Handling Rare Entities for Neural Sequence Labeling

Yangming Li[♠,♡], **Han Li**[♡], **Kaisheng Yao**[♠] and **Xiaolong Li**[♠]
[♠]Ant Financial Services Group, Alibaba Group
[♡]Harbin Institute of Technology
{pangmao.lym,kaisheng.yao,xl.li}@antfin.com

Abstract

One great challenge in neural sequence labeling is the *data sparsity problem* for rare entity words and phrases. Most of test set entities appear only few times and are even unseen in training corpus, yielding large number of out-of-vocabulary (OOV) and low-frequency (LF) entities during evaluation. In this work, we propose approaches to address this problem. For OOV entities, we introduce *local context reconstruction* to implicitly incorporate contextual information into their representations. For LF entities, we present *delexicalized entity identification* to explicitly extract their frequency-agnostic and entity-type-specific representations. Extensive experiments on multiple benchmark datasets show that our model has significantly outperformed all previous methods and achieved new start-of-the-art results. Notably, our methods surpass the model fine-tuned on pre-trained language models without external resource.

1 Introduction

In the context of natural language processing (NLP), the goal of sequence labeling is to assign a categorical label to each entity word or phrase in a text sequence. It is a fundamental area that underlies a range of applications including slot filling and named entity recognition. Traditional methods use statistical models. Recent approaches have been based on neural networks (Collobert et al., 2011; Mesnil et al., 2014; Ma and Hovy, 2016; Strubell et al., 2017; Li et al., 2018; Devlin et al., 2018; Liu et al., 2019a; Luo et al., 2020; Xin et al., 2018) and they have made great progresses in various sequence labeling tasks.

However, a great challenge to neural-network-based approaches is from the *data sparsity problem* (Augenstein et al., 2017). Specifically in the context of sequence labeling, the majority of entities in test dataset may occur in training corpus few times or are absent at all. In this paper, we refer this phenomenon particularly to *rare entity problem*. It is different from other types of data sparsity problems such as the lack of training data for low-resource language (Lin et al., 2018), as this rare entity problem is more related to a mismatch of entity distributions between training and test, rather than the size of training data. We present an example of the problem in Table 1. It shows that less than 5% of test set entities are frequently observed in the training set, and about 65% of test set entities are absent from the training set.

Frequency	Number	Percentage
$= 0$ (OOV)	1611	65.1%
$= 1$ (Low)	191	7.7%
< 10 (Low)	635	25.7%
> 20 (High)	117	4.7%
≥ 0 (Total)	2475	100.0%

Table 1: Number of occurrences of test set entities in the training set. OOV entities are those that have no occurrence (Frequency = 0) in the training set. Low frequency entities are those with fewer than ten occurrences (Frequency < 10). Percentages of entity occurrences are also shown. Data source is CoNLL-03.

The rare entities can be categorized into two types: *out-of-vocabulary* (OOV) for those test set entities that are not observed in the training set, and *low frequency* (LF) for those entities with low frequency (e.g., fewer than 10) occurrences in the training set. Without proper processing, rare entities can incur the following risks when building a neural network. Firstly, OOV terms may act as noise for inference, as they lack lexical information from training set (Bazzi, 2002). Secondly, it is hard to obtain high-quality representations on LF entities (Gong et al., 2018). Lastly, high occurrences of OOV and LF entities expose distribution discrep-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6441–6451
July 5 - 10, 2020.

ancy between training and test, which mostly leads to poor performances during test.

In general, there are two existing strategies attempting to mitigate the above issues: *external resource* and *transfer learning*. The external resource approach, for example (Huang et al., 2015; Li et al., 2018), uses external knowledge such as part-of-speech tags for NER or additional information from intent detection for slot filling. However, external knowledge such as part-of-speech tag is not always available for practical applications and open source taggers such as (Manning et al., 2014) may perform poorly for cross-domain annotations. Character or n-gram feature are mainly designed to deal with morphologically similar OOV words. The transfer learning approach, such as using ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018), fine-tunes pre-trained models on the downstream task (Liu et al., 2019a). Nevertheless, it is not directly addressing problems such as entity distribution discrepancy between training and test. Moreover, our proposed methods surpass these methods without resorting to external resources nor large pre-trained language models.

