathbb{X}_l} \left[\sum_{w \in x} L_{\mathcal{G}}(\mathcal{G}(h_w); l) \right] \quad (5)$$

where $\boldsymbol{h}_w$ in (5) is the BiLSTM hidden representation in $\mathcal{F}_p$ as shown in Figure 2. In addition, note that $\mathcal{D}$ is trained using unlabeled corpora over all languages (Δ), while the training of $\mathcal{F}_p$ and $\mathcal{C}$ (and hence $\mathcal{G}$) only take place on source languages ($\mathcal{S}$). Finally, the overall objective function is:

$$J = J_{\mathcal{C}} - \lambda_1 J_{\mathcal{D}} + \lambda_2 \left(J_{\mathcal{G}}^{(1)} + J_{\mathcal{G}}^{(2)} \right) \quad (6)$$

where $J_{\mathcal{G}}^{(1)}$ and $J_{\mathcal{G}}^{(2)}$ are the two expert gates in $\mathcal{F}_p$ and $\mathcal{C}$, respectively. More implementation details can be found in Appendix B.

4 Experiments

In this section, we present an extensive set of experiments across three datasets. The first experiment is on a real-world multilingual slot filling (sequence tagging) dataset, where the data is used in a commercial personal virtual assistant. In addition, we conduct experiments on two public

[3] We also experimented with an attention mechanism between the shared and private features, or a gating mechanism to modulate each feature channel, but got sub-optimal results.

	English			German			Spanish			Chinese			
Domain	#Train	#Dev	#Test	#Train	#Dev	#Test	#Train	#Dev	#Test	#Train	#Dev	#Test	#Slot
Navigation	311045	23480	36625	13356	1599	2014	13862	1497	1986	7472	1114	1173	8
Calendar	64010	5946	8260	8261	1084	1366	6706	926	1081	2056	309	390	4
Files	30339	2058	5355	3005	451	480	6082	843	970	1289	256	215	5

Domain	Examples
Navigation	$[Driving]_{transportation\_type}$ directions to $[Walmart]_{place\_name}$ in $[New\ York]_{location}$.
Calendar	Add $[school\ meeting]_{title}$ to my calendar on $[Monday]_{start\_date}$ at $[noon]_{start\_time}$.
Files	Search for $[notes]_{data\_type}$ with $[grocery\ list]_{keyword}$.

Table 1: Statistics for the Multilingual Semantic Slot Filling dataset with examples from each domain.

academic datasets, namely the CoNLL multilingual named entity recognition (sequence tagging) dataset (Sang, 2002; Sang and Meulder, 2003), and the multilingual Amazon reviews (text classification) dataset (Prettenhofer and Stein, 2010).

4.1 Cross-Lingual Semantic Slot Filling

As shown in Table 1, we collect data for four languages: English, German, Spanish, and Chinese, over three domains: Navigation, Calendar, and Files. Each domain has a set of pre-determined slots (the slots are the same across languages), and the user utterances in each language and domain are annotated by crowd workers with the correct slots (see the examples in Table 1). We employ the standard BIO tagging scheme to formulate the slot filling problem as a sequence tagging task.

For each domain and language, the data is divided into a training, a validation, and a test set, with the number of samples in each split shown in Table 1. In our experiments, we treat each domain as a separate experiment, and consider each of German, Spanish and Chinese as the target language while the remaining three being source languages, which results in a total of 9 experiments.

4.1.1 Results

In Table 2, we report the performance of MAN-MoE compared to a number of baseline systems. All systems adopt the same base architecture, which is a multi-layer BiLSTM sequence tagger (İrsoy and Cardie, 2014) with a token-level MLP on top (no CRFs were used).

MT baselines employ machine translation (MT) for cross-lingual transfer. In particular, the *train-on-trans(lation)* method translates the entire English training set into each target language which are in turn used to train a supervised system on the target language. On the other hand, the *test-on-trans(lation)* method trains an English sequence tagger, and utilizes MT to translate the test set of each target language into English in order to make predictions. In this work, we adopt the Microsoft Translator[4], a strong commercial MT system. Note that for a MT system to work for sequence tagging tasks, *word alignment* information must be available, in order to project word-level annotations across languages. This rules out many MT systems such as Google Translate since they do not provide word alignment information through their APIs.

