Neural Cross-Lingual Relation Extraction Based on Bilingual Word Embedding Mapping

Jian Ni and **Radu Florian**

IBM Research AI

1101 Kitchawan Road, Yorktown Heights, NY 10598, USA

{nij, raduf}@us.ibm.com

Abstract

Relation extraction (RE) seeks to detect and classify semantic relationships between entities, which provides useful information for many NLP applications. Since the state-of-the-art RE models require large amounts of manually annotated data and language-specific resources to achieve high accuracy, it is very challenging to transfer an RE model of a resource-rich language to a resource-poor language. In this paper, we propose a new approach for cross-lingual RE model transfer based on bilingual word embedding mapping. It projects word embeddings from a target language to a source language, so that a well-trained source-language neural network RE model can be directly applied to the target language. Experiment results show that the proposed approach achieves very good performance for a number of target languages on both in-house and open datasets, using a small bilingual dictionary with only 1K word pairs.

1 Introduction

Relation extraction (RE) is an important information extraction task that seeks to detect and classify semantic relationships between entities like persons, organizations, geo-political entities, locations, and events. It provides useful information for many NLP applications such as knowledge base construction, text mining and question answering. For example, the entity *Washington, D.C.* and the entity *United States* have a *CapitalOf* relationship, and extraction of such relationships can help answer questions like "What is the capital city of the United States?"

Traditional RE models (e.g., Zelenko et al. (2003); Kambhatla (2004); Li and Ji (2014)) require careful feature engineering to derive and combine various lexical, syntactic and semantic features. Recently, neural network RE models (e.g., Zeng et al. (2014); dos Santos et al. (2015); Miwa and Bansal (2016); Nguyen and Grishman (2016)) have become very successful. These models employ a certain level of automatic feature learning by using word embeddings, which significantly simplifies the feature engineering task while considerably improving the accuracy, achieving the state-of-the-art performance for relation extraction.

All the above RE models are supervised machine learning models that need to be trained with large amounts of manually annotated RE data to achieve high accuracy. However, annotating RE data by human is expensive and time-consuming, and can be quite difficult for a new language. Moreover, most RE models require language-specific resources such as dependency parsers and part-of-speech (POS) taggers, which also makes it very challenging to transfer an RE model of a resource-rich language to a resource-poor language.

There are a few existing weakly supervised cross-lingual RE approaches that require no human annotation in the target languages, e.g., Kim et al. (2010); Kim and Lee (2012); Faruqui and Kumar (2015); Zou et al. (2018). However, the existing approaches require aligned parallel corpora or machine translation systems, which may not be readily available in practice.

In this paper, we make the following contributions to cross-lingual RE:

- We propose a new approach for direct cross-lingual RE model transfer based on bilingual word embedding mapping. It projects word embeddings from a target language to a source language (e.g., English), so that a well-trained source-language RE model can be directly applied to the target language, with no manually annotated RE data needed for the target language.

- We design a deep neural network architecture for the source-language (English) RE model that uses word embeddings and generic language-independent features as the input. The English RE model achieves the-state-of-the-art performance without using language-specific resources.

- We conduct extensive experiments which show that the proposed approach achieves very good performance (up to 79% of the accuracy of the supervised target-language RE model) for a number of target languages on both in-house and the ACE05 datasets (Walker et al., 2006), using a small bilingual dictionary with only 1K word pairs. To the best of our knowledge, this is the first work that includes empirical studies for cross-lingual RE on several languages across a variety of language families, without using aligned parallel corpora or machine translation systems.

We organize the paper as follows. In Section 2 we provide an overview of our approach. In Section 3 we describe how to build monolingual word embeddings and learn a linear mapping between two languages. In Section 4 we present a neural network architecture for the source-language (English). In Section 5 we evaluate the performance of the proposed approach for a number of target languages. We discuss related work in Section 6 and conclude the paper in Section 7.

2 Overview of the Approach

We summarize the main steps of our neural cross-lingual RE model transfer approach as follows.

1. Build word embeddings for the source language and the target language separately using monolingual data.

2. Learn a linear mapping that projects the target-language word embeddings into the source-language embedding space using a small bilingual dictionary.

3. Build a neural network source-language RE model that uses word embeddings and generic language-independent features as the input.

4. For a target-language sentence and any two entities in it, project the word embeddings

of the words in the sentence to the source-language word embeddings using the linear mapping, and then apply the source-language RE model on the projected word embeddings to classify the relationship between the two entities. An example is shown in Figure 1, where the target language is Portuguese and the source language is English.

We will describe each component of our approach in the subsequent sections.

3 Cross-Lingual Word Embeddings

In recent years, vector representations of words, known as *word embeddings*, become ubiquitous for many NLP applications (Collobert et al., 2011; Mikolov et al., 2013a; Pennington et al., 2014).

