TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/15 pp633-645
Volume 22, Number 6, December 2017

Multi-Level Cross-Lingual Attentive Neural Architecture for Low Resource Name Tagging

Xiaocheng Feng, Lifu Huang, Bing Qin*, Ying Lin, Heng Ji, and Ting Liu

Abstract: Neural networks have been widely used for English name tagging and have delivered state-of-the-art results. However, for low resource languages, due to the limited resources and lack of training data, taggers tend to have lower performance, in comparison to the English language. In this paper, we tackle this challenging issue by incorporating multi-level cross-lingual knowledge as attention into a neural architecture, which guides low resource name tagging to achieve a better performance. Specifically, we regard entity type distribution as language independent and use bilingual lexicons to bridge cross-lingual semantic mapping. Then, we jointly apply word-level cross-lingual mutual influence and entity-type level monolingual word distributions to enhance low resource name tagging. Experiments on three languages demonstrate the effectiveness of this neural architecture: for Chinese, Uzbek, and Turkish, we are able to yield significant improvements in name tagging over all previous baselines.

Key words: name tagging; deep learning; recurrent neural network; cross-lingual information extraction

1 Introduction

Name tagging plays a vital role in the overall task of Information Extraction (IE) and serves as an intermediate step for subsequent IE tasks, e.g., relation extraction and entity linking. Name tagging is defined as the recognition of a contiguous sequence of textual tokens, which represents the name of an object of a specified class, such as a person, location, or an organization. Traditional methods for English name tagging usually use machine learning algorithms, e.g., Support Vector Machine (SVM) or Conditional Random Field (CRF), and build name tagger from training data with accompanying entity labels. Since the performance of a machine learner is heavily dependent on the choice of data representations[1], a lot of work has focused on designing effective features[2–4] or automatic learning features from data with neural networks[5, 6].

In this paper, we focus on low resource language name tagging. In comparison with English, there are mainly two challenges for low resource name tagging. First, for most low resource languages, language-specific resources, such as gazetteers, Part-of-Speech (POS) tagger or dependency parser, are not available, but also costly and time-consuming to develop, which makes it difficult to adapt English name tagging systems with such traditional and linguistic features to low resource languages within a short time. Recently, deep learning techniques have been proven effective in many NLP tasks, e.g, English name tagging[6, 7], relation classification[8, 9], event extraction[10, 11], and knowledge base representation[12]. Without using any language-specific resources or hand-craft features, neural network based methods heavily depend on the quality of large amount of training data. However, for low resource languages, the training data for both name tagging and semantic learning are insufficient. The modeling of word semantics and tagging process with

• Xiaocheng Feng, Bing Qin, and Ting Liu are with College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China. E-mail: xcfeng@ir.hit.edu.cn; qinb@ir.hit.edu.cn; tliu@ir.hit.edu.cn.
• Lifu Huang, Ying Lin, and Heng Ji are with College of Computer Science, Rensselaer Polytechnic Institute, Troy 12180, USA. E-mail: huangl7@rpi.edu; liny9@rpi.edu; jih@rpi.edu.
* To whom correspondence should be addressed.
Manuscript received: 2016-12-31; revised: 2017-04-26; accepted: 2017-06-14

limited available data is another challenge.

Previous studies[13–15] pointed out that different languages usually contain complementary cues about entities. Motivated by this, we propose to use cross-lingual information, especially information from high resource languages, to improve mono-lingual low resource name tagging. Specifically, we bridge the cross-lingual gap by using bilingual lexicons (the word in Low resource Language (LL) with its translations in High resource Language (HL)) and their feature representations, which contain bilingual distributed and word-formation features. For example, in Fig. 1, the word " 本" is common in Chinese but rarely appears as a translated foreign name. However, an English translation of " 本" is "Ben", which provides a strong semantic clue that this is a personal name in the English space. In addition, most English names are capitalized (word-formation feature), which is an important cue for name tagging systems. Considering Fig. 1 again, "美联储" can be recognized by this feature. These bilingual cues can help recognize monolingual name.

Although the observations are obvious, there are still two remaining problems: first, how to identify the most important target from lexicon-based translation candidates based on contextual information. For example, in Fig. 1, the Chinese word "本" has three candidate translations, namely "Ben", "books", and "originally", and "Ben" should play a bigger role than "book" and "originally" for Chinese name tagging. Second, it is also challenging to effectively map high resource language distribution to low resource languages. For instance, if we know that "Ben" has a high probability to be a personal name, the Chinese word "本" also has a high probability to be a personal name.

Considering the first problem, we design a word-level bilingual attention based recurrent neural network for ranking the importance of translation candidates. Specifically, we first employ a standard Bidirectional Long Short-Term Memory (Bi-LSTM)[16] model to learn the semantic representation of each word in the sentence. Then, we build a word-level attention

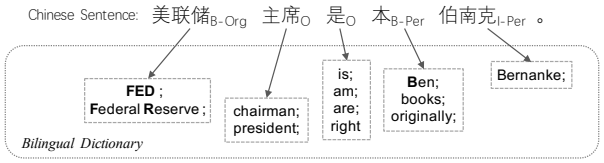


Fig. 1 Example of NER labels with bilingual dictionary.

over all translation candidates, which is expected to dynamically enhance the weight of meaningful translation item. For example, our model will give a higher weight to "Ben" than the weight given to the two other items. This attention mechanism has also been successfully applied to different NLP tasks such as machine translation[17], relation classification[18], and sentiment classification[19].

For the second challenging problem, we develop an entity type-level monolingual attention based neural network for calculating the similarity between translation candidates and each entity type. Unlike previous cross-lingual studies[20, 21], we do not need to project two languages into a common space or map one language space into the other. In this paper, we regard entity type distribution as language independent. For instance, in any languages, the names referring to the United Nations should be an organization and Barack Obama should be a person. We first construct a vector representation of each entity type (person, organization, and location) in low and high resource languages, which can be obtained from training data. Subsequently, we use a cosine measure to calculate the weight between translation candidates and entity types in high resource languages. The higher the attention weight, the higher the probability that the candidate belongs to the specific entity type. Finally, we map word-level bilingual attention and entity type-level monolingual attention weights back to low resource language space and learn new low resource language word representations, which could better reflect the name types.

We combine these two attention models dubbed Multi-level Cross-lingual Attentive Network (MCAN), to capture word semantics both in low resource and high resource simultaneously by bilingual lexicons. Figure 2 shows the work flow of the proposed approach. MCAN takes a sentence with varied lengths as well as the translation of each word by a bilingual dictionary, as input. Then, it outputs a BIO (begin, inside, outside of a name, see Section 2.4 for details) tag for each word in a sentence, such as the example shown in Fig. 1. Our model could be trained in an end-to-end way with standard back-propagation, where the loss function is the cross-entropy error of the output BIO labels.

