Zero-Resource Cross-Lingual Named Entity Recognition

M Saiful Bari,[¶] Shafiq Joty,[¶§] Prathyusha Jwalapuram[¶]
[¶]Nanyang Technological University, Singapore
[§]Salesforce Research Asia, Singapore
{bari0001, jwal0001}@e.ntu.edu.sg, srjoty@ntu.edu.sg

Abstract

Recently, neural methods have achieved state-of-the-art (SOTA) results in Named Entity Recognition (NER) tasks for many languages without the need for manually crafted features. However, these models still require manually annotated training data, which is not available for many languages. In this paper, we propose an unsupervised cross-lingual NER model that can transfer NER knowledge from one language to another in a completely unsupervised way without relying on any bilingual dictionary or parallel data. Our model achieves this through word-level adversarial learning and augmented fine-tuning with parameter sharing and feature augmentation. Experiments on five different languages demonstrate the effectiveness of our approach, outperforming existing models by a good margin and setting a new SOTA for each language pair.

Introduction

Named-entity recognition (NER) is a tagging task that seeks to locate and classify named entities in a text into predefined semantic types such as person, organization, location, etc. It has been a challenging problem mainly because there is not enough labeled data for most languages to learn the specific patterns for words that are part of a named entity. It is also harder to generalize from a small dataset since there can be a wide and often unconstrained variation in what constitutes names. Traditional methods relied on carefully designed orthographic features and language or domain-specific knowledge sources like gazetteers.

With the ongoing neural tsunami, most recent approaches use deep neural networks to circumvent the expensive steps of designing informative features and constructing knowledge sources (Lample et al. 2016; Ma and Hovy 2016; Strubell et al. 2017; Peters et al. 2017; Akbik, Blythe, and Vollgraf 2018; Devlin et al. 2018). However, crucial to their success is the availability of large amounts of labeled training data. Unfortunately, building large labeled datasets for each new language of interest is expensive and time-consuming and we need fairly educated manpower to do the annotation.

As many languages lack suitable corpora annotated with named entities, there have been efforts to design models for *cross-lingual transfer learning*. This offers an attractive solution that allows us to leverage annotated data from a *source* language (e.g., English) to recognize named entities in a *target* language (e.g., German). One possible way to build such a cross-lingual NER system is to encode knowledge about the target language as constraints to regularize the training, which has been tried before for part-of-speech (POS) tagging (Ganchev et al. 2010). However, this would require extensive knowledge of the target language.

Another way is to perform cross-language projection. Most projection-based methods use a parallel sentence-aligned bilingual corpus, or a bi-text. For example, Yarowsky et al. (2001) use an English NER tagger on the English side of a bi-text, then project its token-level predictions to the target side, and finally train a NER tagger on them. Wang and Manning (2014) project model expectations and use them as constraints rather than directly projecting labels, to better transfer information and uncertainty across languages. Joint learning of NER tags and cross-lingual word alignments has also been proposed (Wang, Che, and Manning 2013). Overall, all of these methods require a bi-text with NER tags on one side, which is not typical for low-resource languages. Sentence-aligned parallel corpora are often not available for low-resource languages, and building such corpora could be even more expensive than building the NER dataset.

It is only recently that researchers have proposed cross-lingual NER models for low-resource languages. Lin et al. (2018) propose a multi-lingual multi-task architecture to develop *supervised* NER models with minimal amount of labeled data in the target language. Xie et al. (2018) propose an *unsupervised* transfer model by projecting source language tags into the target language through *word-to-word translation* using the unsupervised word translation model of Conneau et al. (2017). However, this approach has several key limitations. First, for each target language, they need to translate from source to target and learn a brand new NER model. For this, they have to pre-compute a translation dictionary based on nearest neighbour search over the vocabulary items, which is often computationally expensive (e.g., fasttext-en-wiki has ~2M items). This makes it difficult to scale time-

and memory-wise. Furthermore, this often requires (as they do) a target language *labeled development set* to select the best model. Therefore, although the translation process is unsupervised, their NER model is not purely unsupervised.[1] Also, the training of the target language NER model is done without any knowledge about the source.

Comprehensible Output (CO) theory (Swain and Lapkin 1995) of Second-Language Acquisition (SLA) states "learning takes place when a learner encounters a gap in his or her linguistic knowledge of the second language. By noticing this gap, the learner becomes aware of it and may be able to modify his output so that he learns something new about the language". In other words, in SLA, the first language plays an important role in learning the second language.

In this paper, we propose an **unsupervised (or zero-resource)** cross-lingual neural NER model, which allows one to train a model for a target language, using labeled data from a source language. Inspired by the CO theory of SLA, we propose to learn the second language task under the supervision of the first language as opposed to completely forgetting about the first language. Thus, rather than doing word- or phrase-based translation (Xie et al. 2018; Mayhew et al. 2017), we choose to learn a base NER model on the source language first, and then tune the base model further in the presence of both languages to maximize the objective.

Our framework has two encoders – one for the source language and the other for the target. Our source model is based on a bidirectional LSTM-CRF (Lample et al. 2016), which we transfer to a target model in two steps. We first project the mono-lingual word embeddings to a common space through *word-level* adversarial training. The word-level mapping yields initial cross-lingual links between two languages but does not take any NER information into account. Transferring task information in the cross-lingual setup is specifically challenging because languages vary in the word order. To tackle this, we propose an *augmented fine-tuning* method with parameter sharing and feature augmentation, and jointly train the target model in supervision of the source model. In summary, we make the following key contributions:

- We propose a novel unsupervised cross-lingual NER model, assuming no labels in target language, no parallel bi-texts, no cross-lingual dictionaries, and no comparable corpora. To the best of our knowledge, we are the first to show true unsupervised results (validation by source-language) for zero-shot cross-lingual NER.
- Our approach is inspired by the CO theory of how humans acquire a second language, which enables easy transfer to a new language. Our approach only requires the tuning of the pre-trained source model on the (unlabeled) target data.
- We systematically analyze the effect of different components of the model and their contributions for transferring the NER knowledge from one language to another.
- We report sizable improvements over state-of-the-art cross-lingual NER methods on five language pairs encompassing languages from different families (2.43 for Spanish, 2.21 for Dutch, 6.14 for German, 7.1 for Arabic, 5.73 for Finnish). Our method also outperforms the models that use cross-lingual and multilingual external resources.
- We have released our code for research purposes.[2]

[1]We use 'unsupervised' to refer to cross-lingual models that do not use any NER labels in the target language.