This paper proposes novel techniques that enable sequence labeling models to achieve state-of-the-art performances without using external resource nor transfer learning. These are

- *local context reconstruction* (LCR), which is applied on OOV entities, and
- *delexicalized entity identification* (DEI), which is applied on LF entities.

Local context reconstruction enables OOV entities to be related to their contexts. One key point is applying variational autoencoder to model this reconstruction process that is typically a one-to-many generation process. Delexicalized entity identification aims at extracting frequency-agnostic and entity-type-specific representation, therefore reducing the reliance on high-frequency occurrence of entities[1]. It uses a novel adversarial training technique to achieve this goal. Both methods use an effective random entity masking strategy.

We evaluate the methods on sequence labeling tasks on several benchmark datasets. Extensive experiments show that the proposed methods significantly outperform previous models by a large margin. Detailed analysis indicates that the proposed methods indeed alleviate the rare entity problem. Notably, without using any external knowledge nor pre-trained models, the proposed methods surpass the model that uses fine-tuned BERT.

[1]This paper refers slots in slot filling tasks as entities for brevity, although their definitions are not equivalent.

2 Background

Given an input sequence $X = [x_1, x_2, \cdots, x_N]$ with N tokens, the sequence labeling task aims at learning a functional mapping to obtain a target label sequence $Y = [y_1, y_2, \cdots, y_N]$ with equal length. In the following, we briefly introduce a typical method for sequence labeling and review related techniques we use in deriving our model.

2.1 Bidirectional RNN + CRF

Recurrent neural network (RNN) (Hochreiter and Schmidhuber, 1997) has been widely used for sequence labeling. The majority of high performance models use bidirectional RNN (Schuster and Paliwal, 1997) to encode input sequence X and conditional random field (CRF) (Lafferty et al., 2001) as a decoder to output Y.

The bidirectional RNN firstly embeds observation x_i at each position i to a continuous space $\mathbf{x}_i$. It then applies forward and backward operations on the whole sequence time-recursively as

$$\begin{cases} \overrightarrow{\mathbf{h}}_i = \overrightarrow{f}(\mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i-1}) \\ \overleftarrow{\mathbf{h}}_i = \overleftarrow{f}(\mathbf{x}_i, \overleftarrow{\mathbf{h}}_{i+1}) \end{cases}. \quad (1)$$

CRF computes the probability of a label sequence Y given X as

$$\begin{cases} \log p(Y|X) \propto \sum_i (\mathbf{g}_i[y_i] + \mathbf{G}[y_i, y_{i+1}]) \\ \mathbf{g}_i = \mathbf{W} * (\overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i) \end{cases}, \quad (2)$$

where $\oplus$ denotes concatenation operation. $\mathbf{G}$ and $\mathbf{W}$ are learnable matrices. The sequence with the maximum score is the output of the model, typically obtained using Viterbi algorithm.

We use bidirectional RNN + CRF model, in particular, Bi-LSTM+CRF (Huang et al., 2015), as the baseline model in our framework and it is referred in the bottom part of Figure 1.

2.2 Variational Autoencoder

The above model, together with other encoder-decoder models (Sutskever et al., 2014; Bahdanau et al., 2014), learn deterministic and discriminative functional mappings. The variational auto-encoder (VAE) (Kingma and Welling, 2015; Rezende et al.,

