Learning Task-Specific Representation for Novel Words in Sequence Labeling

Minlong Peng, **Qi Zhang**, **Xiaoyu Xing**, **Tao Gui**, **Jinlan Fu** and **Xuanjing Huang**
School of Computer Science, Fudan University, Shanghai, China
{mlpeng16, qz, xyxing18, tgui16, fujl16, xjhuang}@fudan.edu.cn

Abstract

Word representation is a key component in neural-network-based sequence labeling systems. However, representations of unseen or rare words trained on the end task are usually poor for appreciable performance. This is commonly referred to as the out-of-vocabulary (OOV) problem. In this work, we address the OOV problem in sequence labeling using only training data of the task. To this end, we propose a novel method to predict representations for OOV words from their surface-forms (e.g., character sequence) and contexts. The method is specifically designed to avoid the error propagation problem suffered by existing approaches in the same paradigm. To evaluate its effectiveness, we performed extensive empirical studies on four part-of-speech tagging (POS) tasks and four named entity recognition (NER) tasks. Experimental results show that the proposed method can achieve better or competitive performance on the OOV problem compared with existing state-of-the-art methods.

1 Introduction

Word representation (or embedding) is a foundational aspect in many state-of-the-art sequence labeling systems [Ma and Hovy, 2016; Zhang and Yang, 2018]. However, natural language yields a Zipfian distribution [Zipf, 1949] over words. This means that a significant number of words (in the long tail) are rare. To control model size, a sequence labeling system often constrains the training vocabulary to only cover the top N frequent words within the training set. Those words not covered by the training vocabulary are called OOV words. Learning representations for OOV words is challenged since the standard end-to-end supervised learning methods require multiple occurrences of each word for better generalization. Many works [Ma and Hovy, 2016; Madhyastha *et al.*, 2016] have proved that performance on sequence labeling usually drops a lot when encountering OOV words. This is commonly referred to as the OOV problem, which we are to address in this work.

Over the last few years, many methods have been proposed to deal with the OOV problem. These approaches can be roughly divided into two categories: (1) pretraining word representations on very large corpora of raw text [Mikolov *et al.*, 2013; Pennington *et al.*, 2014; Peters *et al.*, 2018]; (2) further exploiting training data of the task. In a data-rich domain, the first category of methods can often bring considerable improvement to the models that are trained with random initialized word vectors [Devlin *et al.*, 2018]. However, the approach can be criticized when encounter the extremely data-hungry situation. Obtaining sufficient data may be difficult for low-resource domains, e.g. in technical domains and bio/medical domains [Delėger *et al.*, 2016].

Therefore, in this work, we highlight methods of the second category. A popular practice of this category is to represent all OOV words with a single shared embedding, which is trained on low-frequency words within the training set, and then assigned to all OOV words at testing time. However, this essentially heuristic solution is inelegant, as it conflates many words thus losing specific information of the OOV words. Another popular practice is to obtain the word representation from its surface-form (e.g., character sequence) [Ling *et al.*, 2015]. This practice is successful at capturing the semantics of morphological derivations (e.g. "running" from "run") but puts significant pressure on the encoder to capture semantic distinctions amongst syntactically similar but semantically unrelated words (e.g. "run" vs. "rung"). Therefore, most state-of-the-art sequence labeling systems will combine the surface-form representation with an unique embedding to represent the word. This again introduces the OOV problem to the systems.

Recently, a new paradigm of the second direction, which we refer to as the teacher-student paradigm, are being studied. Methods of this paradigm address the OOV problem in two steps. In the first step, they train a supervised model (also called the teacher network in this work) to perform label prediction for those within-vocabulary words to obtain their task-specific word representations. In the second step, they train (or heuristically construct) a predicting model (also called the student network) to predict the representation of a word from its surface-form [Pinter *et al.*, 2017], context [Lazaridou *et al.*, 2017], or the both [Schick and Schütze, 2018]. The training object of the student network is usually to reconstruct representations of those within-vocabulary words. At testing time, when encountering OOV words within a sentence, they first use the student network to predict

representations for those OOV words. Then, based on the generated OOV representations, they use the teacher network to perform label prediction.

Intrinsically, methods of the teacher-student paradigm can be seen as pipelines, with the teacher and student networks being two cascaded components of the pipeline. Though methods of this paradigm has achieved notable success on several tasks, they suffer from the typical error propagation problem of the pipeline paradigm [Caselli *et al.*, 2015; Bojarski *et al.*, 2016], since the auxiliary reconstruction object used for training the student network is not guaranteed to be fully compatible with the supervised object used for training the teacher network.