A monolingual word embedding model maps words in the vocabulary $\mathcal{V}$ of a language to real-valued vectors in $\mathbb{R}^{d \times 1}$. The dimension of the vector space d is normally much smaller than the size of the vocabulary $V = |\mathcal{V}|$ for efficient representation. It also aims to capture semantic similarities between the words based on their distributional properties in large samples of monolingual data.

Cross-lingual word embedding models try to build word embeddings across multiple languages (Upadhyay et al., 2016; Ruder et al., 2017). One approach builds monolingual word embeddings separately and then maps them to the same vector space using a bilingual dictionary (Mikolov et al., 2013b; Faruqui and Dyer, 2014). Another approach builds multilingual word embeddings in a shared vector space simultaneously, by generating mixed language corpora using aligned sentences (Luong et al., 2015; Gouws et al., 2015).

In this paper, we adopt the technique in (Mikolov et al., 2013b) because it only requires a small bilingual dictionary of aligned word pairs, and does not require parallel corpora of aligned sentences which could be more difficult to obtain.

3.1 Monolingual Word Embeddings

To build monolingual word embeddings for the source and target languages, we use a variant of the Continuous Bag-of-Words (CBOW) word2vec model (Mikolov et al., 2013a).

The standard CBOW model has two matrices, the input word matrix $\tilde{\mathbf{X}} \in \mathbb{R}^{d \times V}$ and the output word matrix $\mathbf{X} \in \mathbb{R}^{d \times V}$. For the ith word w_i in $\mathcal{V}$, let $\mathbf{e}(w_i) \in \mathbb{R}^{V \times 1}$ be a one-hot vector with 1 at index i and 0s at other indexes, so that $\tilde{\mathbf{x}}_i = \tilde{\mathbf{X}}\mathbf{e}(w_i)$
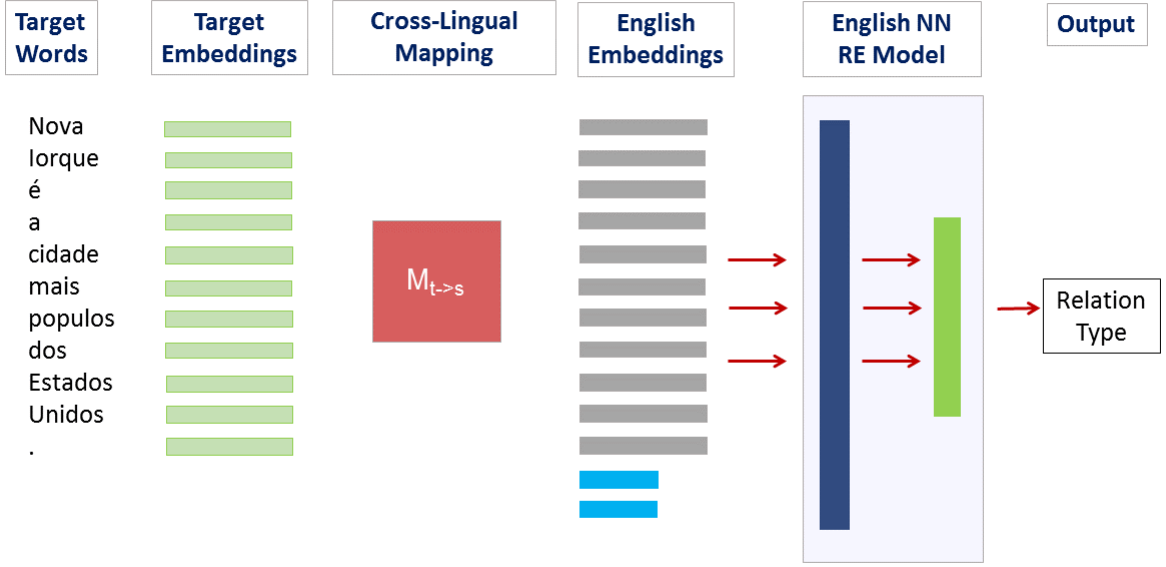
Figure 1: Neural cross-lingual relation extraction based on bilingual word embedding mapping - target language: Portuguese, source language: English.

(the ith column of $\tilde{\mathbf{X}}$) is the input vector representation of word w_i, and $\mathbf{x}_i = \mathbf{X}\mathbf{e}(w_i)$ (the ith column of $\mathbf{X}$) is the output vector representation (i.e., word embedding) of word w_i.

Given a sequence of training words $w_1, w_2, ..., w_N$, the CBOW model seeks to predict a target word w_t using a window of $2c$ context words surrounding w_t, by maximizing the following objective function:

$$\mathcal{L} = \frac{1}{N} \sum_{t=1}^{N} \log P(w_t | w_{t-c}, ..., w_{t-1}, w_{t+1}, ..., w_{t+c})$$

The conditional probability is calculated using a softmax function:

$$P(w_t | w_{t-c}, ..., w_{t+c}) = \frac{\exp(\mathbf{x}_t^{\mathrm{T}} \tilde{\mathbf{x}}_{c(t)})}{\sum_{i=1}^{V} \exp(\mathbf{x}_i^{\mathrm{T}} \tilde{\mathbf{x}}_{c(t)})} \quad (1)$$

where $\mathbf{x}_t = \mathbf{X}\mathbf{e}(w_t)$ is the output vector representation of word w_t, and

$$\tilde{\mathbf{x}}_{c(t)} = \sum_{-c \leq j \leq c, j \neq 0} \tilde{\mathbf{X}}\mathbf{e}(w_{t+j}) \quad (2)$$

is the sum of the input vector representations of the context words.