Problem Definition

Our objective is to transfer NER knowledge from a source language (e.g., English) to a target language (e.g., German) in an unsupervised way. While doing so, we also wish to provide the landscape of the probable solutions and analyze different solution stages and the importance of different components of the neural model. We make the following assumptions.

- We have access to mono-lingual corpora for both source and target languages to create pretrained word embeddings such as fasttext (Grave et al. 2018).
- For training, we assume that we have NER labels only for the source language dataset.
- We consider two validation scenarios for model selection: (*i*) we have access to a labeled target language validation set, and (*ii*) only source language validation set is available.

Learning **cross-lingual models** involves two fundamental steps: (*i*) learn a mapping between the source and the target language, and (*ii*) retrain the mapped resources to maximize the task objective. These two steps can be done separately or jointly. For example, (Xie et al. 2018) first translate the source sequences to target word-by-word (step *i*), then they learn a target language NER model using the translated texts and projected NER tags (step *ii*). However, as mentioned before, this approach has several key limitations. Besides, training over the (translated) source sequence makes the sequence encoder more dependent on the source language order, which could introduce noise for the target language.

In contrast, we propose to perform mapping and task transfer *jointly*. Our model comprises two encoders – one for the source language and the other for the target. We first train a base NER model on the source language, and use it to jointly train the target model through adversarial learning and augmented fine-tuning. This way, the model is able to learn from both source and target sequences. In the following, we first describe our base model, then we present our novel unsupervised cross-lingual transfer approach.

Our Source (Base) Model

Our source (base) model has the same architecture as Lample et al. (2016), as shown in Figure 1 (the left portion). Given an input sentence $s = (w_1, \ldots, w_m)$ of length m, we first encode each token w_k with a *character-level* bi-LSTM (Hochreiter and Schmidhuber 1997), which gives a token representation $\boldsymbol{w}_k^{\text{ch}}$ by sequentially combining the current input character representation with the previous hidden state in both directions. The character bi-LSTM (shown at the bottom in the box) captures orthographic properties (*e.g.,* capitalization, prefix, suffix) of a token. For each token w_k, we also have a word embedding $\boldsymbol{w}_k^{\text{wr}}$ that we fetch from a pretrained word embedding matrix. The pretrained word vectors capture distributional semantics of the words. We concatenate

[2]https://github.com/ntunlp/Zero-Shot-Cross-Lingual-NER

the character-level representation of a word with its word embedding to get the combined representation $\boldsymbol{x}_k = [\boldsymbol{w}_k^{\text{ch}}; \boldsymbol{w}_k^{\text{wr}}]$.

Let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ denote the representation of the words in the sentence that we get from the character bi-LSTM and embedding lookup. $\boldsymbol{X}$ is then fed into another *word-level* bi-LSTM, which is also processed recurrently to obtain contextualised representations of the words.

The word-level bi-LSTM captures contextual information by propagating information through hidden layers, and can be used directly as a feature for NER classification. However, its modeling strength is limited compared to structured models that use global inference to model consistency in the output, especially in tasks having strong dependencies between output labels such as NER. Therefore, instead of classifying words independently with a Softmax layer, we model them jointly with a CRF layer (Lafferty et al. 2001).

For an input-output sequence pair $(\boldsymbol{X}, \boldsymbol{y})$, we define the joint probability distribution as follows.

$$p(\boldsymbol{y}|\boldsymbol{X}) = \frac{1}{Z(\theta_s)} \prod_{i=1}^{m} \underbrace{\psi_n(y_i|\boldsymbol{u}_i, \boldsymbol{V})}_{\text{node factor}} \prod_{i=0}^{m} \underbrace{\psi_e(y_{i,i+1}|\boldsymbol{A})}_{\text{edge factor}} \quad (1)$$

where $(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)$ are the LSTM encoded contextualized word vectors, and $\psi_n(y_i = j|\boldsymbol{u}_i, \boldsymbol{V}) = \exp(\boldsymbol{V}_j^T \boldsymbol{u}_i)$ is the node-level score with $\boldsymbol{V}$ being the weight matrix, ψ_e is the transition matrix parameterized by $\boldsymbol{A}$, and $Z(.)$ is the normalization constant to ensure a valid probability distribution, and θ_s denotes all the parameters of the (source) model. The cross entropy loss for the $(\boldsymbol{X}, \boldsymbol{y})$ sequence pair is:

$$\mathcal{L}_s(\theta_s) = -\sum_{i=1}^{m} \log \psi_n(y_i|\boldsymbol{u}_i, \boldsymbol{V}) - \sum_{i=0}^{m} \log \boldsymbol{A}_{i,i+1} + \log Z \quad (2)$$

We use Viterbi decoding to infer the most probable tag sequence for an input sequence, $\boldsymbol{y}* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{X}, \theta_s)$.

Following Lample et al. (2016), we use a point-wise dense layer to transform the word representations before passing them to the CRF layer. As described later, the dense layer works as a *common encoder* in our cross-lingual model through which the two encoders share task information and common language properties.