In this work, we propose a novel method of the teacher-student paradigm, which is specifically designed to address the error propagation problem. The main difference of this method from existing ones is the training of the student network. Instead of reconstructing representations of those within-vocabulary words, we train the student network to predict word representations that can achieve good performance on the supervised task. Training signal of the student network is directly backpropagated by the parameter-fixed teacher network. This way, we connect learning of the student network with the teacher network, thus avoiding the error propagation from the prediction of word representation using the student network to the label prediction using the teacher network. We explore applicability of this method to eight sequence labeling tasks in part-of-speech tagging (POS) and named entity recognition (NER). Empirically studies on those tasks show that our proposed method can achieve better or comparative performance on OOV words over existing methods in this paradigm.

Contributions of this work can be summarized as follows: ***(i)*** We propose a novel method of the teacher-student paradigm to address the OOV problem in sequence labeling using only training data of the task. It can avoid the error propagation problem suffered by existing approaches in the same paradigm. ***(ii)*** We performed experimental studies on eight part-of-speech tagging and named entity recognition tasks, and achieved better or comparative performance on OOV words over several existing state-of-the-art methods. Source code of this work is available at `https://github.com/v-mipeng/TaskOOV`.

2 Typical Methods of the Teacher-Student Framework

In this section, we highlight some typical methods of the teacher-student framework. The first representative work of this framework was proposed by Lazaridou et al., [2017]. In that work, they proposed to obtain the representation for an OOV word w through summation over representations of within-vocabulary words occurring in its contexts $\mathcal{C}(w)$:

$$\boldsymbol{v}_w(w) = \frac{1}{c(w)} \sum_{C \in \mathcal{C}(w)} \sum_{w' \in C \cap \mathcal{V}} \boldsymbol{e}_w(w'). \tag{1}$$

Here, $\mathcal{C}(w)$ denotes the set of contexts that contain w, $\mathcal{V}$ is the training vocabulary, $c(w) = \sum_{C \in \mathcal{C}(w)} |C \cap \mathcal{V}|$ is the total number of within-vocabulary words in $\mathcal{C}(w)$, and $\boldsymbol{e}_w(\cdot)$ is the embedding function defined on within-vocabulary words.

Following this idea, Khodak et al., [2018] further applied a linear transformation $\mathbf{A}$ to the resulting embedding:

$$\hat{\boldsymbol{v}}_w(w) = \mathbf{A}\boldsymbol{v}_w(w), \tag{2}$$

to get the representation of w. To determine the form of $\mathbf{A}$, they trained it on those within-vocabulary words with the training object being to minimize the reconstruction error of an randomly selected within-vocabulary word w:

$$\mathcal{L}(w) = d\left(\boldsymbol{e}_w(w), \hat{\boldsymbol{v}}_w(w)\right), \tag{3}$$

where $d(\cdot, \cdot)$ defines the distance between two embeddings, e.g., Euclidean distance function.

In this line, Schick and Schutze, [2018] proposed to model both the context and surface-form (subword n-grams) of OOV words to get their representations:

$$\tilde{\boldsymbol{v}}_w(w) = \alpha\hat{\boldsymbol{v}}_w(w) + (1-\alpha)\boldsymbol{v}_c(w), \tag{4}$$

where $\boldsymbol{v}_c(w)$ is a word representation modeling from the surface of w, and $\alpha \in [0, 1]$ is a single learnable parameter or a gate generated from $\hat{\boldsymbol{v}}_w(w)$ and $\boldsymbol{v}_c(w)$ as illustrated in §3.3.

In general, these methods differ from our proposed method in two points. First, they represent an OOV word w in different contexts with a consistent representation, while we relax the representation of the same word to be different in different contexts. Second, they train the student network on the auxiliary reconstruction object, while we directly train it on the supervised object used for training the teacher network. Of course, the first difference can be easily addressed by adjusting Eq. (1) to:

$$\boldsymbol{v}_w(w|C(w)) = \frac{1}{|C(w)|} \sum_{w' \in C(w) \cap \mathcal{V}} \boldsymbol{e}_w(w'), \tag{5}$$

where $C(w)$ denotes a single context of w. According to our experience, this adjustment can generally improve performance of the method in sequence labeling. Therefore, we applied this adjustment to all compared methods of the teacher-student paradigm. Thus, the main difference between our proposed method and existing works is the training strategy of the student network.

3 Methodology

Generally speaking, the method contains a teacher and a student network. The teacher network is first trained to perform the supervised task. Then, the student network (i.e., the representation predict layer) is trained to generate appropriate representations for words from their contexts and surface-forms. The quality of the generated word representation $\boldsymbol{v}(w)$ is measured by the parameter-fixed teacher network (the copied sequence modeling and CRF loss layers), i.e., the training signal of the student network is backpropagated by the teacher network. At testing time, it first uses the student network to generate representations for OOV words and then uses the teacher network to perform label prediction.