In our variant of the CBOW model, we use a separate input word matrix $\tilde{\mathbf{X}}_j$ for a context word

at position $j, -c \leq j \leq c, j \neq 0$. In addition, we employ weights that decay with the distances of the context words to the target word. Under these modifications, we have

$$\tilde{\mathbf{x}}_{c(t)}^{new} = \sum_{-c \leq j \leq c, j \neq 0} \frac{1}{|j|} \tilde{\mathbf{X}}_j \mathbf{e}(w_{t+j}) \quad (3)$$

We use the variant to build monolingual word embeddings because experiments on named entity recognition and word similarity tasks showed this variant leads to small improvements over the standard CBOW model (Ni et al., 2017).

3.2 Bilingual Word Embedding Mapping

Mikolov et al. (2013b) observed that word embeddings of different languages often have similar geometric arrangements, and suggested to learn a linear mapping between the vector spaces.

Let $\mathcal{D}$ be a bilingual dictionary with aligned word pairs $(w_i, v_i)_{i=1,...,D}$ between a source language s and a target language t, where w_i is a source-language word and v_i is the translation of w_i in the target language. Let $\mathbf{x}_i \in \mathbb{R}^{d \times 1}$ be the word embedding of the source-language word w_i, $\mathbf{y}_i \in \mathbb{R}^{d \times 1}$ be the word embedding of the target-language word v_i.

We find a linear mapping (matrix) $\mathbf{M}_{t \rightarrow s}$ such that $\mathbf{M}_{t \rightarrow s} \mathbf{y}_i$ approximates $\mathbf{x}_i$, by solving the fol-

lowing least squares problem using the dictionary as the training set:

$$\mathbf{M}_{t \to s} = \arg \min_{\mathbf{M} \in \mathbb{R}^{d \times d}} \sum_{i=1}^{D} ||\mathbf{x}_i - \mathbf{M}\mathbf{y}_i||^2 \quad (4)$$

Using $\mathbf{M}_{t \to s}$, for any target-language word v with word embedding $\mathbf{y}$, we can project it into the source-language embedding space as $\mathbf{M}_{t \to s}\mathbf{y}$.

3.2.1 Length Normalization and Orthogonal Transformation

To ensure that all the training instances in the dictionary $\mathcal{D}$ contribute equally to the optimization objective in (4) and to preserve vector norms after projection, we have tried length normalization and orthogonal transformation for learning the bilingual mapping as in (Xing et al., 2015; Artetxe et al., 2016; Smith et al., 2017).

First, we normalize the source-language and target-language word embeddings to be unit vectors: $\mathbf{x}' = \frac{\mathbf{x}}{||\mathbf{x}||}$ for each source-language word embedding $\mathbf{x}$, and $\mathbf{y}' = \frac{\mathbf{y}}{||\mathbf{y}||}$ for each target-language word embedding $\mathbf{y}$.

Next, we add an orthogonality constraint to (4) such that $\mathbf{M}$ is an orthogonal matrix, i.e., $\mathbf{M}^{\mathrm{T}}\mathbf{M} = \mathbf{I}$ where $\mathbf{I}$ denotes the identity matrix:

$$\mathbf{M}_{t \to s}^{O} = \arg \min_{\mathbf{M} \in \mathbb{R}^{d \times d}, \mathbf{M}^{\mathrm{T}}\mathbf{M}=\mathbf{I}} \sum_{i=1}^{D} ||\mathbf{x}_i' - \mathbf{M}\mathbf{y}_i'||^2$$
$$(5)$$

$\mathbf{M}_{t \to s}^{O}$ can be computed using singular-value decomposition (SVD).

3.2.2 Semi-Supervised and Unsupervised Mappings

The mapping learned in (4) or (5) requires a seed dictionary. To relax this requirement, Artetxe et al. (2017) proposed a self-learning procedure that can be combined with a dictionary-based mapping technique. Starting with a small seed dictionary, the procedure iteratively 1) learns a mapping using the current dictionary; and 2) computes a new dictionary using the learned mapping.

Artetxe et al. (2018) proposed an unsupervised method to learn the bilingual mapping without using a seed dictionary. The method first uses a heuristic to build an initial dictionary that aligns the vocabularies of two languages, and then applies a robust self-learning procedure to iteratively improve the mapping. Another unsuper-

vised method based on adversarial training was proposed in Conneau et al. (2018).

We compare the performance of different mappings for cross-lingual RE model transfer in Section 5.3.2.