We evaluate MCAN on the name tagging task for a wide range of languages: Chinese, Uzbek, and Turkish as low resource languages and English as the high resource language. We evaluated Chinese name

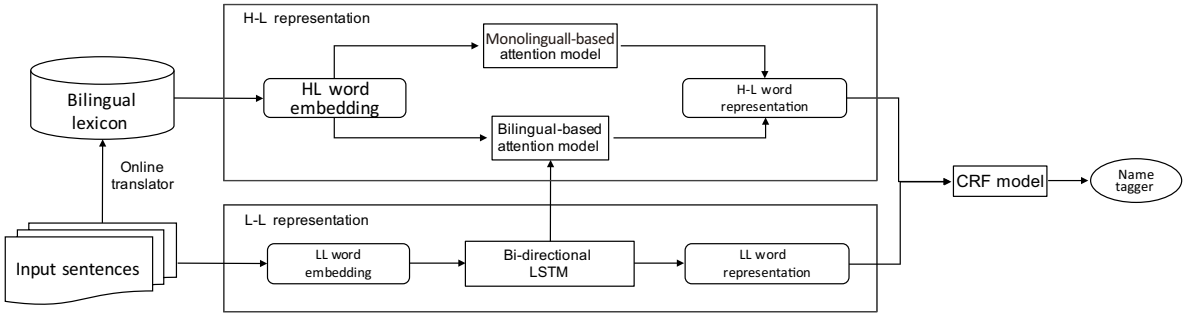


Fig. 2 Flowchart of the proposed system.

tagging on the standard OntoNotes Chinese dataset. Our approach achieved an F-score of 82.13% with 1.64% and 7.11% absolute improvement, compared to previous neural network based methods and bilingual name taggers. Furthermore, for Uzbek and Turkish, experiments show that our bilingual dictionary based method yields significant improvement (average 4%) over the state-of-the-art neural network based approaches. The main contributions of this study are as follows:

- We present a new neural architecture (MCAN) by leveraging cross-lingual information for sentence-level name tagging.
- Our model can be easily applied to any other languages only with bilingual dictionaries.
- We report empirical results on the Chinese OntoNotes dataset, and show that MCAN outperforms state-of-the-art methods for name tagging.

The rest of this paper is organized as follows. In Section 2, we give a brief description of neural name tagging. In Section 3, we introduce the details of our method on how to integrate cross-lingual information into current neural models. We show our experiments and results on three languages in Section 4. We introduce related works in Section 5 and conclude this paper in Section 6.

2 Background: Neural Name Tagger

In this section, we give a brief overview of neural name tagging. More specifically, we first describe a bi-directional LSTM with a sequential conditional random layer for name tagging[17], which will be used in later experiments. In addition, we introduce a character-based model of words which can learn character-level features instead of hand-engineered prefix and suffix information about words.

2.1 Bidirectional LSTM

Recurrent Neural Networks (RNNs) are a family of neural networks operating on sequential data. They take the sequence of words as input and each token is assigned a label. Usually, each token will be first transformed to a vector, which is usually learned by distributional semantics, e.g., Skip-Gram model[22, 23]. All word vectors are stacked in a word embedding matrix $\boldsymbol{L} \in \mathbb{R}^{d \times |V|}$, where d is the dimension of word vector and $|V|$ is the vocabulary size.

Standard RNNs usually suffer from the problem of the gradient vanishing or exploding[24]. Long Short-Term Memory networks (LSTMs) have been designed to address this issue by incorporating a more sophisticated and powerful LSTM cell as the transition function, so that long-distance semantic correlations in a sequence can be modeled better.

In this paper, we employ a Bi-LSTM[25], which is composed of two LSTM neural networks, a forward LSTM_F and a backward LSTM_B, to model the preceding and following contexts, respectively. The input of LSTM_F is the preceding context plus the word as a name candidate, and the input of LSTM_B is the following context plus the word as a name candidate. We run LSTM_F from the beginning to the end of a sentence, and run LSTM_B from the end to the beginning of a sentence. Afterwards, we concatenate the output $\boldsymbol{F}_i$ and $\boldsymbol{B}_i$ of LSTM_F and LSTM_B as the representation of a word using this model $\boldsymbol{h}_i = [\boldsymbol{F}_i, \boldsymbol{B}_i]$, as illustrated in Fig. 3.

2.2 CRF tagging models

A very simple but surprisingly effective tagging model consists of using $\boldsymbol{h}_i$ as features for making independent tagging decisions for each output[26]. Despite this model's success in simple problems like POS tagging, its independent classification decisions

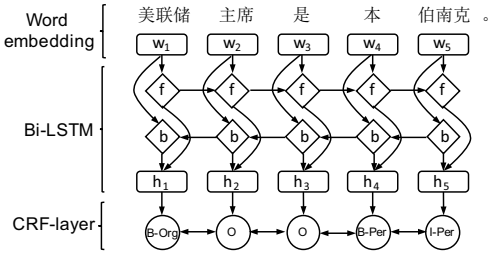


Fig. 3 Main architecture of LSTM-CRF model.

are limited, when there are strong dependencies across output labels. Name is one of such tasks, since the "grammar" characterizing interpretable sequences of tags impose several hard constraints (e.g., I-PER cannot follow B-LOC, see Section 2.3 for details), which would be impossible to model with independence assumptions. Therefore, instead of modeling tagging decisions independently, we model them jointly using a Conditional Random Field (CRF)[27]. For an input sentence,

$$\boldsymbol{X} = (x_1, x_2, ..., x_n),$$

we consider $\boldsymbol{P}$ to be the matrix of scores output by the bidirectional LSTM network. $\boldsymbol{P}$ is of size $n \times k$, where k is the number of distinct tags, and $\boldsymbol{P}_{i,j}$ corresponds to the score of the j-th tag of the i-th word in a sentence. For a sequence of predictions,

$$\boldsymbol{Y} = (y_1, y_2, ..., y_n),$$

we define its score as

$$s(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=0}^{n} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=0}^{n} \boldsymbol{P}_{i, y_i} \tag{1}$$

where $\boldsymbol{A}$ is a matrix of transition scores and $\boldsymbol{A}_{w,v}$ represents the score of a transition from tag w to tag v, $w, v \in \{1, ..., k\}$. $\boldsymbol{Y}$ is the set of tags, each of which corresponds to the specific input word. Thus $\boldsymbol{A}$ is a square matrix of size $k + 2$. A softmax over all possible tag sequences yields the probability for sequence $\boldsymbol{Y}$:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \frac{\mathrm{e}^{s(\boldsymbol{X},\boldsymbol{Y})}}{\sum_{\tilde{y} \in \boldsymbol{Y}_{\mathbf{x}}} \mathrm{e}^{s(\boldsymbol{X}, \tilde{y})}} \tag{2}$$