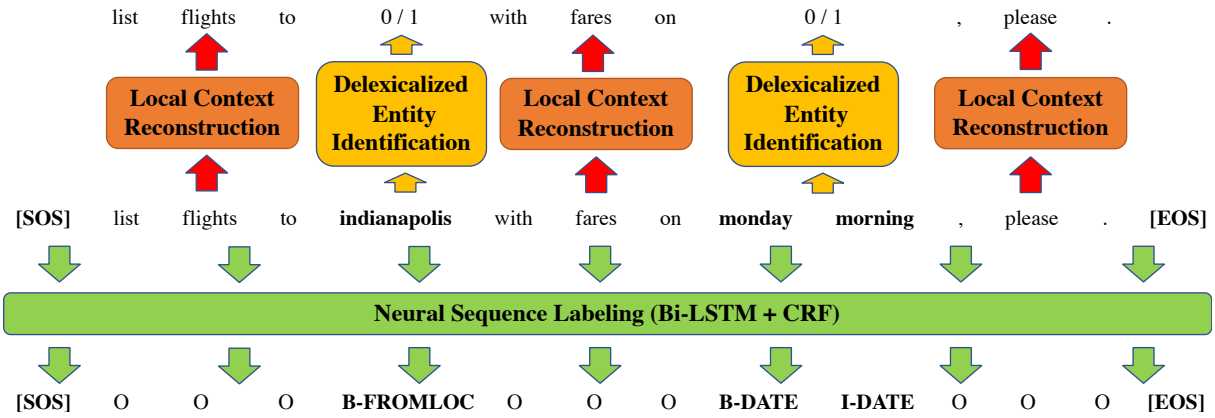


Figure 1: Overall framework to use local context reconstruction and delexicalized entity identification for neural sequence labeling. "[SOS]" and "[EOS]" are used for marking sequence begining and sequence ending, respectively. The local context reconstruction is applied between any two sucessive entities, including the special entities. The delexicalized entity identitification is applied for all entities except for the special entities.

2014; Bowman et al., 2015), on the other hand, is stochastic and generative.

Using VAE, we may assume a sequence $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ is generated stochastically from a latent global variable $\mathbf{z}$ with a joint probability of

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}) * p(\mathbf{z}). \quad (3)$$

where $p(\mathbf{z})$ is the prior probability of $\mathbf{z}$, generally a simple Gaussian distribution, to keep the model from generating $\mathbf{x}$ deterministically. $p(\mathbf{x}|\mathbf{z})$ represents a generation density, usually modeled with a conditional language model with initial state of $\mathbf{z}$.

Maximum likelihood training of a model for Eq. (3) involves computationally intractable integration of $\mathbf{z}$. To circumvent this, VAE uses variational inference with variational distribution of $\mathbf{z}$ coming from a Gaussian density $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$, with vector mean μ and diagonal matrix variance $\text{diag}(\sigma^2)$ parameterized by neural networks. VAE also uses reparameterization trick to obtain latent variable $\mathbf{z}$ as follows:

$$\mathbf{z} = \mu + \sigma \odot \epsilon, \quad (4)$$

where ϵ is sampled from standard Gaussian distribution and $\odot$ denotes element-wise product.

The evidence lower bound (ELBO) of the likelihood $p(\mathbf{x})$ is obtained using Jensen's inequality $E_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}, \mathbf{z}) \leq \log p(\mathbf{x})$ as follows:

$$\begin{aligned} \mathcal{L}^{\text{vae}}(\mathbf{x}) = &-\text{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \\ &-\text{CE}(q(\mathbf{z}|\mathbf{x})|p(\mathbf{x}|\mathbf{z})), \end{aligned} \quad (5)$$

where $\text{KL}(q||p)$ and $\text{CE}(q|p)$ respectively denote the Kullback-Leibler divergence and the cross-entropy between distribution q and p. ELBO can be optimized by alternating between optimizations of parameters of $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x}|\mathbf{z})$.

We apply VAE for local context reconstruction from slot/entity tags in Figure 1. This is a generation process that is inherently one-to-many. We observe that VAE is superior to the deterministic model (Bahdanau et al., 2014) in learning representations of rare entities.