4 Neural Network RE Models

For any two entities in a sentence, an RE model determines whether these two entities have a relationship, and if yes, classifies the relationship into one of the pre-defined relation types. We focus on neural network RE models since these models achieve the state-of-the-art performance for relation extraction. Most importantly, neural network RE models use word embeddings as the input, which are amenable to cross-lingual model transfer via cross-lingual word embeddings. In this paper, we use English as the source language.

Our neural network architecture has four layers. The first layer is the *embedding layer* which maps input words in a sentence to word embeddings. The second layer is a *context layer* which transforms the word embeddings to context-aware vector representations using a recurrent or convolutional neural network layer. The third layer is a *summarization layer* which summarizes the vectors in a sentence by grouping and pooling. The final layer is the *output layer* which returns the classification label for the relation type.

4.1 Embedding Layer

For an English sentence with n words $\mathbf{s} = (w_1, w_2, ..., w_n)$, the *embedding layer* maps each word w_t to a real-valued vector (word embedding) $\mathbf{x}_t \in \mathbb{R}^{d \times 1}$ using the English word embedding model (Section 3.1). In addition, for each entity m in the sentence, the embedding layer maps its entity type to a real-valued vector (entity label embedding) $\mathbf{l}_m \in \mathbb{R}^{d_m \times 1}$ (initialized randomly). In our experiments we use $d = 300$ and $d_m = 50$.

4.2 Context Layer

Given the word embeddings $\mathbf{x}_t$'s of the words in the sentence, the *context layer* tries to build a sentence-context-aware vector representation for each word. We consider two types of neural network layers that aim to achieve this.

4.2.1 Bi-LSTM Context Layer

The first type of context layer is based on Long Short-Term Memory (LSTM) type recurrent neural networks (Hochreiter and Schmidhuber, 1997;

Graves and Schmidhuber, 2005). Recurrent neural networks (RNNs) are a class of neural networks that operate on sequential data such as sequences of words. LSTM networks are a type of RNNs that have been invented to better capture long-range dependencies in sequential data.

We pass the word embeddings $\mathbf{x}_t$'s to a forward and a backward LSTM layer. A forward or backward LSTM layer consists of a set of recurrently connected blocks known as *memory blocks*. The memory block at the t-th word in the forward LSTM layer contains a memory cell $\overrightarrow{\mathbf{c}}_t$ and three *gates*: an input gate $\overrightarrow{\mathbf{i}}_t$, a forget gate $\overrightarrow{\mathbf{f}}_t$ and an output gate $\overrightarrow{\mathbf{o}}_t$ ($\overrightarrow{\cdot}$ indicates the forward direction), which are updated as follows:

$$
\begin{aligned}
\overrightarrow{\mathbf{i}}_t &= \sigma(\overrightarrow{W}_i \mathbf{x}_t + \overrightarrow{U}_i \overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_i) \\
\overrightarrow{\mathbf{f}}_t &= \sigma(\overrightarrow{W}_f \mathbf{x}_t + \overrightarrow{U}_f \overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_f) \\
\overrightarrow{\mathbf{o}}_t &= \sigma(\overrightarrow{W}_o \mathbf{x}_t + \overrightarrow{U}_o \overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_o) \\
\overrightarrow{\mathbf{c}}_t &= \overrightarrow{\mathbf{f}}_t \odot \overrightarrow{\mathbf{c}}_{t-1} + \\
&\quad \overrightarrow{\mathbf{i}}_t \odot \tanh(\overrightarrow{W}_c \mathbf{x}_t + \overrightarrow{U}_c \overrightarrow{\mathbf{h}}_{t-1} + \overrightarrow{\mathbf{b}}_c) \\
\overrightarrow{\mathbf{h}}_t &= \overrightarrow{\mathbf{o}}_t \odot \tanh(\overrightarrow{\mathbf{c}}_t) \quad\quad (6)
\end{aligned}
$$

where σ is the element-wise sigmoid function and $\odot$ is the element-wise multiplication.

The hidden state vector $\overrightarrow{\mathbf{h}}_t$ in the forward LSTM layer incorporates information from the left (past) tokens of w_t in the sentence. Similarly, we can compute the hidden state vector $\overleftarrow{\mathbf{h}}_t$ in the backward LSTM layer, which incorporates information from the right (future) tokens of w_t in the sentence. The concatenation of the two vectors $\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t]$ is a good representation of the word w_t with both left and right contextual information in the sentence.

4.2.2 CNN Context Layer

The second type of context layer is based on Convolutional Neural Networks (CNNs) (Zeng et al., 2014; dos Santos et al., 2015), which applies *convolution-like* operation on successive windows of size k around each word in the sentence. Let $\mathbf{z}_t = [\mathbf{x}_{t-(k-1)/2}, ..., \mathbf{x}_{t+(k-1)/2}]$ be the concatenation of k word embeddings around w_t. The convolutional layer computes a hidden state vector

$$
\mathbf{h}_t = \tanh(\mathbf{W}\mathbf{z}_t + \mathbf{b}) \quad\quad (7)
$$

for each word w_t, where $\mathbf{W}$ is a weight matrix and $\mathbf{b}$ is a bias vector, and $\tanh(\cdot)$ is the element-wise hyperbolic tangent function.