Our Cross-Lingual Model

Our main goal is to learn a mapping of NER distributions between source and target languages. Neural approaches to NER depend heavily on fixed or contextualized pre-trained embeddings (Peters et al. 2018; Devlin et al. 2018; Akbik, Blythe, and Vollgraf 2018). However, when we learn the embeddings for two different languages separately, their distribution spaces are very different even for closely related languages (Søgaard, Ruder, and Vulić 2018). For example, Figure 3a shows the t-SNE plot for NER tagged monolingual embeddings for English and Spanish. We see that the distributions are very different. Mapping these two distributions is indeed a very challenging task, especially in the unsupervised setup where no parallel data or dictionary is given. The challenge is further compounded by the requirement that the mappings should also reflect NER information; the effective

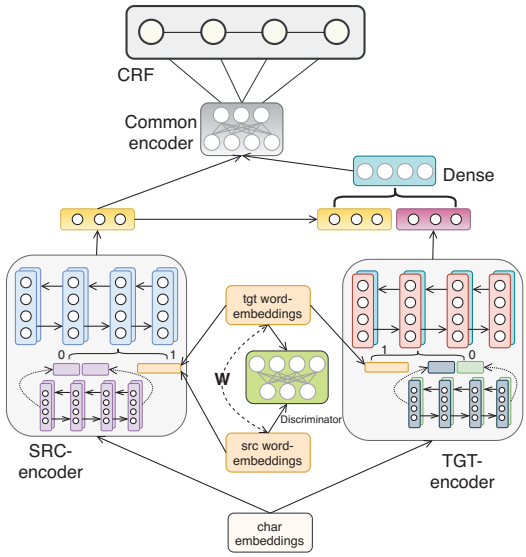


Figure 1: Our proposed model for unsupervised Cross-lingual Named Entity Recognition.

modeling of NER requires the consideration of sequential dependencies, which generally vary between two languages under consideration.

Figure 1 shows the overall architecture of our cross-lingual NER model. We add three new components to the base model described in the previous section: (*i*) a separate encoder for the target language with shared character embeddings (box on the right) followed by a target-specific dense layer, (*ii*) word-level adversarial mappers that can map word embeddings from one language to another (shown in the middle of the two boxes), and (*iii*) an augmented fine-tuning method with parameter sharing and feature augmentation.

Target Encoder with Shared Character Embedding

Our target encoder parameterized by θ_t has the same architecture as the source encoder – a character-level bi-LSTM followed by a word-level bi-LSTM. Having a separate encoder as opposed to a shared one allows us to explicitly model specific characteristics (e.g., morphology, word order) of the respective languages. However, this also adds an additional challenge on how to effectively share the NER knowledge between the two encoders.

To promote knowledge sharing through cross-lingual mapping, we share the character embeddings of the two languages by defining a common embedding matrix. If two languages share alphabets or words, these common features can be used as a prior to learn the mapping.[3]

Word-level Adversarial Mapping

Sharing of character embeddings works only for languages that share alphabets. Even for languages sharing alphabets, it can only provide an initial mapping that is often not good

[3]We also tried subword units with BPE. However, given that the datasets are small, it did not give any additional gain.

enough to learn cross-lingual mappings. To learn the word-level mapping in an unsupervised way, we adopt the adversarial approach of Conneau et al. (2017).

Let $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$ be two sets consisting of n and m word embeddings of d-dimensions for a source and a target language, respectively. We assume that $\mathcal{X}$ and $\mathcal{Y}$ are trained independently from monolingual corpora. Our aim is to learn a mapping $f(\boldsymbol{y})$ in an unsupervised way (*i.e.,* no bi-lingual dictionary is given) such that for every $\boldsymbol{y}_i$, $f(\boldsymbol{y})$ corresponds to its translation in $\mathcal{X}$. Let $\boldsymbol{W}_{t\to s}$ denote the linear mapping weight from target to source, and θ_D denote the parameters of a discriminator D (a binary classifier). We define the discriminator and adversary losses as follows.

$$\mathcal{L}_D(\theta_D|\boldsymbol{W}_{t\to s}) = -\frac{1}{m}\sum_{j=1}^{m}\log P_{\theta_D}(\text{src}=0|\boldsymbol{W}_{t\to s}\boldsymbol{y}_j) - \frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(\text{src}=1|\boldsymbol{x}_i) \quad (3)$$

$$\mathcal{L}_{\text{adv}}(\boldsymbol{W}_{t\to s}|\theta_D) = -\frac{1}{m}\sum_{i=1}^{m}\log P_{\theta_D}(\text{src}=1|\boldsymbol{W}_{t\to s}\boldsymbol{y}_j) - \frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_D}(\text{src}=0|\boldsymbol{x}_i) \quad (4)$$

where $P_{\theta_D}(\text{src}|\boldsymbol{z})$ is the probability according to D to distinguish whether $\boldsymbol{z}$ is coming from the source (src = 1) or from the target-to-source mapping (src = 0). The mapper $\boldsymbol{W}_{t\to s}$ is trained jointly to fool the discriminator D.

Adversarial training gives an initial word-level mapping, which is often not good enough. A refinement step follows, to enrich the initial mapping by considering the global properties of the embedding spaces. Following Conneau et al. (2017), we use refinement with the Procrustes solution, where we first induce a **seed dictionary** using the learned mapper from our adversarial training. In order to find the nearest source word ($\boldsymbol{x}$) of a target word ($\boldsymbol{y}$) in the common space, we use the Cross-domain Similarity Local Scaling (CSLS). With the seed dictionary, we apply the following **Procrustes** solution to improve the initial mappings, $\boldsymbol{W}_{t\to s}$.

$$\boldsymbol{W}_{t\to s} = VU^T, \text{ where } U\Sigma V^T = \text{SVD}(\boldsymbol{X}^T\boldsymbol{Y}) \quad (5)$$

We perform this fine-tuning iteratively: induce a new dictionary using CSLS on the newly learned mapping, then use the dictionary in the Procrustes solution to improve the mapping. The mapper for source to target $\boldsymbol{W}_{s\to t}$ can be similarly trained to map the source embedddings to the target space.

Augmented Fine-tuning

The word-level adversarial training gives a mapping of the words independently. However, NER is a sequence labeling task, and the word order varies from one language to another. Besides, the word-level cross-lingual mapping process does not consider any task information (NER tags); it is simply a word translation model. As a result, the mappings may still lack alignments based on the NER tags. This can be seen in Figure 3b, where the words are mapped to their translations but not clustered according to their NER tags.