3.1 Notation

Throughout this work, we denote Ω the word space and $\mathcal{V}$ the training vocabulary. A sentence consisting of a sequence of words is denoted as $\mathbf{X} = \{w_1, \cdots, w_n\} \in \Omega$ and its corresponding label sequence is denoted as $\mathbf{Y} = \{y_1, \cdots, y_n\} \in \mathcal{Y}$. The context of a word w_i given its corresponding sentence $\mathbf{X}$ is denoted as $\mathbf{X}_{\setminus i} = \{w_1, \cdots, w_{i-1}, w_{i+1}, \cdots, w_n\}$, and $\mathbf{X}_{<i,j>}$ refers to the sub-sequence $\{w_i, \cdots, w_j\}$. The teacher network is denoted as T and the student network is denoted as S. In addition, let f be a function defined on Ω, we denote $f(\mathbf{X})$ a shorthand of $\{f(w_1), \cdots, f(w_n)\}$.

3.2 Train the Teacher Network

The teacher network is used to perform label prediction in the sequence labeling system. It can be of arbitrary architecture suitable for the task. Because the main focus of this work is to deal with the OOV problem instead of designing a superior supervised model, in this work, we tried the typical CNN-based and LSTM-based architectures [Huang *et al.*, 2015] to implement the teacher network. Briefly, this architecture represents a word w with the concatenation of an unique dense vector (embedding) $\boldsymbol{e}_w(w)$ and a vector $\boldsymbol{e}_c(w)$ modeled from its character sequence using a character-level convolutional neural network (CNN) [Kim, 2014]:

$$\boldsymbol{e}(w) = [\boldsymbol{e}_w(w) \oplus \boldsymbol{e}_c(w)], \tag{6}$$

where $\oplus$ denotes the concatenation operation. The corresponding sentence representation $\boldsymbol{e}(\mathbf{X})$ is fed as input into a bidirectional long-short term memory network (BiLSTM) [Hochreiter and Schmidhuber, 1997] or a three-layer CNN network [Kim, 2014] with kernel size set to 3 to model context information of each word, obtaining a hidden representation (not the word embedding) $\mathbf{h}(w_i|\mathbf{X})$ for each word given $\mathbf{X}$. On top of the BiLSTM-based or CNN-based sequence modeling layer, it uses a sequential conditional random field (CRF) [Lafferty *et al.*, 2001] to jointly decode labels for the whole sentence:

$$p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta}_T) = \frac{\prod_{t=1}^{m} \phi_t(y_{t-1}, y_t|\mathbf{X})}{\sum_{\mathbf{Y}' \in \mathcal{Y}} \prod_{t=1}^{m} \phi_t(y'_{t-1}, y'_t|\mathbf{X})} \tag{7}$$

where $\phi_t(y', y|\mathbf{X}) = \exp(\boldsymbol{w}_{y',y}^T \mathbf{h}(w_t) + b_{y',y})$, $\boldsymbol{w}_{y',y}$ and $b_{y',y}$ are trainable parameters corresponding to label pair (y', y), and $\boldsymbol{\theta}_T$ denotes the whole parameters of the teacher network. The training loss of the teacher network is then defined by:

$$\mathcal{L}_T = \sum_{i=1}^{N} \log p(\mathbf{Y}_i|\mathbf{X}_i; \boldsymbol{\theta}_T), \tag{8}$$

where N is the training sentence number. This network is trained on all words occurring in the training set. After that, we fix its parameters during the training of the student network described in the following.

3.3 Train the Student Network

The student network models both surface-form and context information of a word for generating its representation. These two information resources have been demonstrated to be complementary to each other by Schick and Schütze, 2018.

Model Surface-form

The surface-form representation of a word w is obtained from its character sequence $w = \{c_1, c_2, \cdots, c_n\} \in \mathcal{V}_c$, where $\mathcal{V}_c$ is the character vocabulary. Specifically, following the work of [Schick and Schütze, 2018], we first pad the character sequence with special start and end characters $c_0 = \langle s \rangle$, $c_{n+1} = \langle e \rangle$ and obtain its up k-gram set:

$$G(w) = \cup_{m=1}^{k} \cup_{i=0}^{n+2-m} \{c_i, \cdots, c_{i+m-1}\}. \tag{9}$$

Then, we define the surface-form embedding of w to be the average of all its up k-gram embeddings:

$$\boldsymbol{v}_{\text{form}}(w) = \frac{1}{|G(w)|} \sum_{g \in G(w)} \mathbf{W}_{\text{ngram}}(g), \tag{10}$$

where $\mathbf{W}_{\text{ngram}}$ denotes an embedding lookup table for n-grams, which are trainable parameters of the student network.