4.3 Summarization Layer

After the context layer, the sentence $(w_1, w_2, ..., w_n)$ is represented by $(\mathbf{h}_1, ..., \mathbf{h}_n)$. Suppose $m_1 = (w_{b_1}, .., w_{e_1})$ and $m_2 = (w_{b_2}, .., w_{e_2})$ are two entities in the sentence where m_1 is on the left of m_2 (i.e., $e_1 < b_2$). As different sentences and entities may have various lengths, the *summarization layer* tries to build a *fixed-length* vector that best summarizes the representations of the sentence and the two entities for relation type classification.

We divide the hidden state vectors $\mathbf{h}_t$'s into 5 groups:

- $G_1 = \{\mathbf{h}_1, .., \mathbf{h}_{b_1-1}\}$ includes vectors that are left to the first entity m_1.

- $G_2 = \{\mathbf{h}_{b_1}, .., \mathbf{h}_{e_1}\}$ includes vectors that are in the first entity m_1.

- $G_3 = \{\mathbf{h}_{e_1+1}, .., \mathbf{h}_{b_2-1}\}$ includes vectors that are between the two entities.

- $G_4 = \{\mathbf{h}_{b_2}, .., \mathbf{h}_{e_2}\}$ includes vectors that are in the second entity m_2.

- $G_5 = \{\mathbf{h}_{e_2+1}, .., \mathbf{h}_n\}$ includes vectors that are right to the second entity m_2.

We perform element-wise max pooling among the vectors in each group:

$$
\mathbf{h}_{G_i}(j) = \max_{\mathbf{h} \in G_i} \mathbf{h}(j), 1 \le j \le d_h, 1 \le i \le 5 \quad (8)
$$

where d_h is the dimension of the hidden state vectors. Concatenating the $\mathbf{h}_{G_i}$'s we get a fixed-length vector $\mathbf{h}_s = [\mathbf{h}_{G_1}, ..., \mathbf{h}_{G_5}]$.

4.4 Output Layer

The output layer receives inputs from the previous layers (the summarization vector $\mathbf{h}_s$, the entity label embeddings $\mathbf{l}_{m_1}$ and $\mathbf{l}_{m_2}$ for the two entities under consideration) and returns a probability distribution over the relation type labels:

$$
\mathbf{p} = \text{softmax}(\mathbf{W}_s \mathbf{h}_s + \mathbf{W}_{m_1}\mathbf{l}_{m_1} + \mathbf{W}_{m_2}\mathbf{l}_{m_2} + \mathbf{b}_o) \quad (9)
$$

4.5 Cross-Lingual RE Model Transfer

Given the word embeddings of a sequence of words in a target language t, $(\mathbf{y}_1, ..., \mathbf{y}_n)$, we project them into the English embedding space by applying the linear mapping $\mathbf{M}_{t \to s}$ learned in Section 3.2: $(\mathbf{M}_{t \to s}\mathbf{y}_1, \mathbf{M}_{t \to s}\mathbf{y}_2, ..., \mathbf{M}_{t \to s}\mathbf{y}_n)$. The

neural network English RE model is then applied on the projected word embeddings and the entity label embeddings (which are language independent) to perform relationship classification.

Note that our models do not use language-specific resources such as dependency parsers or POS taggers because these resources might not be readily available for a target language. Also our models do not use precise word position features since word positions in sentences can vary a lot across languages.

5 Experiments

In this section, we evaluate the performance of the proposed cross-lingual RE approach on both in-house dataset and the ACE (Automatic Content Extraction) 2005 multilingual dataset (Walker et al., 2006).

5.1 Datasets

Our in-house dataset includes manually annotated RE data for 6 languages: English, German, Spanish, Italian, Japanese and Portuguese. It defines 56 entity types (e.g., *Person*, *Organization*, *Geo-Political Entity*, *Location*, *Facility*, *Time*, *Event_Violence*, etc.) and 53 relation types between the entities (e.g., *AgentOf*, *LocatedAt*, *PartOf*, *TimeOf*, *AffectedBy*, etc.).

The ACE05 dataset includes manually annotated RE data for 3 languages: English, Arabic and Chinese. It defines 7 entity types (*Person*, *Organization*, *Geo-Political Entity*, *Location*, *Facility*, *Weapon*, *Vehicle*) and 6 relation types between the entities (*Agent-Artifact*, *General-Affiliation*, *ORG-Affiliation*, *Part-Whole*, *Personal-Social*, *Physical*).

For both datasets, we create a class label "O" to denote that the two entities under consideration do not have a relationship belonging to one of the relation types of interest.

5.2 Source (English) RE Model Performance

We build 3 neural network English RE models under the architecture described in Section 4:

- The first neural network RE model does not have a context layer and the word embeddings are directly passed to the summarization layer. We call it **Pass-Through** for short.