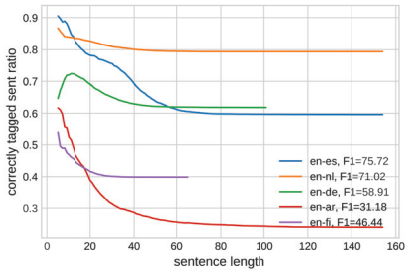


Figure 2: Sentence length vs. correctly tagged target words.

To learn target language ordering information in the target encoder and simultaneously transfer the NER knowledge from the source model, we propose a novel *augmented fine-tuning* method, which works in three steps.

(i) Source model pretraining through weight sharing. We first train an NER model on the source where we have supervision. Our goal is to use this source model to generate pseudo NER labels for the target language sentences in the second step. Therefore, we train the model on the *mapped* representation of the source words. Formally, we optimize:

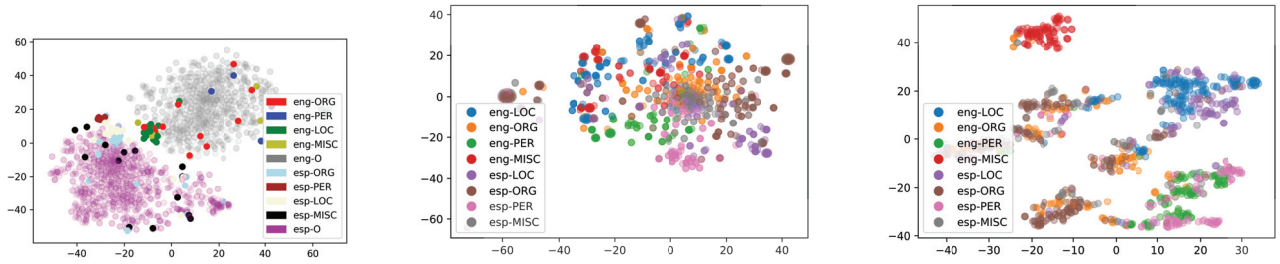
$$\sum_{i=1}^{P}\mathcal{L}_s^i(\theta_s|\boldsymbol{W}_{s\to t}) \quad (6)$$

where $\mathcal{L}_s^i$ is the CRF classification loss in Equation 2 with P being the number of training samples in the source.

The word order in the target language generally differs from the source. To make the model more effective on target sentences, we promote order invariant features in the source encoder by binding the parameters of the forward and backward layers of the character bi-LSTM and word bi-LSTM. Later in our experiments we show its effectiveness. Sharing also reduces the number of parameters and helps to achieve better generalization across languages (Lample et al. 2018). We will refer to this pretrained model as the *mapped source model* or simply *source model* parameterized by θ_s.

(ii) Generating pseudo target labels. Since our source model is already trained in a cross-lingual space, it can directly be applied to infer the NER tags for the target sentences. As shown in Figure 3b, the word-level mapping provides good initial alignments that can be used to produce pseudo training samples in the target language to bootstrap training.

However, since the source model initially does not have any knowledge about the target language word order, it may generate noisy labels as the length of the target sentence increases. For example, Figure 2 shows the ratio of correctly tagged target words for different sentence lengths in different language pairs. We notice that the noise ratio is less for shorter sentences and it increases upto a point as the length increases. To effectively train our models with the pseudo



(a) Mono-lingual embeddings (b) Cross-lingual embeddings (c) Common encoder output distribution

Figure 3: t-SNE plot of NER tagged embeddings of two languages with 1000 samples: (a) Mono-lingual embeddings (fasttext), (b) Cross-lingual embeddings after word-level adversarial training, (c) Embeddings from our common encoder.

target labels, we adopt a stochastic selection method based on sentence length. In particular, we randomly select a length threshold l from a uniform distribution $\mathcal{U}(min, max)$, where min and max are the minimum and maximum (target) sentence lengths respectively, and then we train our models only on sentences that have a maximum of l words; see Algorithm 1. This length restricted stochastic training schedule enables the model to tackle the learning-inference gap between short and long sentences.

(iii) Joint training with feature augmentation. We train our target NER model jointly with the source model with feature augmentation. For each batch from the source, we optimize our source model as before (Equation 6). For each target batch with pseudo labels, we jointly train the source and the target model, and the features from the source encoder are augmented with the features from the target encoder (see Figure 1). The overall loss function of our model is:

$$\mathcal{L}(\theta_s, \theta_t) = \underbrace{\sum_{i=1}^{P} \mathcal{L}_s^i(\theta_s | \boldsymbol{W}_{s\to t})}_{\text{source batch}} + \underbrace{\sum_{j=1}^{Q} \mathcal{L}_t^j(\theta_s)}_{\text{target batch}} + \underbrace{\sum_{j=1}^{Q} \mathcal{L}_t^j(\theta_t)}_{\text{target batch}} \quad (7)$$

where Q is the number of target samples considered for training. This joint training with augmented features ensures that the target model does not overfit on the (potentially) noisy target samples. In a way, the source model guides the target one. Algorithm 1 provides the pseudocode of our training method. Fig. 3c shows a sample output distribution of our common encoder. We can see that the representations are now well clustered based on the NER tags.

Algorithm 1: Augmented fine-tuning for x-lingual NER

Input : Data $\mathcal{D}_S = \{x_i, y_i\}_{i=1}^{P}$, $\mathcal{D}_T = \{x_j\}_{j=1}^{Q}$, Monolingual Embeddings $\boldsymbol{E}_s$ and $\boldsymbol{E}_t$.