During training, we maximize the log probability of the correct tag sequence:

$$\log(p(\boldsymbol{Y}|\boldsymbol{X})) = s(\boldsymbol{X}, \boldsymbol{Y}) - \log\Big(\sum_{\tilde{y} \in \boldsymbol{Y}_x} = \mathrm{e}^{s(\boldsymbol{X}, \tilde{y})}\Big) = s(\boldsymbol{X}, \boldsymbol{Y}) - \log \max_{\tilde{y} \in \boldsymbol{Y}_x} s(\boldsymbol{X}, \tilde{y}) \tag{3}$$

where $\boldsymbol{Y}_x$ represents all possible tag sequences (even those which do not verify the BIO format) for a sentence $\boldsymbol{X}$. From the above formulation, it is evident that we encourage our network to produce a valid sequence of output labels. During decoding, we predict the output sequence, which obtains the maximum score given by

$$\boldsymbol{Y}^* = \arg\max_{\tilde{y} \in \boldsymbol{Y}_x} s(\boldsymbol{X}, \tilde{y}) \tag{4}$$

2.3 Character-based models of words

Many languages have orthographic or morphological evidence about whether the word sequence is a name or not. Word spellings have been found useful for morphologically rich languages and for handling the out-of-vocabulary problem for tasks like part-of-speech tagging, language modeling[26], and dependency parsing[28]. To make the word representations sensitive to their spellings, we introduce a character-based model for learning character-level features.

Figure 4 describes the neural architecture to generate an embedding for a word from its characters[6], A character lookup table is initialized randomly, and contains an embedding for every character. The character embeddings corresponding to each character in a word are given in direct and reverse orders to a forward and a backward LSTMs. The embedding of a word derived from its characters is the concatenation of its forward and backward representations from the bidirectional LSTM. This character-level representation is then concatenated with a word-level representation from a word lookup table.

An advantage of RNNs and LSTMs is the capability of encoding very long sequences. However, they also have a bias toward the most recent input. As a result, the final representation of the forward LSTM is an accurate representation of the word's suffix. The final state of the backward LSTM is a better representation of its prefix. In addition, other neural networks, such as convolutional networks, have also

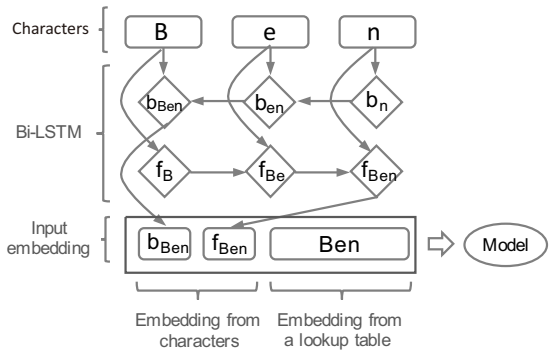


Fig. 4 The character embeddings of the word "Ben" are given as input to a bidirectional LSTM. We concatenate their final output as an embedding from a lookup table to obtain a representation of this word.

been proposed for the learning word representations from their characters[29, 30]. However, convolutional networks are designed to discover the position-invariant features of their inputs. While this is appropriate for many problems, e.g., image recognition (a cat can appear anywhere in a picture), we argue that in our task, the important information is position sensitive (e.g., prefixes and suffixes encode different information from stems). Therefore, we employ a Bi-LSTM model for learning character-level features for morphologically rich languages, such as Uzbek and Turkish, rather than Chinese. In our experiments, the hidden dimensions of the forward and backward character LSTMs are 25 each. As a result, the dimension of our character-based representation of words is 50.

2.4 BIO tagging scheme

The task of name tagging is to assign a tag to every word in a sentence. A single name could contain several tokens within a sentence, therefore name types are usually represented in BIO format (beginning, inside, outside) where every token is labeled as B-label if the token is the beginning of a name, I-label if it is inside a named entity but not the first token within the name, or O, in any other case.

3 MCAN for Name Tagging

In this section, we describe a multi-level cross-lingual attentive network approach for learning low resource word representations with cross-lingual information. We first give an overview of the approach; then, we present a word-level cross-lingual based and an entity type-level monolingual based models, to enrich low resource word representation. Finally, we utilize these representations in name tagging.

3.1 Approach overview

Given a low resource language sentence $\{x_1, x_2, ..., x_i, ..., x_n\}$, by following the setting of Section 2.1, we map each word into its embedding vector and obtain the contextual-/hidden-representation of each word $\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_i, ..., \boldsymbol{h}_n\}$ based on the Bi-LSTM model. In addition, we construct a bilingual lexicon from an online translator. An example is given in Fig. 1. The Chinese word "美联储" has two translation candidates in English, namely "FED" and "Federal Reserve". We also map each English translation into an English embedding vector. If the translation is a single word like "FED", we represent the translation by its word embedding. For cases where the translation is a multi-word expression like "Federal Reserve", the translation representation is the average of its continuous word vectors. To simplify the interpretation, we consider each translation as a single word. For each low resource word x_i, it will have a translation set $\{\mathbf{ht}_{i1}, \mathbf{ht}_{i2}, ..., \mathbf{ht}_{iz}, ..., \mathbf{ht}_{im}\}$, where m is the total number of translations.

An illustration of our approach is given in Fig. 2, we learn the cross-lingual knowledge representation from a bilingual lexicon. We generate consistent entity type vectors in high resource and low resource spaces based on training data. Our joint attention model contains three computational layers. In the first bilingual attention layer, we regard the contextual-representation of each word as input to adaptively selecting important translations from a lexicon-based translation list. In the second computational layer, we build monolingual attention over translations and high resource language entity type vectors, which are expected to learn the similarity between translations and entity types. In the last layer, the output of the previous two layers and the low resource entity type vectors are multiplied and the results are considered as the high resource language entity type distribution of each word x_i, in the low resource space. Figure 5 shows the structure of our full model.

3.2 Word-level bilingual based attention

In this section, we describe our bilingual-based attention model. The basic idea of the attention mechanism is that it assigns a weight/importance to each lower position when computing an upper level representation[31]. In this study, we use an attention model to compute the weight of each translation for learning the representation of a word with cross-lingual semantic features. The intuition is that translation items do not contribute equally to the source word in low resource languages. Furthermore, the meaning of a low resource language word should be different if it occurs in different sentences. Let's take the Chinese sentence "美 联 储 主 席 是 本 伯 南 克 。" (Federal Reserve chairman is Ben Bernanke.) as an example. The context word " 本" is considered as a name and in its English translation setting, "Ben" is more important than "books" and "originally" .