Model Context

For modelling the context of word w_i, we apply a bidirectional long-short term memory network (BiLSTM) on its context word sequence $\mathbf{X}_{\setminus i}$. Here, we consider application of this practice in two different situations. In the first situation, there is only one OOV word within a sentence $\mathbf{X}$. While in the second situation, there are multiple OOV words within $\mathbf{X}$.

When there is only one OOV word w_i within a sentence $\mathbf{X}$, the task-specific representations for the rest words are all known, and we can directly apply BiLSTM to $\mathbf{X}_{\setminus i}$ to predict representation for w_i. Specifically, we use the forward LSTM to model the sequence from the beginning of $\mathbf{X}$ to w_{i-1}:

$$\overrightarrow{\mathbf{h}}(w_i|\mathbf{X}) = \overrightarrow{\text{LSTM}}(\boldsymbol{e}(\mathbf{X}_{<1,i-1>})) \tag{11}$$

and use the backward LSTM to model the sequence from the end of $\mathbf{X}$ to w_{i+1} in a reverse order:

$$\overleftarrow{\mathbf{h}}(w_i|\mathbf{X}) = \overleftarrow{\text{LSTM}}(\boldsymbol{e}(\mathbf{X}_{<i+1,n>})). \tag{12}$$

The forward and backward hidden representations are concatenated to form the context representation of w_i given $\mathbf{X}$:

$$\boldsymbol{v}_{\text{context}}(w_i|\mathbf{X}) = [\overrightarrow{\mathbf{h}}(w_i|\mathbf{X}) \oplus \overleftarrow{\mathbf{h}}(w_i|\mathbf{X})], \tag{13}$$

where $\oplus$ denotes the concatenation operation.

When there are multiple OOV words within a sentence, for a given OOV word $w_i \in \mathbf{X}$, the representations of some other words in $\mathbf{X}_{\setminus i}$ are also unknown. Therefore, we cannot directly apply the BiLSTM to $\mathbf{X}_{\setminus i}$ to get its predicted representation. To address this problem, we propose to iteratively predict representations for the multiple OOV words within a sentence. Let $\boldsymbol{v}^t_{\text{context}}(w_i|\mathbf{X})$ denote the predicted representation of w_i given $\mathbf{X}$ at the t^{th} iteration with $\boldsymbol{v}^0_{\text{context}}(w_i|\mathbf{X}) = \mathbf{0}$. At the $(t+1)^{th}$ iteration, for predicting the representation of w_i, we apply the BiLSTM to its t^{th} iteration context $\boldsymbol{v}^t_{\text{context}}(\mathbf{X}_{\setminus i})$, obtaining $\boldsymbol{v}^{t+1}_{\text{context}}(w_i|\mathbf{X})$. This process repeats for all OOV words to finish the $(t+1)^{th}$ iteration. The iteration proceeds till t reaches a fix value K, obtaining the final predicted representation for each OOV word. According to our experience, it is appropriate to set $K = 2$.

Combine Surface-form and Context.
We finally combine representations of the surface-form and the context to obtain a joint representation $\boldsymbol{v}(w|\mathbf{X})$ of w. In this work, we follow the idea of [Schick and Schütze, 2018] and combine these two representations with a gate:

$$\boldsymbol{v}_w(w|\mathbf{X}) = \alpha \boldsymbol{v}_{\text{form}}(w) + (1-\alpha)\boldsymbol{v}_{\text{context}}(w|\mathbf{X}) \quad (14)$$

where

$$\alpha = \sigma(\boldsymbol{w}^T [\boldsymbol{v}_{\text{form}}(w) \oplus \boldsymbol{v}_{\text{context}}(w|\mathbf{X})] + b). \quad (15)$$

Here, σ denotes the sigmoid function. For expression consistency, we denote

$$\boldsymbol{v}_w(w|\mathbf{X};\boldsymbol{\theta}_S) = \boldsymbol{v}_w(w|\mathbf{X}), \quad (16)$$

where $\boldsymbol{\theta}_S$ are those trainable parameters of the student network.

Training
Because we use the teacher network to perform label prediction, training object of the student network should be learning to generate representations for OOV words that are suitable for the teacher network. According to the idea of previous works [Lazaridou *et al.*, 2017; Khodak *et al.*, 2018; Schick and Schütze, 2018], we may train the student network to minimize the reconstruction errors of those within-vocabulary words with loss function defined by:

$$\mathcal{L}_{recon} = \frac{1}{N}\sum_{i=1}^{N}\sum_{w\in\mathbf{X}_i\cap\mathcal{V}} d\left(\boldsymbol{v}_w(w|\mathbf{X}_i;\boldsymbol{\theta}_S), \boldsymbol{e}_w(w)\right). \quad (17)$$

Here, $d(\cdot,\cdot)$ defines a distance between two embeddings, e.g., Euclidean Distance function. The biggest problem of this practice is that it may suffer from the error propagation problem, since the auxiliary reconstruction training criteria $\mathcal{L}_{recon}$ is not guaranteed to be compatible with training object of the teacher network.