`// Word-level adversarial mapping`

1. **repeat**
   repeat
   i) Sample batches $b_s \sim \boldsymbol{E}_s$ and $b_t \sim \boldsymbol{E}_t$
   ii) Update θ_D on disc. loss $\mathcal{L}_D(\theta_D|\boldsymbol{W})$ for b_s and b_t
   until *n_disc_steps*;
   Sample batches $b_s \sim \boldsymbol{E}_s$ and $b_t \sim \boldsymbol{E}_t$
   Update $\boldsymbol{W}$ on adv. loss $\mathcal{L}_{\text{adv}}(\boldsymbol{W}|\theta_D)$ for b_s and b_t
until *w_steps*;

`// Source model pre-training`

2. **repeat**
   i) Sample a batch of sentences $b_s \sim \mathcal{D}_S$
   ii) Update θ_s on CRF classification loss $\mathcal{L}_s(\theta_s)$ for b_s
until *n_steps*;

`// Augmented fine-tuning`

3. Sample a length-threshold l from $\mathcal{U}(min, max)$
4. Use θ_s to infer on $\mathcal{D}_T$ to create a dataset $\mathcal{D}_T^l = \{x_j, \hat{y}_j\}_{j=1}^{Q_l}$
5. **repeat**
   repeat
   i) Sample a batch of sentences $b_s \sim \mathcal{D}_S$ and $b_t \sim \mathcal{D}_T^l$
   ii) Update θ_s on CRF loss $\mathcal{L}_s(\theta_s)$ for b_s and b_t
   iii) Update θ_t on CRF loss $\mathcal{L}_t(\theta_t)$ for b_t
   until *n_steps*;
   Sample a length-threshold l from $\mathcal{U}(min, max)$
   Create a target dataset $\mathcal{D}_T^l = \{x_j, \hat{y}_j\}_{j=1}^{Q_l}$ using θ_s
until *convergence*;

Experimental Settings

Dataset We experiment with five different target languages — Spanish, Dutch, German, Arabic and Finnish. The source language is always English, for which we have sentences tagged with NER classes. The data for English is from the CoNLL-2003 shared task for NER (Sang and Meulder 2003), while the data for Spanish and Dutch is from the CoNLL-2002 shared task for NER (Sang 2002). We collected the Finnish NER dataset from (Ruokolainen et al. 2019)[4] and refactored a few tags. For Arabic, we use *AQMAR Arabic Wikipedia Named Entity Corpus* (Mohit et al. 2012).[5] The corpus contains 28 annotated Wikipedia articles. We randomly take 20% of the sentences from each article to create

[4]Available from https://github.com/mpsilfve/finer-data

[5]http://www.cs.cmu.edu/ ark/ArabicNER/

Language	Train	Dev.	Test
English	14041	3250	3453
Spanish	8323	1915	1517
Dutch	15519	2821	5076
German	12152	2867	3005
Arabic	2166	267	254
Finnish	13497	986	3512

Table 1: Training, Test and Development splits for different datasets. We exclude document start tags (*DOCSTART*).

development and test sets.[6] The NER data is tagged in the IOB1 format. Following the standard practice, we convert it to IOB2 to facilitate evaluation. We train and validate our model in the IOBES format, which is more expressive, for all languages except Arabic. Table 1 presents some basic statistics of the datasets used in our experiments.

Compared Models We experiment with different baselines and variants of our model as described below.

- **Source-Mono:** We train an NER model on the source language with source word embeddings and apply it to the target language with target embeddings, which can be pre-trained or randomly initialized. This model does not use any cross-lingual information.
- **Cross-Word:** We project source and target word embeddings to a common space using the unsupervised mapper ($\boldsymbol{W}_{s\rightarrow t}$ or $\boldsymbol{W}_{t\rightarrow s}$). This model uses word-level cross-lingual information learned from adversarial training and the Procrustes-CSLS refinement procedure.
- **Cross-Shared:** This model is the same as **Cross-Word**, but the weights of the forward and backward LSTM cells are shared to encourage order invariance in the model.
- **Cross-Augmented:** This is our full cross-lingual model trained with source labels and target pseudo-labels generated by the pretrained model and the model itself.

Model Settings We only use sentences with a maximum length of 250 words for training on the source language data. We use FastText embeddings (Grave et al. 2018), which are trained on Common Crawl and Wikipedia, and SGD with a gradient clipping of 5.0 to train the model. We found that the learning rate was crucial for training, and used a decaying rate to scale it down after every epoch. In particular, the learning rate was set to $\max(\frac{lr_0}{1+decay*epoch}, 0.0001)$. The initial learning rate of $lr_0 = 0.1$ and $decay = 0.01$ worked well with a dropout rate of 0.5. We trained the model for 30 epochs while using a batch size of 16, and evaluated the model after every 150 batches. The sizes of the character embeddings and char-LSTM hidden states were set to 25. Our word LSTM's hidden size was set to 100. The details of the hyperparameters are given in our Github repository.[7] We conducted all the experiments in Table 3 and Table 4 five (5) times, and report the mean, standard deviation and maximum value.

[6]Both Arabic and Finnish dataset splits can be found at http://github.com//ntunlp/Zero-Shot-Cross-Lingual-NER

[7]https://github.com/ntunlp/Zero-Shot-Cross-Lingual-NER

	Emb. type	Emb. dim	F_1 score
English			
(Lample et al. 2016)	random	100	83.63
(Lample et al. 2016)	skip-ngram, no-char	100	90.20
(Lample et al. 2016)	skip-ngram	100	90.94
Our	glove	200	91.05±0.37
Our	fasttext	300	89.77±0.19
Spanish			
(Lample et al. 2016)	skip-ngram	64	85.75
(Xie et al. 2018)	glove	300	86.26 ±0.40
Our	fasttext	300	84.71±0.06
Dutch			
(Lample et al. 2016)	skip-ngram	64	81.74
(Xie et al. 2018)	glove	300	86.40±.17
Our	fasttext	300	85.16±0.21
German			
(Lample et al. 2016)	skip-gram–no-char	64	75.06
(Lample et al. 2016)	skip-gram	64	78.76
(Xie et al. 2018)	glove	200	78.16 ± 0.45
Our	fasttext	300	78.14 ± 0.32
Arabic			
Our	fasttext	300	75.49±.53
Finnish			
Our	fasttext	300	84.21±0.13

Table 2: **Monolingual** NER results in the supervised setting.

Results

Monolingual Results

In Table 2, we show the effect of different embeddings on the NER task. We observe that character embeddings contribute very little towards learning the monolingual NER task. Though the monolingual model performs better with GloVe embeddings (Pennington et al. 2014), adversarial training performs better with FastText (Bojanowski et al. 2017), so we use FastText embeddings for all of our experiments.