BWE baselines rely on Bilingual Word Embeddings (BWEs) and weight sharing for CLTL. Namely, the sequence tagger trained on the source language(s) are directly applied to the target language, in hopes that the BWEs could bridge the language gap. This simple method has been shown to yield strong results in recent work (Upadhyay et al., 2018). The MUSE (Lample et al., 2018) BWEs are used by all systems in this experiment. *1-to-1* indicates that we are only transferring from English, while *3-to-1* means the training data from all other three languages are leveraged.[5]

The final baseline is the MAN model (Chen and Cardie, 2018a), presented before our MAN-MoE approach. As shown in Table 2, MAN-MoE substantially outperforms all baseline systems that do not employ cross-lingual supervision on almost all domains and languages. Another interesting observation is that MAN performs strongly on Chinese while being much worse on German and Spanish compared to the BWE baseline. This corroborates our hypothesis that MAN only leverages features that are invariant across *all* languages for CLTL, and it learns such features better than weight sharing. Therefore, when transferring to German or Spanish, which is similar to a subset of source languages, the performance of

[4] https://azure.microsoft.com/en-us/services/cognitive-services/translator-text-api/

[5] MAN and MAN-MoE results are always 3-to-1.

Domain	German				Spanish				Chinese			
	Navi.	Cal.	Files	*avg.*	Navi.	Cal.	Files	*avg.*	Navi.	Cal.	Files	*avg.*
Methods with cross-lingual resources												
MT (train-on-trans.)	59.95	63.53	38.68	54.05	64.37	**59.93**	67.55	**63.95**	**60.56**	**66.49**	61.01	**62.69**
MT (test-on-trans.)	54.49	51.74	55.87	54.03	52.13	58.10	55.00	55.08	54.23	22.71	**64.01**	46.98
Methods without cross-lingual resources												
BWE (1-to-1)	57.53	58.28	35.73	50.51	62.54	44.44	57.56	54.85	17.62	22.48	21.32	20.47
BWE (3-to-1)	61.03	67.66	51.30	60.00	63.74	45.10	64.47	57.77	20.91	13.70	28.47	21.03
MAN	59.07	60.24	39.35	52.89	58.86	37.90	46.75	47.84	34.45	13.53	40.63	29.54
MAN-MoE	**62.73**	**75.13**	**59.19**	**65.68**	**66.57**	50.21	**70.91**	62.56	34.18	29.36	41.70	35.08

Table 2: F1 scores on the Multilingual Semantic Slot Filling dataset. The highest performance is in bold; the highest performance within method group (with vs. without cross-lingual resources) is underlined (*sic passim*).

Domain	German				Spanish				Chinese			
	Navi.	Cal.	Files	*avg*	Navi.	Cal.	Files	*avg*	Navi.	Cal.	Files	*avg*
MAN-MoE	62.73	75.13	**59.19**	**65.68**	**66.57**	**50.21**	**70.91**	**62.56**	34.18	**29.36**	41.70	**35.08**
- $\mathcal{C}$ MoE	**63.42**	**76.68**	55.68	65.26	65.50	47.51	69.67	60.89	27.71	21.75	**41.77**	30.41
- $\mathcal{F}_p$ MoE	58.33	48.85	37.35	48.18	58.99	36.67	48.39	48.02	**39.61**	14.64	38.08	30.78
- both MoE	59.07	60.24	39.35	52.89	58.86	37.90	46.75	47.84	34.45	13.53	40.63	29.54
- MAN	60.64	67.69	55.10	61.14	65.38	46.71	68.25	60.11	18.43	10.82	28.90	19.38

Table 3: Ablation (w.r.t. MAN-MoE) results on the Multilingual Semantic Slot Filling dataset.

MAN degrades significantly. On the other hand, when Chinese serves as the target language, where all source languages are rather distant from it, MAN has its merit in extracting language-invariant features that could generalize to Chinese. With MAN-MoE, however, this trade-off between close and distant language pairs is well addressed by the combination of MAN and MoE. By utilizing both language-invariant and language-specific features for transfer, MAN-MoE outperforms all cross-lingually unsupervised baselines on all languages.

Furthermore, even when compared with the MT baseline, which has access to hundreds of millions of parallel sentences, MAN-MoE performs competitively on German and Spanish. It even significantly beats both MT systems on German as MT sometimes fails to provide accurate word alignment for German. On Chinese, where the unsupervised BWEs are much less accurate (BWE baselines only achieve 20% F1), MAN-MoE is able to greatly improve over the BWE and MAN baselines and shows promising results for zero-resource CLTL even between distant language pairs.