In this work, we propose to directly connect training of the student network with the training object of the teacher network, sidestepping designing an appropriate auxiliary training criteria. Without loss of generality, we first consider training of the student network in the situation that there is only one OOV word within a given sentence. To simulate this situation, for a training sentence $\mathbf{X}$ and its corresponding label sequence $\mathbf{Y}$, we randomly sample a word w_i from $\mathbf{X}$. The resultant pair $(w_i, \mathbf{X}_{\setminus i}, \mathbf{Y})$ forms a training example of the student network. By replacing the i^{th} column of $\boldsymbol{e}(\mathbf{X})$ with $\boldsymbol{v}(w|\mathbf{X};\boldsymbol{\theta}_S) = [\boldsymbol{v}_w(w|\mathbf{X};\boldsymbol{\theta}_S) \oplus \boldsymbol{e}_c(w)]$, we obtain the input $\boldsymbol{v}(\mathbf{X}) = \{\boldsymbol{e}(w_1), \cdots, \boldsymbol{v}(w_i|\mathbf{X};\boldsymbol{\theta}_S), \cdots, \boldsymbol{e}(w_n)\}$ of the teacher network. Based on $\boldsymbol{v}(\mathbf{X})$, we obtain new hidden representation for each word $\hat{\mathbf{h}}(w_i|\mathbf{X})$ and the task loss is then defined by:

$$\mathcal{L}_S(\mathbf{X},\mathbf{Y};\boldsymbol{\theta}_S,\boldsymbol{\theta}_T) = \log\frac{\prod_{t=1}^{m}\phi_t(y_{t-1},y_t|\mathbf{X})}{\sum_{\mathbf{Y}'\in\mathcal{Y}}\prod_{t=1}^{m}\phi_t(y'_{t-1},y'_t|\mathbf{X})}, \quad (18)$$

where, this time, $\phi_t(y',y|\mathbf{X}) = \exp(\boldsymbol{w}^T_{y',y}\hat{\mathbf{h}}(w_t) + b_{y',y})$. Note that the training loss of the student network is the same as that of the teacher network, making sure that their training is compatible. For training the student network, we perform parameter update by:

$$\boldsymbol{\theta}_S \leftarrow \boldsymbol{\theta}_S - \alpha\frac{\partial\mathcal{L}_S(\mathbf{X},\mathbf{Y};\boldsymbol{\theta}_S,\boldsymbol{\theta}_T)}{\partial\boldsymbol{\theta}_S}. \quad (19)$$

During the training of the student network, parameters of the teacher network $\boldsymbol{\theta}_T$ is fixed. Algorithm 1 shows the general training process of the student network in the situation that there is only one OOV word within a sentence.

To extend this process to the multiple OOV word situation, we sample multiple words from $\mathbf{X}$, e.g., w_i and w_j, generating an training example of the student network. The loss defined on this example is similar to that in the one OOV situation and minimized over $\boldsymbol{\theta}_S$.

Algorithm 1 Training of the student network

Input: the teacher network T, training dataset $\mathcal{D}$
Result: the student network S
while S does not converge **do**
  sample $\mathbf{X} = \{w_1, \cdots, w_n\}$ and its corresponding $\mathbf{Y} = \{y_1, \cdots, y_n\}$ from $\mathcal{D}$
  for $i \in [1, \cdots, n]$ **do**
    Generate $\boldsymbol{v}(w_i; X)$ for $w_i \in \mathbf{X}$;
    $\boldsymbol{v}(\mathbf{X}) = \{\boldsymbol{e}(w_1), \cdots, \boldsymbol{v}(w_i|\mathbf{X}), \cdots, \boldsymbol{e}(w_n)\}$;
    Get $\mathcal{L}_S(\mathbf{X},\mathbf{Y};\boldsymbol{\theta}_S,\boldsymbol{\theta}_T)$ based on $\boldsymbol{v}(\mathbf{X})$ according to Eq. (18);
    Update $\boldsymbol{\theta}_S \leftarrow \boldsymbol{\theta}_S - \alpha\frac{\partial\mathcal{L}_S(\mathbf{X},\mathbf{Y};\boldsymbol{\theta}_S,\boldsymbol{\theta}_T)}{\partial\boldsymbol{\theta}_S}$.