By taking a low resource language word embedding matxix $\boldsymbol{L}$, a Bi-LSTM hidden layer representation $\boldsymbol{h}_i \in \mathbb{R}^d$, and its high resource language translations $\mathbf{ht}_{iz} \in$

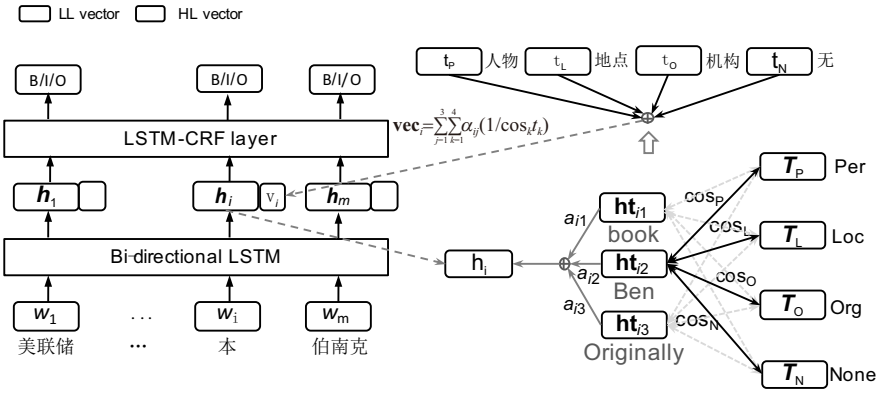


Fig. 5 Main architecture of MCAN model.

$\mathbb{R}^{m\times d}$ as input, the attention model outputs a continuous vector $\mathbf{vec}_i \in \mathbb{R}^d$. The output vector is computed as a weighted sum of each piece of translation in $\mathbf{ht}_{ij}$, namely:

$$\mathbf{vec}_i = \sum_{z=1}^{m} \alpha_z \mathbf{ht}_{iz} \tag{5}$$

where m is the translation size of word x_i, $\alpha_z \in [0,1]$ is the weight of $\mathbf{ht}_{iz}$ and $\sum_z \alpha_z = 1$. For each item of translation $\mathbf{ht}_{iz}$, we use a feed forward neural network to compute its semantic relatedness to the low resource language word x_i. The scoring function is calculated as follows, where $\boldsymbol{W}_{\text{att}} \in \mathbb{R}^{2d}$ and $b_{\text{att}} \in \mathbb{R}$

$$g_z = \tanh(\boldsymbol{W}_{\text{att}}[\mathbf{ht}_{iz}; \boldsymbol{h}_i] + b_{\text{att}}) \tag{6}$$

After obtaining $\{g_1, g_2, ..., g_z\}$, we feed them into a softmax function to calculate the final importance scores $\{\alpha_1, \alpha_2, ..., \alpha_m\}$.

$$\alpha_z = \frac{\exp(g_z)}{\sum_{z=1}^{m} \exp(g_z)} \tag{7}$$

We believe that such an attention model has two advantages: (1) the model could adaptively assign an importance score to each translation item $\mathbf{ht}_{iz}$, according to its semantic relatedness to the low resource language word x_i; (2) this attention model is differentiable, such that it could be easily trained along with other components in an end-to-end fashion.

3.3 Entity type-level monolingual based attention

In this section, we introduce an entity type-level monolingual based attention, which aims to learn the probability distribution of high resource language translation with entity types. One of the significant advantage for deep learning approaches is that word embedding can learn similarities between each word. On this basis, our model is intended to group the vectors of similar entities in the vector space. The details are described below.

First, we generate three high resource language entity type representations, including person, organization, and location. For each entity type, we select 10 typical entities and average their embedding as the entity type representation. Specifically, we aim to generate a rough vector representation for each entity type. Later, these entity type vectors will be fine-tuned by the model. At the same time, we randomly generate one vector representing non-entity; therefore, we obtain four entity type vectors $\boldsymbol{T}=\{\boldsymbol{T}_{\text{P}}, \boldsymbol{T}_{\text{O}}, \boldsymbol{T}_{\text{L}}, \boldsymbol{T}_{\text{N}}\}$, which representing *person*, *organization*, *location*, and *non-entity*, $\boldsymbol{T} \in \mathbb{R}^{4\times d}$, $e \in \{\text{P, O, L, N}\}$, $\boldsymbol{T}_e \in \mathbb{R}^{1\times d}$. In addition, we formalize a scoring function $f(\mathbf{ht}_{iz}, \boldsymbol{T}_e)$, which is capable of measuring the semantic relatedness between the translation item $\boldsymbol{T}_{iz}$ and the high resource language entity type $\boldsymbol{T}_k$. We use standard *cosine* as the dissimilarity measure f, namely:

$$\cos_{ize} = \frac{\mathbf{ht}_{iz}^{\text{T}} \cdot \boldsymbol{T}_e}{||\mathbf{ht}_{iz}|| \times ||\boldsymbol{T}_e||} \tag{8}$$

In this function, the basic idea of the optimizing objective is that the angle between the two vectors should get a low score if the two vectors have high similarity. On the contrary, the angle gets a high score if two vectors have low similarity. Using *cosine* similarity has two advantages: it does not require other parameters and it can keep two vectors in common vector space.

3.4 Low resource language name tagging

We have described our word-level bilingual based model and entity type-level monolingual based model in previous subsections, and obtained two kinds of attention weights: one presenting the relatedness between a low resource language word and its translation, and the other being the entity type distribution of translation items. In this section, we consider the entity type distribution as language

independent and use a linear model to multiply the previous attention weights. The result is a new vector of the low resource language word x_i, in high resource language entity type distributions.

In the first one, we learn an entity type weight distribution of the low resource language word x_i in high resource language space. This distribution can be composed by our previous two-level attention weights; namely,

$$d_{ie} = \sum_{z=1}^{m} \alpha_z \cdot \cos_{ize} \tag{9}$$

Furthermore, we use this distribution to generate a new representation of the low resource language word x_i, by which it is represented in high resource language entity type distribution. We generate four low resource language entity type representations $\boldsymbol{t_e} \in \{\boldsymbol{t}_\mathrm{P}, \boldsymbol{t}_\mathrm{O}, \boldsymbol{t}_\mathrm{L}, \boldsymbol{t}_\mathrm{N}\}$, and calculate the new representation $\boldsymbol{r}_i$ with multiplication distribution d_{ie} :

$$\boldsymbol{r}_i = \sum_{e=1}^{4} d_{ie} \boldsymbol{t}_e \tag{10}$$

In the last one, we regard the output vector $\boldsymbol{r}_i$ as a feature, and concatenate with the Bi-LSTM hidden layer output $\boldsymbol{h}_i$, $\boldsymbol{H}_i = [\boldsymbol{h}_i; \boldsymbol{r}_i]$. Equation (10) means that a semantic representation of the low resource word in low resource language space based on high resource entity type weight distribution. Specifically, entity type weight is language independent. Then, we feed $\boldsymbol{H}_i$ to an LSTM-CRF model (Section 2.2) for the purpose of name tagging. In addition, we apply a dropout mask to the final representation before CRF layer and set the dropout rate to 0.5. The model is trained in a supervised manner by minimizing the cross entropy error of output labels.