4.1.2 Feature Ablation

In this section, we take a closer look at the various modules of MAN-MoE and their impacts on performance (Table 3). When the MoE in $\mathcal{C}$ is removed, moderate decrease is observed on all languages. The performance degrades the most on Chinese, suggesting that using a single MLP in $\mathcal{C}$ is not ideal when the target language is not similar to the sources. When removing the private MoE, the MoE in $\mathcal{C}$ no longer makes much sense as $\mathcal{C}$ only has access to the shared features, and the performance is even slightly worse than removing both MoEs. With both MoE modules removed, it reduces to the MAN model, and we see a significant drop on German and Spanish. Finally, when removing MAN while keeping MoE, where the shared features are simply learned via weight-sharing, we see a slight drop on German and Spanish, but a rather great one on Chinese. The ablation results support our hypotheses and validate the merit of MAN-MoE.

4.2 Cross-Lingual Named Entity Recognition

In this section, we present experiments on the CoNLL 2002 & 2003 multilingual named entity recognition (NER) dataset (Sang, 2002; Sang and Meulder, 2003), with four languages: English, German, Spanish and Dutch. The task is also formulated as a sequence tagging problem, with four types of tags: PER, LOC, ORG, and MISC.

The results are summarized in Table 4. We observe that using only word embeddings does not yield satisfactory results, since the out-of-vocabulary problem is rather severe, and morphological features such as capitalization is crucial for NER. We hence add character-level word embeddings for this task (§3.1) to capture subword fea-

Target Language	de	es	nl	avg
Methods with cross-lingual resources				
Täckström et al. (2012)	40.4	59.3	58.4	52.7
Nothman et al. (2013)	55.8	61.0	64.0	60.3
Tsai et al. (2016)	48.1	60.6	61.6	56.8
Ni et al. (2017)	**58.5**	65.1	65.4	63.0
Mayhew et al. (2017)	57.5	66.0	64.5	62.3
Methods without cross-lingual resources				
MAN-MoE	55.1	59.5	61.8	58.8
BWE+CharCNN (1-to-1)	51.5	61.0	67.3	60.0
BWE+CharCNN (3-to-1)	55.8	70.4	69.8	65.3
Xie et al. (2018)*	56.9	71.0	71.3	66.4
MAN-MoE+CharCNN	56.7	71.0	70.9	66.2
MAN-MoE+CharCNN+UMWE	56.0	**73.5**	**72.4**	**67.3**

* Contemporaneous work

Table 4: F1 scores for the CoNLL NER dataset on German (de), Spanish (es) and Dutch (nl).

tures and alleviate the OOV problem. For German, however, all nouns are capitalized, and the capitalization features learned on the other three languages would lead to poor results. Therefore, for German only, we lowercase all characters in systems that adopt CharCNN.

Table 4 also shows the performance of several state-of-the-art models in the literature[6]. Note that most of these systems are specifically designed for the NER task, and exploit many task-specific resources, such as multilingual gazetteers, or metadata in Freebase or Wikipedia (such as entity categories). Among these, Täckström et al. (2012) rely on parallel corpora to learn cross-lingual word clusters that serve as features. Nothman et al. (2013); Tsai et al. (2016) both leverage information in external knowledge bases such as Wikipedia to learn useful features for cross-lingual NER. Ni et al. (2017) employ noisy parallel corpora (aligned sentence pairs, but not always translations) and bilingual dictionaries (5k words for each language pair) for model transfer. They further add external features such as entity types learned from Wikipedia for improved performance. Finally, Mayhew et al. (2017) propose a multi-source framework that utilizes large cross-lingual lexica. Despite using none of these resources, general or task-specific, MAN-MoE nonetheless outperforms all these methods. The only exception is German, where task-specific resources remain helpful due to its unique capitalization rules and high OOV rate.

[6]We also experimented with the MT baselines, but it often failed to produce word alignment, resulting in many empty predictions. The MT baselines attain only a F1 score of ∼30%, and were thus excluded for comparison.

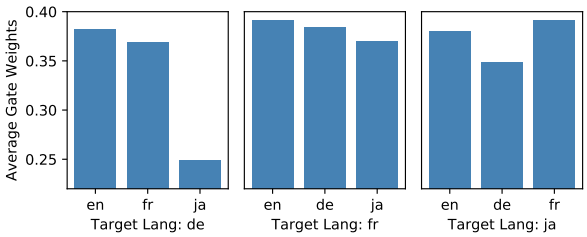

Figure 3: Average expert gate weights aggregated on a language level for the *Amazon Reviews* dataset.