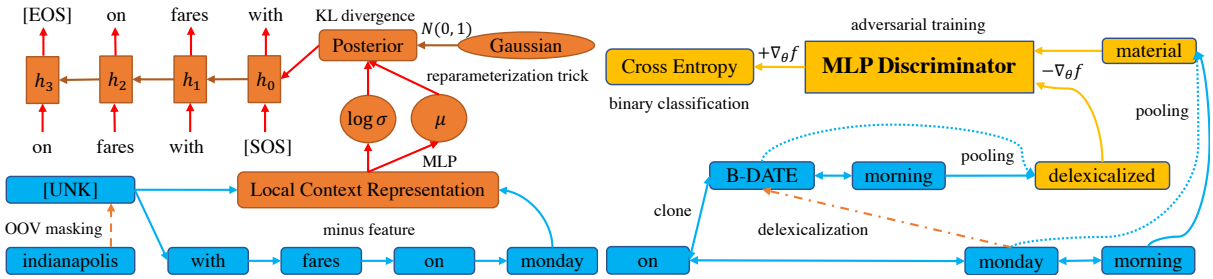
2.3 Adversarial Training

Adversarial training (Goodfellow et al., 2014), originally proposed to improve robustness to noise in image, is later extended to NLP tasks such as text classification (Miyato et al., 2015, 2016) and learning word representation (Gong et al., 2018).

We apply adversarial training to learn better representations of low frequency entities via delexicalized entity identification in Figure 1. It has a discriminator to differentiate representations from the original low-frequency entities and the representations of the delexicalized entities. Training aims at obtaining representations that can fool the discriminator, therefore achieving frequency-agnostics and entity-type-specificity.

3 The Model

We illustrate the overall framework of the proposed model in Figure 1. Its baseline sequence labeling module is described in Section 2.1. We describe the details of local context reconstruction in Sec. 3.1 and delexicalized entity identification in Sec. 3.2, together with an example to illustrate them in Figure 2. We denote parameters in Sec. 2.1 as θ^{rnn} and θ^{emb}, respectively, for its RNN and matrix to obtain



(a) VAE for local context reconstruction. (b) AT for delexicalized entity identification.

Figure 2: An example to illustrate local context reconstruction and delexicalized entity identification.

embedding. Parameters in Sec. 3.1 and Sec. 3.2 are each denoted as θ^{lcr} and θ^{D}.

3.1 Local Context Reconstruction

Contrary to the conventional methods that explicitly provide abundant lexical features from external knowledge, we implicitly enrich word representations with contextual information by training them to reconstruct their local contexts.

Masking Every entity word x_i in X, which is defined to be not associated with non-entity label "O", in sequence X is firstly randomly masked with OOV symbol "[UNK]" as follows:

$$x_i^u = \begin{cases} \text{"[UNK]"} & \text{if } y_i \neq \text{"O"} \cap \epsilon > p \\ x_i & \text{otherwise} \end{cases}, \quad (6)$$

where constant p is a threshold and ϵ is uniformly sampled between 0 and 1.

Forward Reconstruction In the forward reconstruction process, the forward pass of Eq. (1) is firstly applied on sequence $X^u = [x_1^u, x_2^u, \cdots, x_N^u]$ to obtain hidden states $\overrightarrow{h}_i^u$. Then, a forward span representation, $\mathbf{m}_{jk}^f$, of the local context between position k and j is obtained using RNN-minus feature (Wang and Chang, 2016) as follows:

$$\mathbf{m}_{jk}^f = \overrightarrow{\mathbf{h}}_k^u - \overrightarrow{\mathbf{h}}_j^u. \quad (7)$$

To apply VAE to reconstruct the local context, the mean μ and log-variance $\log\sigma$ are firstly computed from the above representation as follows:

$$\begin{cases} \mu_{jk}^f = \mathbf{W}_1^\mu \tanh(\mathbf{W}_0^\mu \mathbf{m}_{jk}^f) \\ \log\sigma_{jk}^f = \mathbf{W}_1^\sigma \tanh(\mathbf{W}_0^\sigma \mathbf{m}_{jk}^f) \end{cases}, \quad (8)$$

where $\mathbf{W}_*^*$ are all learnable matrices. Then, the reparameterization trick in Eq. (4) is applied on μ_{jk}^f and $\sigma_{jk}^f = \exp(\log\sigma_{jk}^f)$ to obtain a global latent variable $\mathbf{z}_{jk}^f$ for the local context.