We use back propagation to calculate the gradients of all the parameters, and update them with stochastic gradient descent. We learn the 100-dimensional hidden layer of LSTM and attention models, in addition, pre-train the word embedding using word2vec with default settings. We randomize other parameters with uniform distribution $U(-0.01, 0.01)$, and set the learning rate as 0.01. Table 1 illustrates the word embedding parameters used in three low resource language (Chinese (https://catalog.ldc.upenn.edu/Chinese), Uzbek, and Turkish (Low Resource Languages for Emergent Incidents (LORELEI), which contain 23 989 Turkish documents and 16 736 Uzbek documents respectively. http://www.darpa.mil/program/low-resource-languages-for-emergent-incidents) and one high resource language (English (https://catalog.ldc.upenn.edu/LDC2003T05)) in our experiments.

Table 1 Embedding parameters used in our experiments on four languages.

Language	Embedding corpus	Embedding dimension	Vocabulary size
Chinese	Gigword V5	300	66 785
Uzbek	LORELEI	150	382 974
Turkish	LORELEI	300	107 346
English	Gigword V5	300	117 473

4 Experimental

We apply our neural architecture for name tagging on various datasets and evaluate the effectiveness with precision (P), recall (R), and F measure (F1)[32, 33]. In this section, we will describe the detailed experimental settings and discuss the results.

4.1 Data set and evaluation

We evaluate the proposed approach on three low resource languages (including Chinese, Uzbek, and Turkish). In this paper, we regard English as high resource language and all the other languages as low resource, because except for the annotated data, linguistic tools for these languages are not available or suffer low performance.

For Chinese, we aim to verify the effectiveness of our approach on the language, which is distinct from Latin-based languages. We utilize the Ontonotes 4.0 corpus as a Chinese benchmark dataset. OntoNotes annotates 18 named entity types, such as person, location, date, and money. In this paper, we selected the three most common entity types, i.e., PER (person), LOC (location), ORG (organization). For Uzbek and Turkish, we used a data set from the DARPA program, namely, Low Resource Languages for Emergent Incidents (LORELEI). The LORELEI program released the labeled corpora for a bunch of low resource languages, including Uzbek and Turkish. We chose Uzbek and Turkish because most LORELEI related publications are based on these two languages. In this dataset, we also focus on four entity types (Person, Location, Organization, Others). These four entity types are commonly adopted in previous name tagging studies[14, 15], especially for low resource language name tagging[34]. In addition, we fed low resource language words to Bing (http://cn.bing.com/dict/) and two online translators (http://www.translatos.com/en/)

and obtained their translations.

Table 2 shows the detailed descriptions of the data sets used in our experiments. Specifically, we follow the settings in previous studies[15] and do not perform parameter tuning on the dev set, to optimize performance. Instead, we fix the initial learning rate as 0.01 and maximum iterations as 50.

4.2 Chinese name tagging results

We compare with the following baseline methods on the Chinese dataset.

4.2.1 Baseline methods

- Stanford NER. This is a CRF-based NER tagger (using Viterbi decoding) as our baseline monolingual NER tool. English features were taken from Finkel et al.[35] Table 3 lists the basic features of Chinese NER, where $\circ$ means string concatenation and y_i is the named entity tag of the i-th word w_i. Moreover, shape(w_i) is the shape of w_i, such as date and number. prefix/suffix(w_i, k) denotes the k-character's prefix/suffix of w_i. radical(w_i, k) denotes the radical of the k-th Chinese character w_i (the radical of a Chinese character can be found at www.unicode.org/charts/unihan.html). len(w_i) is the number of Chinese characters in w_i.
- Integer Linear Program (ILP) was proposed by Roth and Yih[36]. This algorithm can capture more task-specific and global constraints than the vanilla Viterbi algorithm. In this task, we re-define the conditional probability as $P_{\mathrm{MAR}}(y|x) = \prod_{i=1} P(y_i|x)$, where $P(y_i|x)$ is the marginal probability given by an underlying CRF model computed using forward-backward inference.
- Soft-Align. Che et al.[14] proposed a novel ILP-based inference algorithm with bilingual constraints for NER. This method can jointly infer bilingual named entities without using any annotated bilingual corpus.
- Joint Model. Wang et al.[15] introduced a graphical model, which combines two Hidden Markov Model (HMM) word aligners and two CRF NER taggers into a joint model, and presented a dual decomposition inference method for performing efficient decoding over this model.

Our model has several variations, which are detailed below.

- Recurrent Neural Network (RNN) is a standard recurrent neural network, which is used for learning sequence representation and the representation of these words can be naturally considered as the features to identity and classify the named entity.
- LSTM-CRF. This model is proposed by Lample et al.[6], which is introduced in Section 2.2. It is an update version of RNN. We replace the standard recurrent neural network with LSTM and add a CRF layer to impose several hard constraints of the "grammar".
- MCAN* adds one-dimensional word-formation feature in the last hidden layer, which means that, if a translation item is capitalized, the value is 1, else the value is 0.
- MCAN$^-$ is a variant version of our approach, which replaces $\boldsymbol{H}_i = [\boldsymbol{h}_i; \boldsymbol{r}_i]$ with $\boldsymbol{H}_i = [\boldsymbol{h}_i; \boldsymbol{x}_i; \boldsymbol{r}_i]$.

Table 2 Number of sentences used in our experiments on three languages.

Language	Train	Test
Chinese	22 761	2746
Uzbek	8298	3040
Turkish	3622	2121

Table 3 Basic features of Chinese name tagging.

Chinese NER Templates
00:1(class bias param)
01: $w_{i+k}, -1 \leqslant k \leqslant 1$
02: $w_{i+k-1} \circ w_{i+k}, -1 \leqslant k \leqslant 1$
03: shape(w_{i+k}), $-4 \leqslant k \leqslant 4$
04: prefix(w_i, k), $1 \leqslant k \leqslant 4$
05: prefix(w_{i-1}, k), $1 \leqslant k \leqslant 4$
06: suffix(w_i, k), $1 \leqslant k \leqslant 4$
07: prefix(w_{i-1}, k), $1 \leqslant k \leqslant 4$
08: radical(w_i, k), $-1 \leqslant k \leqslant$ len(w_i)
Unigrams Features
$y_i \circ$ 00-08
Unigrams Features
$y_{i-1} \circ y_i \circ$ 00-08

4.2.2 Results and analysis

Table 4 shows the empirical results of the baseline methods and our approach on the Chinese dataset. Compared with feature-based methods, such as Stanford NER, ILP, Soft-align, and Joint Model, neural network based methods (including RNN, Bi-LSTM, MCAN) perform better because they can make better use of word semantic information and avoid the errors propagated from NLP tools which may hinder the performance for entity recognition. In addition, the performance of ILP with only monolingual constraints is quite lower with the CRF results. The better Stanford

Table 4 Comparison of different methods on Chinese name tagging.