In a contemporaneous work by (Xie et al., 2018), they propose a cross-lingual NER model using Bi-LSTM-CRF that achieves similar performance compared to MAN-MoE+CharCNN. However, our architecture is not specialized to the NER task, and we did not add task-specific modules such as a CRF decoding layer, etc.

Last but not least, we replace the MUSE embeddings with the recently proposed unsupervised multilingual word embeddings (Chen and Cardie, 2018b), which further boosts the performance, achieving a new state-of-the-art performance as shown in Table 4 (last row).

4.3 Cross-Lingual Text Classification on Amazon Reviews

Finally, we report results on a multilingual text classification dataset (Prettenhofer and Stein, 2010). The dataset is a binary classification dataset where each review is classified into positive or negative sentiment. It has four languages: English, German, French and Japanese.

As shown in Table 5, MT-BOW uses machine translation to translate the bag of words of a target sentence into the source language, while CL-SCL learns a cross-lingual feature space via structural correspondence learning (Prettenhofer and Stein, 2010). CR-RL (Xiao and Guo, 2013) learns bilingual word representations where part of the word vector is shared among languages. Bi-PV (Pham et al., 2015) extracts bilingual paragraph vector by sharing the representation between parallel documents. UMM (Xu and Wan, 2017) is a multilingual framework that could utilize parallel corpora between multiple language pairs, and pivot as needed when direct bitexts are not available for a specific source-target pair. Finally CLDFA (Xu and Yang, 2017) proposes cross-lingual distillation on parallel corpora for CLTL. Unlike other works listed, however, they adopt a task-specific parallel corpus (translated Amazon reviews) that are difficult to obtain in practice, making the num-

	German				French				Japanese			
Domain	books	dvd	music	*avg*	books	dvd	music	*avg*	books	dvd	music	*avg*
Methods with general-purpose cross-lingual resources												
MT-BOW[1]	79.68	77.92	77.22	78.27	80.76	78.83	75.78	78.46	70.22	71.30	72.02	71.18
CL-SCL[1]	79.50	76.92	77.79	78.07	78.49	78.80	77.92	78.40	73.09	71.07	75.11	73.09
CR-RL[2]	79.89	77.14	77.27	78.10	78.25	74.83	78.71	77.26	71.11	73.12	74.38	72.87
Bi-PV[3]	79.51	78.60	**82.45**	80.19	**84.25**	79.60	80.09	81.31	71.75	75.40	75.45	74.20
UMM[4]	81.65	81.27	81.32	81.41	80.27	80.27	79.41	79.98	71.23	72.55	75.38	73.05
Methods with task-specific cross-lingual resources												
CLDFA[5]	**83.95**	**83.14**	79.02	**82.04**	83.37	82.56	**83.31**	**83.08**	**77.36**	**80.52**	**76.46**	**78.11**
Methods without cross-lingual resources												
BWE (1-to-1)	76.00	76.30	73.50	75.27	77.80	78.60	78.10	78.17	55.93	57.55	54.35	55.94
BWE (3-to-1)	78.35	77.45	76.70	77.50	77.95	79.25	79.95	79.05	54.78	54.20	51.30	53.43
MAN-MoE	82.40	78.80	77.15	79.45	81.10	**84.25**	80.90	82.08	62.78	69.10	72.60	68.16

[1] Prettenhofer and Stein (2010) [2] Xiao and Guo (2013) [3] Pham et al. (2015)
[4] Xu and Wan (2017) [5] Xu and Yang (2017)

Table 5: Results for the Multilingual Amazon Reviews dataset. Numbers indicate binary classification accuracy. VecMap embeddings (Artetxe et al., 2017) are used for this experiment as MUSE training fails on Japanese (§3.1).

bers not directly comparable to others.

Among these methods, UMM is the only one that does not require direct parallel corpus between all source-target pairs. It can instead utilize pivot languages (e.g. English) to connect multiple languages. MAN-MoE, however, takes another giant leap forward to completely remove the necessity of parallel corpora while achieving similar results on German and French compared to UMM. On Japanese, the performance of MAN-MoE is again limited by the quality of BWEs. (BWE baselines are merely better than randomness.) Nevertheless, MAN-MoE remains highly effective and the performance is only a few points below most SoTA methods with cross-lingual supervision.