To generate the i-th word in the local context sequence $[x_{j+1}, x_{j+2}, \cdots, x_{k-1}]$, we first apply a RNN-decoder with its initial hidden state from the latent variable $\mathbf{z}_{jk}^f$ and the first observation from the embedding of "[SOS]" symbol to recursively obtain hidden state $\overrightarrow{\mathbf{r}}_i^f$ as follows:

$$\overrightarrow{\mathbf{r}}_i^f = \overrightarrow{f}(\mathbf{x}_i, \overrightarrow{\mathbf{r}}_{i-1}^f), \quad (9)$$

This RNN-decoder specifically does *parameter sharing* with the forward pass RNN-encoder in Eq. (1). We then use softmax to compute the distribution of word at position l as

$$\overrightarrow{P}_i^{\text{vae}} = Softmax(\mathbf{W}_g^f * \mathbf{r}_i^f), \quad (10)$$

where $\mathbf{W}_g^f$ is a learnable matrix.

Lastly, we compute KL distance and cross-entropy for length-L local context sequence in Eq. (5) as follows:

$$\begin{cases} \text{KL}_{jk}^f = \sum_d \zeta(\mu_{jk}^f[d], \sigma_{jk}^f[d]), \\ \text{CE}_{jk}^f = -\frac{1}{L}\sum_i \log(\overrightarrow{P}_i^{\text{vae}}[x_i]), \\ \overrightarrow{\mathcal{L}}_{jk}^{\text{vae}} = -\text{KL}_{jk}^f - \text{CE}_{jk}^f, \end{cases} \quad (11)$$

where d denotes hidden dimension index and the closed form KL divergence ζ is defined as

$$\zeta(\mu, \sigma) = \mu^2 + \sigma - (1 + \log\sigma). \quad (12)$$

Backward Reconstruction Same as the forward reconstruction, the backward reconstruction is applied on non-adjacent successive entities. The backward pass of Eq. (1) is firstly applied on the entity-masked sequence X^u. Once the backward span representation, $\mathbf{m}_{kj}^b$, of the local context between position k and j is obtained as $\mathbf{m}_{kj}^b = \overleftarrow{\mathbf{h}}_j^u - \overleftarrow{\mathbf{h}}_k^u$, the same procedures of the above described forward reconstruction are conducted, except using

the backward RNN-encoder $\overleftarrow{f}(\cdot)$ in lieu of the forward RNN-encoder in Eq. (9).

The objective for local context reconstruction is

$$\mathcal{J}^{\text{vae}}(X;\theta^{\text{lcr}},\theta^{\text{rnn}}) = \max_{\theta^{\text{lcr}},\theta^{\text{rnn}}} \sum_{jk} \overrightarrow{\mathcal{L}}^{\text{vae}}_{jk} + \overleftarrow{\mathcal{L}}^{\text{vae}}_{jk}, \tag{13}$$

which is to maximize the ELBO w.r.t. parameters θ^{lcr} and θ^{rnn}.

3.2 Delexicalized Entity Identification

For low-frequency entities, the delexicalized entity identification aims at obtaining frequency-agnostic and entity-type-specific representations.

Delexicalization We first randomly substitute entity words in input sequence X with their corresponding labels as

$$x_i^d = \begin{cases} y_i & \text{if } y_i \neq \text{“O”} \cap \epsilon > p \\ x_i & \text{otherwise} \end{cases}, \tag{14}$$

where p is a threshold and ϵ is uniformly sampled from $[0, 1]$. We refer this to delexicalization (Wen et al., 2015), but insert randomness in it.