- The second neural network RE model has a Bi-LSTM context layer. We call it **Bi-LSTM** for short.

Model	F_1
FCM (S) (Gormley et al., 2015)	55.06
Hybrid FCM (E) (Gormley et al., 2015)	58.26
BIDIRECT (S) (Nguyen and Grishman, 2016)	57.73
VOTE-BW (E) (Nguyen and Grishman, 2016)	60.60
Pass-Through (S)	54.99
Bi-LSTM (S)	58.92
CNN (S)	57.91

Table 1: Comparison with the state-of-the-art RE models on the ACE05 English data (S: Single Model; E: Ensemble Model).

In-House	Training	Dev	Test
English (Source)	1137	140	140
German (Target)	280	35	35
Spanish (Target)	451	55	55
Italian (Target)	322	40	40
Japanese (Target)	396	50	50
Portuguese (Target)	390	50	50
ACE05	**Training**	**Dev**	**Test**
English (Source)	479	60	60
Arabic (Target)	323	40	40
Chinese (Target)	507	63	63

Table 2: Number of documents in the training/dev/test sets of the in-house and ACE05 datasets.

- The third neural network model has a CNN context layer with a window size 3. We call it **CNN** for short.

First we compare our neural network English RE models with the state-of-the-art RE models on the ACE05 English data. The ACE05 English data can be divided to 6 different domains: broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and webblogs (wl). We apply the same data split in (Plank and Moschitti, 2013; Gormley et al., 2015; Nguyen and Grishman, 2016), which uses news (the union of bn and nw) as the training set, a half of bc as the development set and the remaining data as the test set.

We learn the model parameters using Adam (Kingma and Ba, 2015). We apply dropout (Srivastava et al., 2014) to the hidden layers to reduce overfitting. The development set is used for tuning the model hyperparameters and for early stopping.

In Table 1 we compare our models with the best models in (Gormley et al., 2015) and (Nguyen and Grishman, 2016). Our Bi-LSTM model outperforms the best model (single or ensemble) in (Gormley et al., 2015) and the best single model in (Nguyen and Grishman, 2016), without using any language-specific resources such as dependency
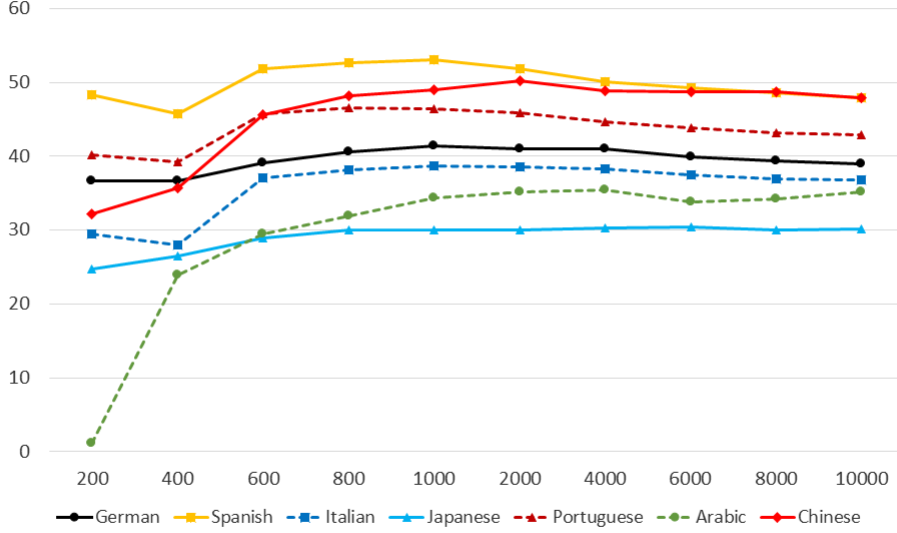
Figure 2: Cross-lingual RE performance (F_1 score) vs. dictionary size (number of bilingual word pairs for learning the mapping (4)) under the Bi-LSTM English RE model on the target-language development data.

Model	In-House			ACE05		
	P	**R**	**F_1**	**P**	**R**	**F_1**
Pass-Through	60.8	63.1	61.9	59.4	55.1	57.2
Bi-LSTM	66.5	67.7	67.1	65.1	65.8	65.5
CNN	63.4	68.8	66.0	61.7	67.1	64.3

Table 3: Performance of the supervised English RE models on the in-house and ACE05 English test data.

parsers.

While the data split in the previous works was motivated by domain adaptation, the focus of this paper is on cross-lingual model transfer, and hence we apply a random data split as follows. For the source language English and each target language, we randomly select 80% of the data as the training set, 10% as the development set, and keep the remaining 10% as the test set. The sizes of the sets are summarized in Table 2.

We report the Precision, Recall and F_1 score of the 3 neural network English RE models in Table 3. Note that adding an additional context layer with either Bi-LSTM or CNN significantly improves the performance of our English RE model, compared with the simple Pass-Through model. Therefore, we will focus on the Bi-LSTM model and the CNN model in the subsequent experiments.