For a better understanding of the model behavior, Figure 3 visualizes the expert weights when transferring to different languages, which corroborates our model hypothesis and the findings in §4.1.2 (see Appendix A for more details).

5 Conclusion

In this paper, we propose MAN-MoE, a multilingual model transfer approach that exploits both language-invariant (shared) features and language-specific (private) features, which departs from most previous models that can only make use of shared features. Following earlier work, the shared features are learned via language-adversarial training (Chen et al., 2016). On the other hand, the private features are extracted by a mixture-of-experts (MoE) module, which is able to dynamically capture the relation between the target language and each source language on a token level. This is extremely helpful when the target language is similar to a subset of source languages, in which case traditional models that solely rely on shared features would perform poorly. Furthermore, MAN-MoE is a purely model-based transfer method, which does not require parallel data for training, enabling fully zero-resource MLTL when combined with unsupervised cross-lingual word embeddings. This makes MAN-MoE more widely applicable to lower-resourced languages.

Our claim is supported by a wide range of experiments over multiple text classification and sequence tagging tasks, including a large-scale industry dataset. MAN-MoE significantly outperforms all cross-lingually unsupervised baselines regardless of task or language. Furthermore, even considering methods with strong cross-lingual supervision, MAN-MoE is able to match or outperform these models on closer language pairs. When transferring to distant languages such as Chinese or Japanese (from European languages), where the quality of cross-lingual word embeddings are unsatisfactory, MAN-MoE remains highly effective and substantially mitigates the performance gap introduced by cross-lingual supervision.

For future work, we plan to apply MAN-MoE to more challenging languages for tasks such as syntactic parsing, where multilingual data exists (Nivre et al., 2017). Furthermore, we would like to experiment with multilingual contextualized embeddings such as the Multilingual BERT (Devlin et al., 2018).

References

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 (Volume 1: Long Papers)*, pages 451–462. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798. Association for Computational Linguistics.

Nuria Bel, Cornelis H. A. Koster, and Marta Villegas. 2003. Cross-lingual text categorization. In *Research and Advanced Technology for Digital Libraries*, pages 126–139, Berlin, Heidelberg. Springer Berlin Heidelberg.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems 29*, pages 343–351. Curran Associates, Inc.

Xilun Chen and Claire Cardie. 2018a. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240. Association for Computational Linguistics.

Xilun Chen and Claire Cardie. 2018b. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270, Brussels, Belgium. Association for Computational Linguistics.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *ArXiv e-prints*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Cícero Nogueira Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354. Association for Computational Linguistics.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *AAAI Conference on Artificial Intelligence*.

Jiang Guo, Darsh Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Sadegh Hajmohammadi, Roliana Ibrahim, Ali Selamat, and Alireza Yousefpour. 2014. Combination of multi-view multi-source language classifiers for cross-lingual sentiment classification. In *Intelligent Information and Database Systems*, pages 21–30. Springer International Publishing.

Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 720–728.

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*, pages 2832–2838. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Herv Jgou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems 21*, pages 1041–1048. Curran Associates, Inc.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545. Association for Computational Linguistics.

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

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 629–637, Jeju Island, Korea. Association for Computational Linguistics.

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Aljoscha Burchardt, Marie Candito, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Silvie Cinková, Çağrı Çöltekin, Miriam Connor, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Marhaba Eli, Ali Elkahky, Tomaž Erjavec, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Pinkey Nainwani, Anna Nedoluzhko, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Livy Real, Siva Reddy, Georg Rehm, Larissa Rinaldi, Laura Rituma, Rudolf Rosa, Davide Rovati, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Antonio Stella, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zhuoran Yu, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0 CoNLL 2017 shared task development and test data.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning mul-

tilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Hieu Pham, Minh-Thang Luong, and Christopher Manning. 2015. Learning distributed representations for multilingual text sequences. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 88–94, Denver, Colorado. Association for Computational Linguistics.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127. Association for Computational Linguistics.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *AAAI Conference on Artificial Intelligence*.

Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

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*, pages 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487. Association for Computational Linguistics.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 219–228. Association for Computational Linguistics.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243. Association for Computational Linguistics.

Min Xiao and Yuhong Guo. 2013. Semi-supervised representation learning for cross-lingual text classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1465–1475. Association for Computational Linguistics.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379. Association for Computational Linguistics.