Representation for Identification To obtain representation to identify whether an entity has been delexicalized to its label, we first use forward and backward RNN-encoders in Eq. (1) on the sentence $X^d = [x_1^d, x_2^d, \cdots, x_N^d]$ and obtain hidden states $\overleftarrow{\mathbf{h}}_i^d$ and $\overrightarrow{\mathbf{h}}_i^d$ for each position i. Their concatenation is $\mathbf{h}_i^d = \overleftarrow{\mathbf{h}}_i^d \oplus \overrightarrow{\mathbf{h}}_i^d$. For position i in the original sequence without delexicalization, its concatenated hidden state $\mathbf{h}_i = \overleftarrow{\mathbf{h}}_i \oplus \overrightarrow{\mathbf{h}}_i$.

For an entity with a span from position j to k, its representation $\mathbf{e}_{jk}^d$ is obtained from the following average pooling

$$\mathbf{e}_{jk}^d = \frac{1}{k-j+1} \sum_i \mathbf{h}_i^d. \tag{15}$$

Average pooling is also applied on $\mathbf{h}_i$s to obtain $\mathbf{e}_{jk}$ for the original entity with that span.

Discriminator A multi-layer perceptron (MLP) based discriminator with parameter θ^{D} is employed to output a confidence score in $[0, 1]$, indicating the probability of the delexicalization of an entity; i.e.,

$$\begin{cases} p_{jk}^d = \sigma(\mathbf{v}_d^T \tanh(\mathbf{W}_d * \mathbf{e}_{jk}^d)) \\ p_{jk} = \sigma(\mathbf{v}_d^T \tanh(\mathbf{W}_d * \mathbf{e}_{jk})) \end{cases}, \tag{16}$$

where paramters $\mathbf{v}_d$ and $\mathbf{W}_d$ are learnable and $\sigma(x)$ is Sigmoid function $\frac{1}{1+\exp(-x)}$.

Algorithm 1: Training Algorithm

Input: Dataset S, θ^{rnn}, θ^{emb}, θ^{lcr}, θ^{D}.

repeat
- Sample a minibatch with pairs (X, Y).
- Update θ^{D} by gradient descent according to Eq. (17).
- Update θ^{lcr} and θ^{rnn} by gradient ascent to joint maximization of $\mathcal{J}^{\text{vae}} + \mathcal{J}^{\text{at}}$ according to Eqs. (13) and (17).
- Update θ^{rnn} and θ^{emb} by gradient ascent according to Eq. (2).

until *Convergence*;

Output: θ^{rnn}, θ^{emb}, θ^{lcr}, θ^{D}.

Following the principle of adversarial training, we develop the following minimax objective to train RNN model θ^{rnn} and the discriminator θ^{D}:

$$\mathcal{J}^{\text{at}}(X, Y; \theta^{\text{D}}, \theta^{\text{rnn}}) = \min_{\theta^{\text{D}}} \max_{\theta^{\text{rnn}}} \sum_{jk} \log(p_{jk}) + \log(1 - p_{jk}^d), \tag{17}$$

which aims at fooling a strong discriminator θ^{D} with parameter θ^{rnn} optimized, leading to frequency-agnostics.

4 Training Algorithm

Notice that the model has three modules with their own objectives. We update their parameters jointly using Algorithm 1. The algorithm first improves discriminator θ^{D} to identify delexicalized items. It then updates θ^{lcr} and θ^{rnn} with joint optimization $\mathcal{J}^{\text{vae}}$ and $\mathcal{J}^{\text{at}}$ to improve θ^{rnn} to fool the discriminator. As VAE optimization of $\mathcal{J}^{\text{vae}}$ has posterior collapse problem, we adopt KL cost annealing strategy and word dropout techniques (Bowman et al., 2015). Finally, the algorithm updates both of θ^{rnn} and θ^{emb} in Bi-LSTM+CRF by gradient ascent according to Eq. (2). Note that θ^{lcr} shares the same parameters with θ^{rnn} and θ^{emb}.

During experiments, we also find it is beneficial to have a few epochs of pretraining of parameters θ^{rnn} and θ^{emb} with optimization of Eq. (2).

5 Experiments

This section compares the proposed model against state-of-the-art models on benchmark datasets.