Source-Mono In Table 3, we show how the source base models perform when they are directly applied to the target language. We can see that the model only learns when character embeddings (shared) are used. Random word embeddings provide better results than monolingual word embeddings.

Cross-lingual Results

Word-level Mapping For all language pairs except En-Ar and En-Fi, projecting word embeddings from the source to the target language achieves the best results. For En-Ar, we could not get reasonable results for source-to-target projection, which is an issue, as discussed by Hoshen and Wolf (2018).[8]

Baseline Results From Table 4 we can see that the **Cross-Word** model with character LSTM performs significantly

[8]See https://github.com/ntunlp/Zero-Shot-Cross-Lingual-NER for detailed results.

Model	Language pair	F_1^* score	F_1 score
Mono-lingual word-emb			
Wrd-LSTM-CRF	en-{es,nl,de,ar,fi}	x	x
Ch-LSTM-Wrd-LSTM-CRF	en-es	33.66±0.90	26.76 ± 1.45
Ch-LSTM-Wrd-LSTM-CRF	en-nl	25.692 ± 1.75	20.94 ± 0.74
Ch-LSTM-Wrd-LSTM-CRF	en-de	12.54 ± 3.07	8.34 ± 1.43
Ch-LSTM-Wrd-LSTM-CRF	en-ar	x	x
Ch-LSTM-Wrd-LSTM-CRF	en-fi	25.05 ± 0.54	22.44 ± 2.23
Random word-emb			
Wrd-LSTM-CRF	en-{es,nl,de,ar,fi}	x	x
Ch-LSTM-Wrd-LSTM-CRF	en-es	36.87 ± 2.46	32.61 ±1.71
Ch-LSTM-Wrd-LSTM-CRF	en-nl	32.47 ±0.92	24.74 ± 0.48
Ch-LSTM-Wrd-LSTM-CRF	en-de	14.70 ± 0.35	11.51 ±0.71
Ch-LSTM-Wrd-LSTM-CRF	en-ar	x	x
Ch-LSTM-Wrd-LSTM-CRF	en-fi	26.05 ± 0.44	17.36 ± 3.34

Table 3: Results for monolingual models applied to target language NER task. 'x' means the model fails to learn anything. F_1^* and F_1 scores are calculated by tuning on the development datasets of the target and source, respectively.

better than the monolingual model (Source-Mono, Table 2) for all languages.

Our Main Results The Cross-Shared model, in which the weights of the forward and backward LSTM cells are shared, gives us 1.73 and 0.71 absolute F1 score increments for the English to Spanish and Dutch language pairs respectively, over the **Cross-Word** (with/without character) model (Table 4). This already achieves a SOTA result by an absolute F1 score of +0.89 for the English-Spanish language pair.

Our Cross-Augmented model, (Tables 4-5) that performs adaptation from source to target language, achieves SOTA performance for all language pairs. It improves over the previous SOTA by 2.43, 2.21, 6.14 and 5.73 F1 for the English to Spanish, Dutch, German and Finnish language pairs respectively, even outperforming multi-lingual models. Our model also outperforms the models that use cross-lingual resources for all languages - including German, which has not been the case in previous works. We also show the effectiveness of our model by reporting results on a "proxy" low-resource[9] dataset (Arabic), where there is no improvement using the **Cross-Shared** model, but a gain of +7.1 F1 using the **Cross-Augmented** method.

Analysis

Char embeddings Contrary to the monolingual case, we find that pretrained source character embeddings make a significant contribution towards transferring NER knowledge in the cross-lingual task, if the two languages have similar morphological features (en-es, en-nl, en-fi). For Arabic (does not share characters with English), the character embeddings only seem to work as noise. However, in case of German, there is a similar noise effect despite the shared characters. Presumably, this is because of the differences in the capitalisation patterns, since German capitalises all nouns.

[9]In the Wikipedia dump as of September 2019, Arabic/English size ratio is 891/16384 (in MB)=.0543 (5.5% of en)

Embedding distribution In the cross lingual model, the baseline results improve significantly. 3a and 3b show the distributions of the pairs of monolingual and cross-lingual embeddings. As the two languages do not share (Fig 3a) any space in their distribution, it is impossible for the model to learn anything. Monolingual embeddings also hamper training; random embeddings increase the transfer score (Table 3), but the model performs poorly with random embeddings for monolingual training (Table 2). However, the result improves in 3. This suggests that we need to search for a better common space for both languages; thus, we perform cross-lingual projection by adversarial training.

Shared LSTM cell In order to get better sequence invariance, we experimented with shared weights in forward and backward LSTM cells. This comes from the idea of **learning less to transfer more**. For Spanish and Dutch, this leads to significant improvements in results along with a 47% reduction in parameters. For German and Finnish there is no significant difference, but the number of parameters are reduced by 54% and 47%. However, for Arabic, there is a drop in the results, probably because of significant word-order differences with the source language (English).

Effect of Sentence Length One of our main assumptions is that pseudo-labels can reduce the entropy of the model (Grandvalet and Bengio 2004). Sentence length is a good feature for finding better pseudo-labels. However, this comes with a cost. To study the effect of sentence length while training the Cross-Augmented model, we perform experiments with sentences of lengths varying from 30 to 150. Figure 2 shows that as the sentence length increases, the ratio of correctly tagged sentences reduces. But if we only train the model on short sentences, the model will overfit on the short sentences of the target language data. Our main model addresses this issue by adding a teacher model and randomly sampling sentence lengths from a uniform distribution.

Source vs. Target NER distribution We report the results of our model tuned on both target and source development data. We see that the model tuned on target development data performs better than the model tuned on source dev data. The results of the source dev data tuned model should be considered as the results under a purely unsupervised setting. These results highlight the differences between the source and target NER distributions. Tuning on the target dev data therefore plays a significant role in the results obtained in cross-lingual NER research thus far. We also tried tuning the model with target test data. Here also we observe a gap between the results. To report stable results, the standard practice should be to report the results of multiple experiments with their standard deviations. Until now, to our knowledge, the only other paper to follow this has been Xie et al. (2018).