4 Experiments

To evaluate the effectiveness of our proposed method, we performed experiments on four part-of-speech tagging (POS) tasks and four named entity recognition (NER) tasks. These tasks have varying OOV rates, which is defined by the percentage of testing words occurring less than five times in the training set. These tasks share the same architectures of the teacher and student network as illustrated in §3.

4.1 Datasets

POS: For POS, we conducted experiments on: (**1**) PTB-English: the Wall Street Journal portion of the English Penn Treebank dataset [Marcus *et al.*, 1993], (**2**) RIT-English: a dataset created from Tweets in English [Derczynski *et al.*, 2013], (**3**) GSD-Russian: the Russian Universal Dependencies Treebank annotated and converted by Google[1], and (**4**) RRT-Romanian: the Romanian UD treebank (called RoRefTrees) [Verginica Barbu Mititelu, 2016]. For PTB-English, we followed the standard splits: sections 2-21 for training, section 22 for validation, and section 23 for testing. For RIT-English we followed the split protocol of Gui et al., [2017]. While, for UD-Russian and UD-Romanian, we used their given data splits.

NER: For NER, we performed experiments on: (**1**) CoNLL02-Spanish: the CoNLL2002 Spanish NER Shared Task dataset [Sang, 2002]; (**2**) CoNLL02-Dutch: the

[1]https://universaldependencies.org/

Dataset	Dev		Test	
	#OOV	OOV Rate	#OOV	OOV Rate
POS				
PTB-English	8,392	6.37%	7,528	5.81%
RIT-English	774	34.52%	760	33.17%
GSD-Russian	21,323	17.96%	21,523	18.31%
RRT-Romanian	3,965	23.22%	3,702	22.67%
NER				
CoNLL02-Spanish	2,216	50.91%	1,544	43.38%
CoNLL02-Dutch	1,819	69.53%	2,564	65.05%
Twitter-English	1,266	79.15%	4,131	79.13%
CoNLL03-German	3,928	81.27%	2,685	73.10%

Table 1: Number of OOV words (for POS) and entities (for NER) in the development and testing sets, when treating words occurring less than 5 times in the training set as OOV. An entity is treated as OOV if it contains at least one OOV word.

CoNLL2002 dataset of Dutch language; (**3**) Twitter-English: an English NER dataset created from Tweets [Zhang *et al.*, 2018]; and (**4**) CoNLL03-German: the CoNLL2003 NER dataset in German [Tjong Kim Sang and De Meulder, 2003]. These datasets are annotated by four types: PER, LOC, ORG and MISC. For datasets except Twitter-English, we used the official split training set for model training, testa for validating and testb for testing. While for Twitter-English, we followed data splits of Zhang et al., [2018].

Table 1 reports the statistic results of the OOV problem on the development and testing sets of each dataset. From the table, we can see that the OOV rate varies a lot over different datasets.

4.2 Compared Methods

- **RandomUNK**: This baseline refers to the teacher network trained on all words occurring in the training set. At testing time, it represents words not occurring in the training set with a consistent random vector.
- **SingleUNK**: This baseline trains the teacher network on words that occur no less than 5 times in the training set. The other infrequent words and those words not occurring in the training set are all mapped to a single trainable embedding e_{UNK}, which is trained during model training.
- [Lazaridou *et al.*, 2017]: This baseline uses RandomUNK as the teacher network. At testing time, it first represents OOV words with their context word representations by mean-pooling as defined in Eq. (5) based on RandomUNK and then performs label prediction using RandomUNK.
- [Khodak *et al.*, 2018]: This baseline additionally trains a linear transformer $\mathbf{A}$ to transform the mean-pooled representation of [Lazaridou *et al.*, 2017] as illustrated by Eq. (2) for predicting representations of OOV words. The training of $\mathbf{A}$ is performed on words that occur no less than 5 times in the training set.
- [Schick and Schütze, 2018]: This baseline is a variant of the proposed method, but its student network is trained on the reconstruction object as defined in Eq. (17). It is also related to the work of [Schick and Schütze, 2018] but using the same architecture of the proposed method to model word context and surface-form.
- [Akbik *et al.*, 2018]: This baseline converts the input sentence into a character sequence. Then, it applies a character language model to the character sequence to get the representation of every word within the sentence. Based on the obtained word representations, it applies a LSTM network to model the word sequence and performs final tag recommendation. To make it compatible with the setting of this work, we did not pre-train the language model on external data and not use pre-trained static word embeddings for this baseline.

4.3 Implementation Detail

For data prepossessing, all digits were replaced with the special token "<NUM>", and all url were replaced with the special token "<URL>". Dimension of word embedding, character embedding, and LSTM were respectively set to 50, 16, and 50 for both the teacher and student networks. Kernel size of the character CNN was set to 25 for kernel width 3 and 5. Optimization was performed using the Adam step rule [Kinga and Adam, 2015] with the learning rate set to 1e-3.