Method	Precision (%)	Recall (%)	F-score (%)
Stanford NER	82.50	66.58	73.69
ILP	76.2	63.06	69.01
Soft-Align	77.71	72.51	75.02
Joint Model	76.43	72.32	74.32
RNN	78.69	70.54	74.39
LSTM-CRF	79.56	81.44	80.49
MCAN	**85.02**	78.56	81.66
MCAN*	82.38	**81.89**	**82.13**
MCAN$^-$	79.92	80.94	80.42

NER performance on Chinese is probably due to more accurate marginal probabilities estimated by the CRF model. What is more, for the soft-align model, which contains more word alignment pairs and uses probabilities to cut wrong word alignments, both the recall and precision are improved in comparison to ILP. The joint model can acquire the best performance among feature-based methods. The reason may be that the joint decoding algorithm promotes an effect of "soft-union", by encouraging the two unidirectional aligners to agree more often.

For neural network based methods, MCAN* can achieve 1.7% performance improvement on the name tagging's F-measure over the LSTM-CRF. This is due to MCAN* incorporating high resource language semantics to low resource language. Compared to standard RNN, LSTM-CRF does not suffer from the same problem of the gradient vanishing or exploding[24]. The reason lies in that LSTM uses a more sophisticated and powerful LSTM cell as the transition function, for the long-distance semantic correlations in a sequence to be modeled better. Compared with MCAN, MCAN* gets better recall. This is because MCAN* considers character-level information (e.g., capitalization). Considering Fig. 1 again, the Chinese word "主席" has two translation items and both "chairman" and "president" are not capitalized. Therefore the value is 0. However, for Chinese word "本", the value is 1 because the translation is "Ben", which is capitalized. Specifically, we do not use character-level features on Chinese, because the number of Chinese characters is very large, which seriously affects the speed of computing the model.

4.3 Uzbek and turkish NER results

In this section, we compare MCAN model and LSTM-CRF model, the latter uses only monolingual information. Tables 5 and 6 present the results regarding NER for Uzbek and Turkish, in comparison to LSTM-CRF models. In these two languages, the MCAN model outperforms the LSTM-CRF model, which shows that adding cross-lingual information is useful for low resource language name tagging. For Turkish, we provide the true example of the sentence "Sudan' da kolera salgn.", which means "Sudan cholera outbreak." Our model can recognize "Sudan" as a location, while LSTM-CRF cannot. This is because the training data do not contain "Sudan" as a positive instance. However, MCAN can transfer the English semantics of "Sudan" to Turkish and guide the Turkish name tagger to make a correct decision.

4.4 Comparison on three languages

In this part, we consider our method (MCAN) as an example and conduct a horizontal comparison on Chinese, Uzbek, and Turkish.

Figure 6 shows the experimental results of MCAN and LSTM-CRF for three languages. We can see that both models achieve better precision, recall, and F-score on Chinese, in comparison to the other two languages, and the result of Uzbek is better than Turkish. For this observation, we believe that the reason is caused by the size of training data. It is well accepted that the performance of a machine learner depends on feature representation and size of training data. In addition, we can find that the MCAN model gains reduced gradually with the growth of training data.

4.5 Visualize attention models

We visualize the attention weight of each translation items to obtain a better understanding of the mechanism attention approach. The results of word-level bilingual attention model are given in Fig. 7.

The x-axis and y-axis of each plot correspond to the words in the low resource language sentence (Chinese) and compared translation items in high resource language (English), respectively. Each pixel

Table 5 Results on Uzbek name tagging.

Method	Precision (%)	Recall (%)	F-score (%)
LSTM-CRF	69.14	**68.13**	68.63
MCAN	**74.31**	68.03	**72.45**

Table 6 Results on Turkish name tagging.

Method	Precision (%)	Recall (%)	F-score (%)
LSTM-CRF	67.59	64.11	65.80
MCAN	**75.89**	**65.71**	**70.44**

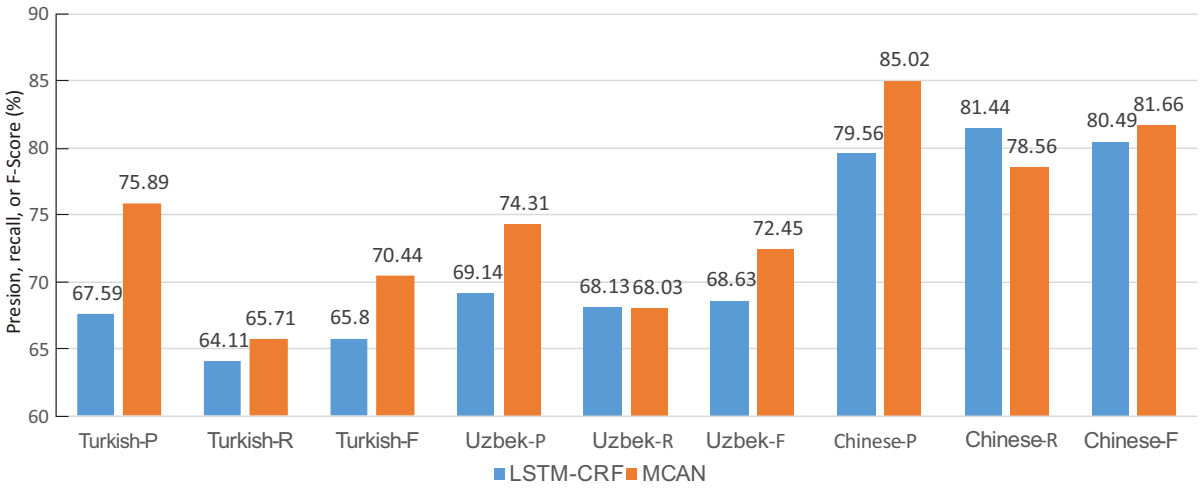


Fig. 6 The comparison of three languages.