5.3 Cross-Lingual RE Performance

We apply the English RE models to the 7 target languages across a variety of language families.

5.3.1 Dictionary Size

The bilingual dictionary includes the most frequent target-language words and their translations in English. To determine how many word pairs are needed to learn an effective bilingual word embedding mapping for cross-lingual RE, we first evaluate the performance (F_1 score) of our cross-lingual RE approach on the target-language development sets with an increasing dictionary size, as plotted in Figure 2.

We found that for most target languages, once the dictionary size reaches 1K, further increasing the dictionary size may not improve the transfer performance. Therefore, we select the dictionary size to be 1K.

5.3.2 Comparison of Different Mappings

We compare the performance of cross-lingual RE model transfer under the following bilingual word embedding mappings:

- **Regular-1K**: the regular mapping learned in (4) using 1K word pairs;

Mapping	German	Spanish	Italian	Japanese	Portuguese	Arabic	Chinese	Average
Regular-1K	41.4	53.0	38.7	30.0	46.4	34.4	49.0	41.8
Orthogonal-1K	41.0	49.2	35.7	27.1	44.0	32.1	47.2	39.5
Semi-Supervised-1K	38.1	46.7	31.2	25.3	43.1	37.3	47.1	38.4
Unsupervised	37.8	45.5	28.9	21.2	40.8	37.5	49.1	37.3

Table 4: Comparison of the performance (F_1 score) using different mappings on the target-language development data under the Bi-LSTM model.

- **Orthogonal-1K**: the orthogonal mapping with length normalization learned in (5) using 1K word pairs (in this case we train the English RE models with the normalized English word embeddings);

- **Semi-Supervised-1K**: the mapping learned with 1K word pairs and improved by the self-learning method in (Artetxe et al., 2017);

- **Unsupervised**: the mapping learned by the unsupervised method in (Artetxe et al., 2018).

The results are summarized in Table 4. The regular mapping outperforms the orthogonal mapping consistently across the target languages. While the orthogonal mapping was shown to work better than the regular mapping for the word translation task (Xing et al., 2015; Artetxe et al., 2016; Smith et al., 2017), our cross-lingual RE approach directly maps target-language word embeddings to the English embedding space without conducting word translations. Moreover, the orthogonal mapping requires length normalization, but we observed that length normalization adversely affects the performance of the English RE models (about 2.0 F_1 points drop).

We apply the vecmap toolkit[1] to obtain the semi-supervised and unsupervised mappings. The unsupervised mapping has the lowest average accuracy over the target languages, but it does not require a seed dictionary. Among all the mappings, the regular mapping achieves the best average accuracy over the target languages using a dictionary with only 1K word pairs, and hence we adopt it for the cross-lingual RE task.

5.3.3 Performance on Test Data

The cross-lingual RE model transfer results for the in-house test data are summarized in Table 5 and the results for the ACE05 test data are summarized in Table 6, using the regular mapping learned

---
[1]https://github.com/artetxem/vecmap

with a bilingual dictionary of size 1K. In the tables, we also provide the performance of the supervised RE model (Bi-LSTM) for each target language, which is trained with a few hundred thousand tokens of manually annotated RE data in the target-language, and may serve as an upper bound for the cross-lingual model transfer performance.

Among the 2 neural network models, the Bi-LSTM model achieves a better cross-lingual RE performance than the CNN model for 6 out of the 7 target languages. In terms of *absolute* performance, the Bi-LSTM model achieves over 40.0 F_1 scores for German, Spanish, Portuguese and Chinese. In terms of *relative* performance, it reaches over 75% of the accuracy of the supervised target-language RE model for German, Spanish, Italian and Portuguese. While Japanese and Arabic appear to be more difficult to transfer, it still achieves 55% and 52% of the accuracy of the supervised Japanese and Arabic RE model, respectively, without using any manually annotated RE data in Japanese/Arabic.

We apply model ensemble to further improve the accuracy of the Bi-LSTM model. We train 5 Bi-LSTM English RE models initiated with different random seeds, apply the 5 models on the target languages, and combine the outputs by selecting the relation type labels with the highest probabilities among the 5 models. This **Ensemble** approach improves the single model by 0.6-1.9 F_1 points, except for Arabic.

5.3.4 Discussion

Since our approach projects the target-language word embeddings to the source-language embedding space preserving the word order, it is expected to work better for a target language that has more similar word order as the source language. This has been verified by our experiments. The source language, English, belongs to the SVO (Subject, Verb, Object) language family where in a sentence the subject comes first, the verb second, and the object third. Spanish, Italian, Portuguese, German (in conventional typology) and Chinese

Model	German			Spanish			Italian			Japanese			Portuguese		
	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$	P	R	F$_1$
Bi-LSTM	39.6	48.9	43.8	54.5	47.6	50.8	41.8	34.2	37.6	33.9	25.1	28.9	52.9	44.5	48.4
CNN	32.5	50.5	39.5	49.3	48.3	48.8	36.6	34.9	35.7	27.3	31.5	29.3	49.0	44.0	46.3
Ensemble	39.6	50.5	44.4	56.9	49.1	52.7	42.6	35.3	38.6	35.3	26.4	30.2	54.9	45.2	49.6
Supervised	*59.3*	*56.4*	*57.8*	*68.4*	*65.4*	*66.8*	*51.4*	*48.3*	*49.8*	*52.7*	*52.0*	*52.4*	*64.0*	*61.3*	*62.6*

Table 5: Performance of the cross-lingual RE approach on the in-house target-language test data.