4.4 Evaluation

We partitioned the testing (or development) set into two subsets: within-vocabulary (WIV) words and out-of-vocabulary (OOV) words. A word is considered WIV if it occurs more than 5 times in the training set, otherwise OOV. For NER, an entity is considered being of OOV if it contains at least one word of OOV word set. We report model performance (accuracy for POS and F1 for NER) on OOV set. This is because we can exactly tell which subset a word belongs to, thus we can easily combine the best model on each subset to achieve the best overall performance on the whole testing set. For example, we can perform label prediction for OOV words using our proposed model, and perform label prediction for WIV words using the best performing model on the WIV subset.

4.5 Main Results

Table 2 and 3 reports model performance on the OOV set for POS-tagging and NER, respectively, when using BiLSTM and CNN to implement the teacher network. From this table, we have the following observations: (1) on most tasks, methods dealing with the OOV problem outperform the RandomUNK baseline. This verifies the necessity to deal with the OOV problem in sequence labeling. (2) the method [Schick and Schütze, 2018] using both surface-form and context information to generate representations of OOV words outperforms the method [Khodak *et al.*, 2018] using only context information on most datasets. This shows the complementary of surface-form and context information; (3) the most comparative baseline in the teacher-student paradigm [Schick and Schütze, 2018] in general outperforms

Arch	Model	PTB-English		RIT-English		GSD-Russian		RRT-Romanian	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
LSTM	RandomUNK	85.25	85.95	63.07	61.05	82.06	80.97	86.73	87.36
	SingleUNK	86.90	88.78	61.37	61.97	85.22	84.68	90.11	89.03
	[Lazaridou *et al.*, 2017]	83.90	85.67	63.82	63.16	85.22	84.62	89.53	90.14
	[Khodak *et al.*, 2018]	84.03	85.67	64.21	64.07	86.23	85.41	89.94	90.11
	[Schick and Schütze, 2018]	87.62	89.37	64.73	62.24	86.35	85.49	90.34	90.01
	[Akbik *et al.*, 2018]	87.82	88.90	58.01	59.60	83.69	83.93	88.75	89.03
	Proposed	**88.68**	**90.53**	**66.54**	**64.87**	**87.28**	**86.47**	**91.64**	**90.28**
CNN	RandomUNK	88.10	88.54	61.88	63.42	85.28	89.87	88.31	89.32
	SingleUNK	87.16	88.84	60.85	59.34	85.28	86.17	87.59	87.16
	[Lazaridou *et al.*, 2017]	89.75	90.74	61.49	63.02	88.86	90.06	89.43	89.57
	[Khodak *et al.*, 2018]	89.89	90.84	61.88	63.42	89.18	90.34	90.01	90.08
	[Schick and Schütze, 2018]	89.10	90.61	62.79	62.63	89.13	90.08	90.51	90.43
	Proposed	**91.33**	**91.74**	**65.82**	**65.27**	**90.64**	**91.52**	**91.74**	**91.68**

Table 2: Model performance on the OOV set for part-of-speech tagging when implementing the sequence modeling layer of the teacher network with LSTM-based and CNN-based architectures.

Arch	Model	CoNLL02-Spanish		CoNLL02-Dutch		Twitter-English		CoNLL03-German	
		Dev	Test	Dev	Test	Dev	Test	Dev	Test
LSTM	RandomUNK	69.36	72.06	64.23	64.08	56.88	56.38	55.92	56.89
	SingleUNK	68.79	71.59	67.83	66.39	56.82	56.39	59.69	60.16
	[Lazaridou *et al.*, 2017]	68.61	69.08	65.99	65.43	47.72	47.20	47.87	49.17
	[Khodak *et al.*, 2018]	68.74	69.53	66.34	65.70	48.22	47.28	47.97	49.33
	[Schick and Schütze, 2018]	70.84	72.88	68.88	67.51	59.18	57.21	55.83	58.42
	[Akbik *et al.*, 2018]	61.78	64.06	60.49	62.09	49.68	50.22	55.06	53.01
	Proposed	**73.91**	**74.63**	**70.33**	**70.12**	**60.14**	**58.32**	**60.55**	**61.79**
CNN	RandomUNK	61.07	61.61	53.48	57.31	44.82	43.72	56.23	56.94
	SingleUNK	56.87	58.30	**60.68**	**60.46**	**57.13**	**57.34**	62.33	62.07
	[Lazaridou *et al.*, 2017]	54.97	60.39	53.73	56.99	42.91	46.99	43.38	43.54
	[Khodak *et al.*, 2018]	55.12	60.41	54.20	57.00	48.21	47.78	53.19	53.50
	[Schick and Schütze, 2018]	61.23	61.54	53.60	57.48	46.79	46.59	56.16	57.44
	Proposed	**63.38**	**63.02**	59.24	60.33	57.06	57.32	**62.42**	**63.01**

Table 3: Model performance on the OOV set for named entity recognition.