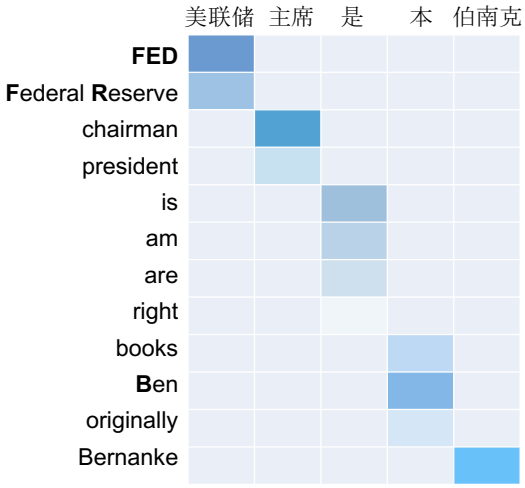


Fig. 7 A sample important weights found by MCAN.

shows the weight α_{ij} of the important weight of the translation item word j-th for the i-th low resource language word (see Eq. (3)), in gray scale (0: gray, 1: blue). Furthermore, we can see from the alignments in Fig. 7 that the alignment of words between Chinese and English is largely monotonic. We see strong weights along the diagonal of each matrix. For instance, attention model is able to correctly attend the Chinese word [“美联储”] with translation [“FED”].

5 Related Work

In this section, we briefly review existing studies on cross-lingual name tagging and neural network approaches to name tagging.

5.1 Cross-lingual name tagging

The idea of utilizing cross-lingual resources to improve monolingual name tagger systems has been studied extensively. Li et al.[13] presented a cyclic CRF model and performed approximate inference using loopy belief propagation. The feature-rich CRF formulation of bilingual edge potentials in their model is powerful; however, a big drawback of this approach is that training such a feature-rich model requires manually annotated bilingual NER data, which can be prohibitively expensive to generate. How and where to obtain the training signals without manual supervision is an interesting and open question. In addition to bilingual corpora, bilingual dictionaries are also useful resources. Huang and Vegel[37] and Chen et al.[38] proposed approaches to extracting bilingual named entity pairs from unannotated bitext, in which verification was based on bilingual named entity dictionaries. Our approach differs in that it does not acquire any unannotated bitext and hand-craft features.

In this regard, one of the most interesting papers is Burkett et al.[39], which explored an “up-training” mechanism by using the outputs from a strong monolingual model as ground-truth, and thereby simulated a learning environment, where a bilingual model is trained to help a “weakened” monolingual model recover the results of the strong model. Kim et al.[40] proposed a method of labeling bilingual corpora with named entity labels automatically based on Wikipedia. However, this method is restricted to topics covered by Wikipedia.

Chen et al.[38] tackled the problem of jointly recognizing and aligning bilingual named entities. Their method employs a set of heuristic rules to expand a candidate named entity set generated by monolingual taggers, and then rank those candidates by

using a bilingual named entity dictionary. In addition, Wang et al.'s approach[15] differs in that it provides a probabilistic formulation of the problem and does not require pre-existing NE dictionaries. Che et al.[14] proposed a novel ILP-based inference algorithm with bilingual constraints for NER. This method can jointly infer bilingual named entities without using any annotated bilingual corpus. In addition, Zhang et al.[34] conducted a thorough study for low resource language name tagging and on various ways of acquiring, encoding and composing expectations from multiple non-traditional sources. Experiments demonstrate that this framework can be used to build a promising name tagger for a new IL within a few hours.

5.2 Neural network for name tagging

Neural network approaches have shown promising results with regard to English name tagging. The power of the neural model lies in its ability of learning continuous text representation from data without any feature engineering. As for the named entity recognition, most of the previous studies consist of two steps. First, they learn the continuous word vector embeddings from the data[23, 41]. Afterwards, sequential compositional approaches are used to compute the vector of each word, with context information. The representative sequential approaches to learning word representation include convolutional neural network[42, 43] and long short-term memory[26].

Gillick et al.[44] modeled the task of sequence-labeling as a sequence to sequence learning problem and incorporated character-based representations into their encoder model. Chiu and Nichols[45] developed a Bi-LSTM architecture to learn word representation in context. They used CNNs to learn character-level features which replace previous word embeddings. Lample et al.[6] employed an architecture similar to Chiu and Nichols[45], but using Bi-LSTM to learn character-level features, in a way similar to the work by dos Santos and Guimarães[7].

6 Conclusion

Low resource name tagging is a very important yet challenging problem in natural language processing. In this work, we designed a multi-level cross-lingual attentive neural architecture, which incorporates both the language independent entity type distribution and bilingual lexicons, to guide low resource name tagging as consistently as in high resource language tagging; namely, English name tagging. Experiments on three low resource languages, namely, Chinese, Uzbek, and Turkish, demonstrate the effectiveness of our neural architecture for name tagging. In the future, we will explore more methods of transferring knowledge from high resource language to low resource languages and mutually improving their name tagging performances.

Acknowledgment

This work was supported by the National High-Tech Development (863) Program of China (No. 2015AA015407) and the National Natural Science Foundation of China (Nos. 61632011 and 61370164).

References

[1] P. Domingos, A few useful things to know about machine learning, *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[2] H. Isozaki and H. Kazawa, Efficient support vector classifiers for named entity recognition, in *Proc. 19th Int. Conference on Computational Linguistics-Volume 1*, Taipei, China, 2002, pp. 1–7.

[3] J. Kazama, T. Makino, Y. Ohta, and J. Tsujii, Tuning support vector machines for biomedical named entity recognition, in *Proc. ACL-02 Workshop on Natural Language Processing in the Biomedical Domain-Volume 3*, Phildadelphia, PA, USA, 2002, pp. 1–8.

[4] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, in *Proc. International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Geneva, Switzerland, 2004, pp. 104–107.

[5] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in *Proc. 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 1188–1196.

[6] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360, 2016.

[7] C. N. dos Santos and V. Guimarães, Boosting named entity recognition with neural character embeddings, arXiv preprint arXiv:1505.05008, 2015.

[8] D. J. Zeng, K. Liu, S. W. Lai, G. Y. Zhou, and J. Zhao, Relation classification via convolutional deep neural network, in *Proc. COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, 2014, pp. 2335–2344.

[9] Y. Xu, L. L. Mou, G. Li, Y. C. Chen, H. Peng, and Z. Jin, Classifying relations via long short term memory networks along shortest dependency paths, in *Proc. Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1785–1794.

[10] X. C. Feng, L. F. Huang, D. Y. Tang, B. Qin, H. Ji, and T. Liu, A language-independent neural network for event detection, in *Proc. 54thAnnual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 66–71.

[11] L. F. Huang, T. Cassidy, X. C. Feng, H. Ji, C. R. Voss, J. W. Han, and A. Sil, Liberal event extraction and event schema induction, in *Proc. the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 258–268.