Kui Xu and Xiaojun Wan. 2017. Towards a universal sentiment classifier in multiple languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 511–520, Copenhagen, Denmark. Association for Computational Linguistics.

Ruochen Xu and Yiming Yang. 2017. Cross-lingual distillation for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425. Association for Computational Linguistics.

D. Yarowsky, G. Ngai, and R. 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*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,

pages 1959–1970, Vancouver, Canada. Association for Computational Linguistics.

Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, Joao P Costeira, and Geoffrey J Gordon. 2018. Adversarial multiple source domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8568–8579. Curran Associates, Inc.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics.

Appendix A Visualization of Expert Gate Weights

In Figure 4 and 5, we visualize the average expert gate weights for each of the three target languages in the Amazon and CoNLL datasets, respectively. For each sample, we first compute a sentence-level aggregation by averaging over the expert gate weights of all its tokens. These sentence-level expert gate weights are then further averaged across all samples in the validation set, which forms a final language-level average expert gate weight for each target language. For the Amazon dataset, we take the combination of all three domains (books, dvd, music).

The visualization further collaborates with our hypothesis that our model makes informed decisions when selecting what features to share to the target language. On the Amazon dataset, it can be seen that when transferring to German or French (from the remaining three), the Japanese expert is less utilized compared to the European languages. On the other hand, it is interesting that when transferring to Japanese, the French and English experts are used more than the German one, and the exact reason remains to be investigated. However, this phenomenon might be of less significance since the private features may not play a very important role when transferring to Japanese as the model is probably focusing more on the shared features, according to the ablation study in Section 4.1.2.

In addition, on the CoNLL dataset, we observe that when transferring to German, the experts from the two more similar lanaguages, English and Dutch, are favored over the Spanish one. Similarly, when transferring to Dutch, the highly relevant German expert is heavily used, and the Spanish expert is barely used at all. Interestingly, when transferring to Spanish, the model also shows a skewed pattern in terms of expert usage, and prefers the German expert over the other two.

Appendix B Implementation Details

In all experiments, Adam (Kingma and Ba, 2015) is used for both optimizers (main optimizer and $\mathcal{D}$ optimizer), with learning rate 0.001 and weight decay 10^{-8}. Batch size is 64 for the slot filling experiment and 16 for the NER and Amazon Reviews experiments, which is selected mainly due to memory concerns. CharCNN increases the GPU memory usage and NER hence could only use a batch size of 16 to fit in 12GB of GPU memory. The Amazon experiment does not employ character embeddings but the documents are much longer, and thus also using a smaller batch size. All embeddings are fixed during training. Dropout (Srivastava et al., 2014) with $p = 0.5$ is applied in all components. Unless otherwise mentioned, ReLU is used as non-linear activation.

	λ_1	λ_2	k
Slot Filling	0.01	1	5
CoNLL NER	0.0001	0.01	1
Amazon	0.002	0.1	1

Table 6: The hyperparameter choices for different experiments.

Bidirectional LSTM is used in the feature extractors for all experiments. In particular, $\mathcal{F}_s$ is a two-layer BiLSTM of hidden size 128 (64 for each direction), and $\mathcal{F}_p$ is a two-layer BiLSTM of hidden size 128 stacked with a `MoE` module (see Figure 2). Each expert network in the `MoE` module of $\mathcal{F}_p$ is a two-layer MLP again of hidden size of 128. The final layer in the MLP has a $tanh$ activation instead of ReLU to match the LSTM-extracted shared features (with $tanh$ activations). The expert gate is a linear transformation (matrix) of size $128 \times N$, where N is the number of source languages.

On the other hand, the architecture of the task specific predictor $\mathcal{C}$ depends on the task. For sequence tagging experiments, the structure of $\mathcal{C}$ is shown in Figure 6, where each expert in the `MoE` module is a token-level two-layer MLP with a softmax layer on top for making token label predictions. For text classification tasks, a dot-product attention mechanism (Luong et al., 2015) is added after the shared and private features are concatenated. It has a length 256 weight vector that attends to the feature vectors of each token and computes a softmax mixture that pools the token-level feature vectors into a single sentence-level feature vector. The rest of $\mathcal{C}$ remains the same for text classification.