Related Work

Lample et al. (2016) proposed an LSTM-CRF model for NER, which passes a hierarchical bi-LSTM encoding to a CRF layer to encourage global consistency of the NER tags.

Model	Emb. prj.	$F_1^{\spadesuit}$ (tuned on tgt-dev)	$F_1^{\spadesuit}$-max	$F_1^{\diamond}$ (tuned on src-dev)	$F_1^{\diamond}$-max	$F_1^{\clubsuit}$ (tuned on tgt-test)	$F_1^{\clubsuit}$-max	# of params
Cross-Word (No char LSTM)	en → es	68.63 ± 1.49	70.62	64.79 ± 1.68	67.42	68.90 ± 1.10	70.62	342906 (~13↓)
	en → nl	65.01 ± 0.53	65.73	64.28 ± 0.71	65.05	65.86 ± 0.29	66.25	342906 (~13↓)
	en → de	58.76 ± 0.70	59.7	57.12 ± 0.53	58.15	59.11 ± 0.37	59.7	342906 (~13↓)
	en ← ar	29.81 ± 1.01	31.18	24.79 ± 0.65	25.46	30.74 ± 0.71	31.18	341890 (~14↓)
	en ← fi	28.77 ± 1.19	30.01	26.55 ± 0.61	27.71	29.99 ± 0.36	30.44	342906 (~13↓)
Cross-Word	en → es	72.66 ± 0.39	73.19	70.49 ± 1.34	72.82	73.62 ± 0.70	74.76	395581 (=1x)
	en → nl	70.31 ± 1.01	71.5	69.24 ± 1.32	70.98	71.22 ± 0.41	71.71	395881 (=1x)
	en → de	45.20 ± 2.78	48.94	30.99 ± 1.08	32.82	46.10 ± 1.68	48.94	395756 (=1x)
	en ← ar	21.39 ± 1.85	24.6	11.84 ± 3.69	15.36	23.37 ± 1.40	24.77	396215 (=1x)
	en ← fi	47.84 ± 1.12	49.53	44.90 ± 1.26	46.09	48.15 ± 0.88	49.53	395356 (=1x)
Cross-Shared	en → es	74.39 ± 0.94	75.72	71.97 ± 0.85	72.54	74.91 ± 0.81	75.72	210081 (~47↓)
	en → nl	71.02 ± 1.20	72.89	68.85 ± 1.87	70.69	71.62 ± 0.89	72.89	210381 (~47↓)
	en → de	58.91 ± 1.03	60.35	56.20 ± 1.38	57.62	59.52 ± 0.62	60.35	182506 (~54↓)
	en ← ar	28.28 ± 1.61	29.82	23.32 ± 0.76	24.35	29.89 ± 0.49	30.72	181490 (~54↓)
	en ← fi	48.04 ± 1.40	49.3	44.36 ± 2.52	48.37	49.31 ± 0.69	50.13	209856 (~47↓)
Cross-Augmented	en → es	75.93 ± 0.81	77.03	72.36 ± 1.17	73.7	76.82 ± 0.84	77.81	661281 (~67↑)
	en → nl	74.61 ±1.24	76.43	69.43 ± 2.43	72.03	75.47 ± 1.25	77.45	661581 (~67↑)
	en → de	65.24 ± 0.56	65.83	59.45 ± 2.56	62.61	65.76 ± 0.41	66.02	636356 (~61↑)
	en ← ar	36.91 ± 2.74	40.36	27.12 ± 3.00	31.84	38.02 ± 2.41	41.63	797215 (~101↑)
	en ← fi	53.77 ± 1.54	56.05	45.69 ± 2.61	50.67	54.42 ± 1.33	56.54	661056 (~67↑)

Table 4: **Cross-lingual** results for **English → Spanish**, **English → Dutch**, **English → German**, **English → Finnish** and **English → Arabic** with respect to different settings. We pick the best performing model amongst the **Cross-Word (No char LSTM)**, **Cross-Word** and **Cross-Shared** models. Using this model as the base, we train the **Cross Augmented** model.

Model	Method	Word Emb.	Lang. Pair				
			en → es	en → nl	en → de	en → ar	en → fi
with cross-lingual resources							
Tackstrom et al. (2012)	Wiki article induction, parallel corpus	-	59.30	58.40	40.40	-	-
(Nothman et al. 2013)	Word cluster features	-	60.55	61.60	48.10	-	-
(Tsai, Mayhew, and Roth 2016)	Feature based methods	-	61.0	64.00	55.80	-	-
(Ni, Dinu, and Florian 2017)	parallel corpus, dict	polyglot emb.	65.10	65.40	58.50	-	-
(Mayhew, Tsai, and Roth 2017)	Cheap Translation, multi-lingual	-	65.95	66.50	59.11	-	-
(Mayhew, Tsai, and Roth 2017)	Cheap Translation, english-only	-	51.82	53.94	50.96	-	-
without cross-lingual resources							
(Xie et al. 2018)	Translate (train on translated src)	fasttext/MUSE, glove	71.03 ± 0.44	71.25 ± 0.79	56.90 ± 0.76	-	-
(Rahimi, Li, and Cohn 2019)	Ranking and Retraining	fasttext/MUSE	71.8	67.6	**59.1**	-	-
(Chen et al. 2018)	MAN-MoE+CharCNN, multi-lingual	fasttext/MUSE	71.0	70.9	56.7	-	-
(Chen et al. 2018)	MAN-MoE+CharCNN, multi-lingual	fasttext/UMWE	**73.5**	**72.4**	56.0	-	-
Our method							
Cross-Shared	Common space proj (tgt→ src)	fasttext/MUSE	74.39 ± .94	71.02 ± 1.20	58.91 ± 1.03	28.28 ± 1.61	48.04 ± 1.40
Cross-Augmented	adaptation to tgt lang	fasttext/MUSE	**75.93 ± 0.81**	**74.61 ± 1.24**	**65.24 ± 0.56**	**36.91 ± 2.74**	**53.77 ± 1.54**

Table 5: Comparison of **Cross-lingual** NER results.