SingleUNK. This verifies the effectiveness of the motivation beneath the teacher-student paradigm; (4) our proposed method consistently outperforms [Schick and Schütze, 2018], which differs from the proposed method in the training of the student network. This, on one hand, shows the existence of error propagation from the student network to the teacher network, and on the other hand approves the effectiveness of our proposed method for addressing OOV problem; finally (5) our proposed method consistently outperforms the character language model [Akbik *et al.*, 2018]. A possible explanation of this result is that our method can use word-level embedding without suffering from the OOV problem while the character language model cannot.

5 Conclusion

In this work, we proposed a novel method to address the out-of-vocabulary problem in sequence labeling systems using only training data of the task. It is designed to generate representations for OOV words from their surface-forms and contexts. Moreover, it is designed to avoid the error propagation problem suffered by existing methods in the same paradigm. Extensive experimental studies on POS-tagging (POS) and named entity recognition (NER) show that this method can achieve superior or comparable performance over existing methods on the OOV problem.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No. 2018YFC0831105, 2017YFB1002104, 2018YFB1005104), National Natural Science Foundation of China (No. 61532011, 61751201), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), STCSM (No.16JC1420401,17JC1420200), ZJLab.

References

[Akbik *et al.*, 2018] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[Bojarski *et al.*, 2016] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[Caselli *et al.*, 2015] Tommaso Caselli, Piek Vossen, Marieke van Erp, Antske Fokkens, Filip Ilievski, Rubén Izquierdo, Minh Le, Roser Morante, and Marten Postma. When it's all piling up: investigating error propagation in an nlp pipeline. In *WNACP@ NLDB*, 2015.

[Delėger *et al.*, 2016] Louise Delėger, Robert Bossy, Estelle Chaix, Mouhamadou Ba, Arnaud Ferrė, Philippe Bessieres, and Claire Nėdellec. Overview of the bacteria biotope task at bionlp shared task 2016. In *Proceedings of the 4th BioNLP Shared Task Workshop*, pages 12–22, 2016.

[Derczynski *et al.*, 2013] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206, 2013.

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

[Gui *et al.*, 2017] Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. Part-of-speech tagging for twitter with adversarial neural networks. In *EMNLP*, pages 2411–2420, 2017.

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

[Huang *et al.*, 2015] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[Khodak *et al.*, 2018] Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *ACL (Long Papers)*, volume 1, pages 12–22, 2018.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *EMNLP*, 2014.

[Kinga and Adam, 2015] D Kinga and J Ba Adam. A method for stochastic optimization. In *ICLR*, volume 5, 2015.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[Lazaridou *et al.*, 2017] Angeliki Lazaridou, Marco Marelli, and Marco Baroni. Multimodal word meaning induction from minimal exposure to natural text. *Cognitive science*, 41:677–705, 2017.

[Ling *et al.*, 2015] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*, pages 1520–1530, 2015.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, volume 1, pages 1064–1074, 2016.

[Madhyastha *et al.*, 2016] Pranava Swaroop Madhyastha, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Mapping unseen words to task-trained embedding spaces. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 100–110, 2016.

[Marcus *et al.*, 1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Peters *et al.*, 2018] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[Pinter *et al.*, 2017] Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. Mimicking word embeddings using subword rnns. In *EMNLP*, pages 102–112, 2017.

[Sang, 2002] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *Computer Science*, pages 142–147, 2002.

[Schick and Schütze, 2018] Timo Schick and Hinrich Schütze. Learning semantic representations for novel words: Leveraging both form and context. *arXiv preprint arXiv:1811.03866*, 2018.

[Tjong Kim Sang and De Meulder, 2003] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[Verginica Barbu Mititelu, 2016] Cenel-Augusto Perez Radu Ion Radu Simionescu Martin Popel Verginica Barbu Mititelu, Elena Irimia. The romanian treebank annotated according to universal dependencies. In *Proceedings of The Tenth International Conference on Natural Language Processing (HrTAL2016)*, 2016.

[Zhang and Yang, 2018] Yue Zhang and Jie Yang. Chinese ner using lattice lstm. In *ACL*, volume 1, pages 1554–1564, 2018.

[Zhang *et al.*, 2018] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. Adaptive co-attention network for named entity recognition in tweets. In *AAAI*, 2018.

[Zipf, 1949] George Kingsley Zipf. Human behavior and the principle of least effort. 1949.