[12] C. Zhang, M. Zhou, X. Han, Z. Hu, and Y. Ji, Knowledge graph embedding for hyper-relational data, *Tsinghua Sci. Technol.*, vol, 22, no. 2. pp. 185–197, 2017.

[13] Q. Li, H. B. Li, H. Ji, W. Wang, J. Zheng, and F. Huang, Joint bilingual name tagging for parallel corpora, in *Proc. 21st ACM Int. Conference on Information and Knowledge Management*, Maui, HI, USA, 2012, pp. 1727–1731.

[14] W. X. Che, M. Q. Wang, C. D. Manning, and T. Liu, Named entity recognition with bilingual constraints, in *HLT-NAACL*, Atlanta, GA, USA, 2013, pp. 52–62.

[15] M. Q. Wang, W. X. Che, and C. D. Manning, Effective bilingual constraints for semi-supervised learning of named entity recognizers, in *Proc. 27th AAAI Conference on Artificial Intelligence*, Bellevue, WA, USA, 2013, pp. 919–925.

[16] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[17] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2015.

[18] Y. K. Lin, S. Q. Shen, Z. Y. Liu, H. B. Luan, and M. S. Sun, Neural relation extraction with selective attention over instances, in *Proc. 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 2124–2133.

[19] D. Y. Tang, B. Qin, and T. Liu, Aspect level sentiment classification with deep memory network, arXiv preprint arXiv:1605.08900, 2016.

[20] Y. Q. Song, S. Upadhyay, H. R. Peng, and D. Roth, Cross-lingual dataless classification for many languages, in *Proc. 25th Int. Joint Conference on Artificial Intelligence*, New York, NY, USA, 2016, pp. 2901–2907.

[21] J. Guo, W. X. Che, H. F. Wang, and T. Liu, A universal framework for inductive transfer parsing across multi-typed treebanks, in *Proc. COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan, 2016, pp. 12–22.

[22] M. Baroni, G. Dinu, and G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in *Proc. 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, 2014, pp. 238–247.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, in *Advances in Neural Information Processing Systems 26*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.

[24] S. Hochreiter and J. Schmidhuber, LSTM can solve hard long time lag problems, in *Proc. 9th Int. Conference on Neural Information Processing Systems*, Denver, CO, USA, 1996, pp. 473–479.

[25] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks, in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, Curitiba, Brazil, 2007, pp. 367–371.

[26] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso, Finding function in form: Compositional character models for open vocabulary word representation, arXiv preprint arXiv: 1508.02096, 2015.

[27] J. Lafferty, A. McCallum, and F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in *Proc. 18th Int. Conference on Machine Learning*, San Francisco, CA, USA, pp. 282–289.

[28] M. Ballesteros, C. Dyer, and N. A. Smith, Improved transition-based parsing by modeling characters instead of words with LSTMs, arXiv preprint arXiv: 1508.00657, 2015.

[29] X. Zhang, J. B. Zhao, and Y. LeCun, Character-level convolutional networks for text classification, in *Advances in Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 649–657.

[30] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, Character-aware neural language models, arXiv preprint arXiv: 1508.06615, 2015.

[31] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv: 1409.0473, 2014.

[32] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[33] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall, 2000.

[34] B. L. Zhang, X. M. Pan, T. L. Wang, A. Vaswani, H. Ji, K. Knight, and D. Marcu, Name tagging for low-resource incident languages based on expectation-driven learning, in *Proc. NAACL-HLT*, San Diego, CA, USA, 2016, pp. 249–259.

[35] J. R. Finkel, T. Grenager, and C. Manning, Incorporating non-local informating into information extraction systems by Gibbs sampling, in *Proceedings of the 43rd Annual Meeting on Association for Computationa Linguistics*, 2005.

[36] D. Roth and W. Yih, Integer linear programming inference for conditional random fields, in *ICML'05 Proceedings of the 22nd International Conference on Machine Learning*, Bonnn, Germany, 2005.

[37] F. Huang and S. Vogel, Improved named entity translation and bilingual named entity extraction, in *Proc. 4th IEEE Int. Conference on Multimodal Interfaces*, Pittsburgh, PA, USA, 2002, pp. 253–258.

[38] Y. F. Chen, C. Q. Zong, and K.-Y. Su, On jointly recognizing and aligning bilingual named entities, in *Proc. 48th Annual Meeting of the Association for Computational*

Linguistics, Uppsala, Sweden, 2010, pp. 631–639.

[39] D. Burkett, S. Petrov, J. Blitzer, and D. Klein, Learning better monolingual models with unannotated bilingual text, in *Proc. 14th Conference on Computational Natural Language Learning*, Uppsala, Sweden, 2010, pp. 46–54.

[40] S. Kim, K. Toutanova, and H. Yu, Multilingual named entity recognition using parallel data and metadata from Wikipedia, in *Proc. 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, Jeju Island, Korea, 2012, pp. 694–702.

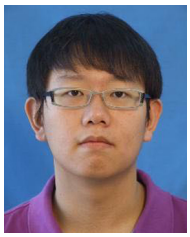
[41] J. Pennington, R. Socher, and C. D. Manning, GloVe: Global vectors for word representation, in *Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.

[42] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv: 1404.2188, 2014.

[43] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv: 1408.5882, 2014.

[44] D. Gillick, C. Brunk, O. Vinyals, and A. Subramanya, Multilingual language processing from bytes, arXiv preprint arXiv: 1512.00103, 2015.

[45] J. P. C. Chiu and E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, arXiv preprint arXiv: 1511.08308, 2015.



Xiaocheng Feng received the master degree in 2013 from Harbin Institute of Technology. Since 2013, he has been a PhD candidate at the Department of Computer Science, Harbin Institute of Technology. His current research interest is information extraction and deep learning.

Lifu Huang received the master degree in 2014 from Peking University. He is currently a PhD student at the School of Computer Science in Rensselaer Polytechnic Institute. His current research interest is event extraction and machine learning.

Bing Qin received PhD degree in computer science from Harbin Institute of Technology in 2005. She is currently a professor at Department of Computer Science and Technology, Harbin Institute of Technology. Her current research interests are information extraction, sentiment classification, and textual entailment.

Ying Lin received the bachelor degree in 2015 from Beijing University of Posts and Telecommunications. He is currently a PhD student at the Department of Computer Science in Rensselaer Polytechnic Institute. His current research interest is machine translation and machine learning.

Heng Ji received the PhD degree in computer science from New York University in 2007. She is currently an associate professor at Department of Computer Science, Rensselaer Polytechnic Institute. Her current research interests are information extraction, image processing, and machine learning.

Ting Liu received the PhD degree in computer science from Harbin Institute of Technology in 1998. He is currently a professor at Department of Computer Science and Technology, Harbin Institute of Technology. His current research interests are social computing, Information Extraction (IE), and QA.