This model achieved impressive results for EN, NL, DE and ES despite not using any explicit feature engineering or manual gazetteers. We extend this base model to a cross-lingual named entity recognizer for a target language using annotated data for a source language and only monolingual, unannotated data for the target.

Mayhew et al. (2017) use a dictionary and co-occurrence probabilities to generate word and phrase based translations of the source data into a target data and then transfer the labels; although the translation quality is poor, the words/phrases and most of the relevant context is preserved, and they are able to achieve good results using a combination of orthographic and Wikifier (Tsai et al. 2016) features. Ni et al. (2017) use weak supervision for cross-lingual NER where they do annotation projection to get target labels and project word embeddings from the target language to the source language. Finally, Yang et al. (2017) used a hierarchical recurrent network for semi-supervised cross-language transfer learning, where the source and the target language share the same character embeddings. Xie et al. (2018) are the first to propose a neural-based model for cross-lingual NER using the (Lample et al. 2016) model, with the addition of a self-attention layer on top of word representation, and validate the model based on target side development dataset.

Acknowledgement

We thank Jiateng Xie, Guillaume Lample and Emma Strubell for sharing their code and embeddings, and for their helpful replies on Github issues and e-mail. Also thanks to Tasnim Mohiuddin for a useful discussion on the hyperparameters of the Word Translation model.

Conclusions and Future Work

In this paper, we contribute a detailed definition of the problem of cross-lingual NER, thus providing a structure to the research to come hereafter. We also propose a new method for

cross-lingual NER that generalizes well by weight-sharing and iteratively adapting to the target language domain, achieving SOTA in the process across languages from different language families. In future work, we want to explore pre-trained language models for cross-lingual NER transfer.

References

Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual string embeddings for sequence labeling. In *COLING*, 1638–1649.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *TACL* 5:135–146.

Chen, X.; Awadallah, A. H.; Hassan, H.; Wang, W.; and Cardie, C. 2018. Zero-resource multilingual model transfer: Learning what to share. *CoRR* abs/1810.03552.

Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; and Jégou, H. 2017. Word translation without parallel data. *CoRR* abs/1710.04087.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Ganchev, K.; Graça, J. a.; Gillenwater, J.; and Taskar, B. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.* 11:2001–2049.

Grandvalet, Y., and Bengio, Y. 2004. Semi-supervised learning by entropy minimization. In *NIPS*, NIPS'04, 529–536. Cambridge, MA, USA: MIT Press.

Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; and Mikolov, T. 2018. Learning word vectors for 157 languages. In *LREC*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Hoshen, Y., and Wolf, L. 2018. An iterative closest point method for unsupervised word translation. *CoRR* abs/1801.06126.

Lafferty, J.; McCallum, A. K.; and Pereira, F. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *CoRR* abs/1603.01360.

Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; and Ranzato, M. 2018. Phrase-based & neural unsupervised machine translation. In *EMNLP*.

Lin, Y.; Yang, S.; Stoyanov, V.; and Ji, H. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *ACL*, 799–809. Melbourne, Australia: Association for Computational Linguistics.

Ma, X., and Hovy, E. H. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR* abs/1603.01354.

Mayhew, S. D.; Tsai, C.-T.; and Roth, D. 2017. Cheap translation for cross-lingual named entity recognition. In *EMNLP*.

Mohit, B.; Schneider, N.; Bhowmick, R.; Oflazer, K.; and Smith, N. A. 2012. Recall-oriented learning of named entities in arabic wikipedia. In *EACL*, EACL '12, 162–173. Stroudsburg, PA, USA: ACL.

Ni, J.; Dinu, G.; and Florian, R. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. *CoRR* abs/1707.02483.

Nothman, J.; Ringland, N.; Radford, W.; Murphy, T.; and Curran, J. R. 2013. Learning multilingual named entity recognition from wikipedia. *Artif. Intell.* 194:151–175.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP'14*, 1532–1543.

Peters, M. E.; Ammar, W.; Bhagavatula, C.; and Power, R. 2017. Semi-supervised sequence tagging with bidirectional language models. *CoRR* abs/1705.00108.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL*.

Rahimi, A.; Li, Y.; and Cohn, T. 2019. Multilingual NER transfer for low-resource languages. *CoRR* abs/1902.00193.

Ruokolainen, T.; Kauppinen, P.; Silfverberg, M.; and Lindén, K. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*.

Sang, E. T. K., and Meulder, F. D. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*.

Sang, E. T. K. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *CoRR* cs.CL/0209010.

Søgaard, A.; Ruder, S.; and Vulić, I. 2018. On the limitations of unsupervised bilingual dictionary induction. In *ACL*, 778–788. ACL.

Strubell, E.; Verga, P.; Belanger, D.; and McCallum, A. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*, EMNLP '17, 2670–2680.

Swain, M., and Lapkin, S. 1995. Problems in Output and the Cognitive Processes They Generate: A Step Towards Second Language Learning. *Applied Linguistics* 16(3):371–391.

Täckström, O.; McDonald, R.; and Uszkoreit, J. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL*, NAACL HLT '12, 477–487. Stroudsburg, PA, USA: ACL.

Tsai, C.-T.; Mayhew, S. D.; and Roth, D. 2016. Cross-lingual named entity recognition via wikification. In *CoNLL*.

Wang, M., and Manning, C. D. 2014. Cross-lingual projected expectation regularization for weakly supervised learning. *TACL* 2:55–66.

Wang, M.; Che, W.; and Manning, C. D. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *ACL*, 1073–1082.

Xie, J.; Yang, Z.; Neubig, G.; Smith, N. A.; and Carbonell, J. G. 2018. Neural cross-lingual named entity recognition with minimal resources. *CoRR* abs/1808.09861.

Yang, Z.; Salakhutdinov, R.; and Cohen, W. W. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In *ICLR*, ICLR '17.

Yarowsky, D.; Ngai, G.; and Wicentowski, R. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *HLT*, HLT '01, 1–8. San Diego, CA, USA: Association for Computational Linguistics.