For the language discriminator $\mathcal{D}$, a CNN text classifier (Kim, 2014) is adopted in all experiments. It takes as input the shared feature vectors of each token, and employs a CNN with max-pooling to pool them into a single fixed-length feature vector, which is then fed into a MLP for clas-

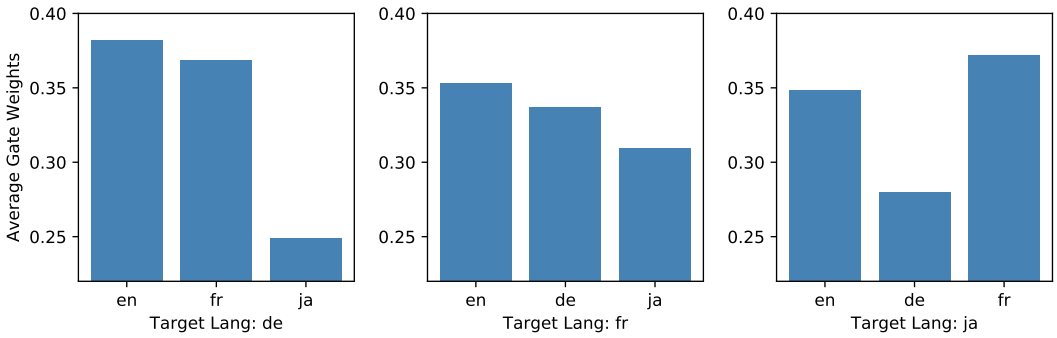

Figure 4: Average expert gate weights aggregated on a language level for the Amazon dataset.

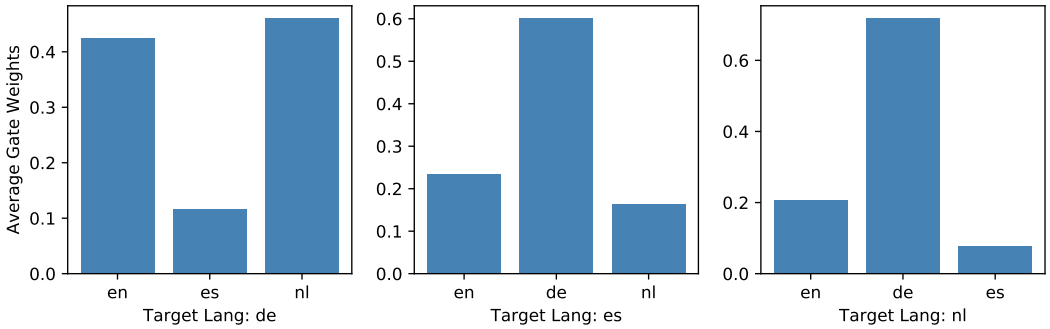

Figure 5: Average expert gate weights aggregated on a language level for the CoNLL dataset.

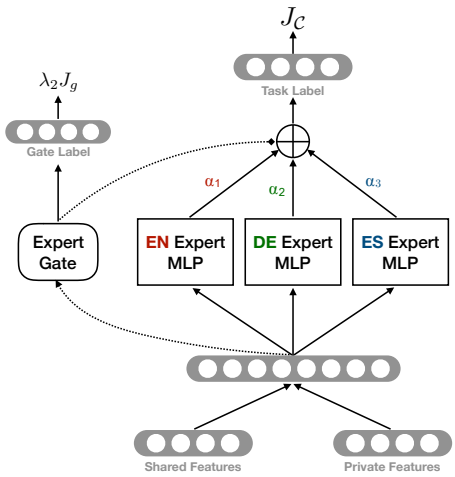

Figure 6: The MoE Predictor $\mathcal{C}$ for Sequence Tagging.

sifying the language of the input sequence. The number of kernels is 200 in the CNN, while the kernel sizes are 3, 4, and 5. The MLP has one hidden layer of size 128.

The MUSE, VecMap, and UMWE embeddings are trained with the monolingual $300d$ fastText Wikipedia embeddings (Bojanowski et al., 2017). When character-level word embeddings are used, a CharCNN is added that takes randomly initialized character embeddings of each character in a word, and passes them through a CNN with kernel number 200 and kernel sizes 3, 4, and 5. Finally, the character embeddings are max-pooled and fed into a single fully-connected layer to form a 128 dimensional character-level word embedding, which is concatenated with the pre-trained cross-lingual word embedding to form the final word representation of that word.

The remaining hyperparameters such as λ_1, λ_2 and k (see Algorithm 1) are tuned for each individual experiment, as shown in Table 6.