Model	Arabic			Chinese		
	P	R	F$_1$	P	R	F$_1$
Bi-LSTM	30.3	45.7	36.4	61.7	37.8	46.8
CNN	24.0	39.7	29.9	56.4	33.8	42.3
Ensemble	27.5	48.7	35.2	61.0	40.4	48.6
Supervised	*70.0*	*69.1*	*69.5*	*66.9*	*69.4*	*68.1*

Table 6: Performance of the cross-lingual RE approach on the ACE05 target-language test data.

also belong to the SVO language family, and our approach achieves over 70% relative accuracy for these languages. On the other hand, Japanese belongs to the SOV (Subject, Object, Verb) language family and Arabic belongs to the VSO (Verb, Subject, Object) language family, and our approach achieves lower relative accuracy for these two languages.

6 Related Work

There are a few weakly supervised cross-lingual RE approaches. Kim et al. (2010) and Kim and Lee (2012) project annotated English RE data to Korean to create weakly labeled training data via aligned parallel corpora. Faruqui and Kumar (2015) translates a target-language sentence into English, performs RE in English, and then projects the relation phrases back to the target-language sentence. Zou et al. (2018) proposes an adversarial feature adaptation approach for cross-lingual relation classification, which uses a machine translation system to translate source-language sentences into target-language sentences. Unlike the existing approaches, our approach does not require aligned parallel corpora or machine translation systems. There are also several multilingual RE approaches, e.g., Verga et al. (2016); Min et al. (2017); Lin et al. (2017), where the focus is to improve monolingual RE by jointly modeling texts in multiple languages.

Many cross-lingual word embedding models have been developed recently (Upadhyay et al., 2016; Ruder et al., 2017). An important application of cross-lingual word embeddings is to enable cross-lingual model transfer. In this paper, we apply the bilingual word embedding mapping technique in (Mikolov et al., 2013b) to cross-lingual RE model transfer. Similar approaches have been applied to other NLP tasks such as dependency parsing (Guo et al., 2015), POS tagging (Gouws and Søgaard, 2015) and named entity recognition (Ni et al., 2017; Xie et al., 2018).

7 Conclusion

In this paper, we developed a simple yet effective neural cross-lingual RE model transfer approach, which has very low resource requirements (a small bilingual dictionary with 1K word pairs) and can be easily extended to a new language. Extensive experiments for 7 target languages across a variety of language families on both in-house and open datasets show that the proposed approach achieves very good performance (up to 79% of the accuracy of the supervised target-language RE model), which provides a strong baseline for building cross-lingual RE models with minimal resources.

Acknowledgments

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

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, Vancouver, Canada. 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, Melbourne, Australia. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

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

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1356. Association for Computational Linguistics.

Matthew R. Gormley, Mo Yu, and Mark Dredze. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1774–1784, Lisbon, Portugal. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 748–756. JMLR Workshop and Conference Proceedings.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *NEURAL NETWORKS*, 18(5-6):602–610.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1234–1244, Beijing, China. Association for Computational Linguistics.

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

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Seokhwan Kim, Minwoo Jeong, Jonghoon Lee, and Gary Geunbae Lee. 2010. A cross-lingual annotation projection approach for relation detection. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 564–571, Stroudsburg, PA, USA. Association for Computational Linguistics.

Seokhwan Kim and Gary Geunbae Lee. 2012. A graph-based cross-lingual projection approach for weakly supervised relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 48–53, Stroudsburg, PA, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, ICLR '15.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–43. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

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

Bonan Min, Zhuolin Jiang, Marjorie Freedman, and Ralph Weischedel. 2017. Learning transferable representation for bilingual relation extraction via convolutional neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 674–684, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2016. Combining neural networks and log-linear models to improve relation extraction. In *Proceedings of IJCAI Workshop on Deep Learning for Artificial Intelligence (DLAI)*.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1498–1507, Sofia, Bulgaria. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634. Association for Computational Linguistics.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

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(1):1929–1958.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1670. Association for Computational Linguistics.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 886–896. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. Philadelphia: Linguistic Data Consortium.

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, Brussels, Belgium. Association for Computational Linguistics.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics.

Bowei Zou, Zengzhuang Xu, Yu Hong, and Guodong Zhou. 2018. Adversarial feature adaptation for cross-lingual relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 437–448, Santa Fe, New Mexico, USA. Association for Computational Linguistics.