5.1 Settings

Slot Filling We use available ATIS dataset (Tur et al., 2010) and SNIPS dataset (Coucke et al.,

Models		ATIS	SNIPS	CoNLL-03
Lample et al. (2016)	Bi-LSTM + CRF w/ char	95.17	93.71	90.94
Liu et al. (2018)	LM-LSTM-CRF	95.33	94.07	91.24
Liu et al. (2019a)	GCDT	95.98	95.03	91.96
Qin et al. (2019)	Stack-propagation†	95.9	94.2	-
Liu et al. (2019b)	CM-Net†	95.82	97.15	-
This Work	Bi-LSTM + CRF	95.02	93.37	90.11
	w/ external resources‡	95.67	94.76	91.04
	w/ BERT fine-tuned embedding*	95.94	96.15	92.53
	w/ Proposed Methods	**96.01**	**97.20**	**92.67**

Table 2: Sequence labeling test results of baselines and the proposed model on benchmark datasets. * refers to fine tuning on pretrained large models. † refers to using multi-task learning. ‡ refers to adopting external resources. The improvements over all prior methods are statistically significant with $p < 0.01$ under t-test.

2018). Meanwhile, we follow the same setup as (Goo et al., 2018; Qin et al., 2019).

NER We use the public CoNLL-03 dataset (Sang and Meulder, 2003) as in (Huang et al., 2015; Lample et al., 2016; Liu et al., 2019a). The dataset is tagged with four named entity types, including PER, LOC, ORG, and MISC.

Baselines We compare the proposed model with five types of methods: 1) strong baseline (Lample et al., 2016) use character embedding to improve sequence tagger; 2) recent state-of-the-art models for slot filling (Qin et al., 2019; Liu et al., 2019b) that utilize multi-task learning to incorporate additional information from intent detection; 3) recent state-of-the-art models, including Liu et al. (2018) and Liu et al. (2019a), for NER; 4) Bi-LSTM + CRF model augmented with external resources, (i.e., POS tagging using Stanford Parser[2]); and 5) Bi-LSTM + CRF model with word embedding from fine-tuned BERT$_{LARGE}$ (Devlin et al., 2018). Results are reported in F1 scores.

We follow most of the baseline performances reported in (Lample et al., 2016; Liu et al., 2019b; Qin et al., 2019; Liu et al., 2019a) and rerun the open source toolkit NCRFpp[3], LM-LSTM-CRF[4], and GCDT[5] on slot filling tasks[6].

Implementation Details We use the same configuration setting for all datasets. The hidden dimensions are set as 500. We apply dropout to hidden states with a rate of 0.3. L2 regularization is set as 1×10^{-6} to avoid overfit. Following (Liu et al., 2018, 2019a,b), we adopt the cased, 300d Glove (Pennington et al., 2014) to initialize word embeddings. We utilize Adam algorithm (Kingma and Ba, 2015) to optimize the models and adopt the suggested hyper-parameters.

5.2 Main Results

The main results of the proposed model on ATIS and CoNLL-03 are illustrated in Table 2. The proposed model outperforms all other models on all tasks by a substantial margin. On slot filling tasks, the model obtains averaged improvements of 0.15 points on ATIS and 1.53 points on SNIPS over CM-Net and Stack-propagation, without using extra information from jointly modeling of slots and intents in these models. In comparison to the prior state-of-the-art models of GCDT, the improvements are 0.03 points on ATIS, 2.17 points on SNIPS and 0.71 points on CoNLL-03.

Compared with strong baseline (Lample et al., 2016) that utilizes char embedding to improve Bi-LSTM + CRF, the gains are even larger. The model obtains improvements of 0.84 points on ATIS, 3.49 points on SNIPS and 1.73 points on CoNLL-03, over Bi-LSTM + CRF and LM-LSTM-CRF.

Finally, we have tried improving the baseline Bi-LSTM+CRF in our model with external resources of lexical information, including part-of-speech tags, chunk tags and character embeddings. However, their F1 scores are consistently below the proposed model by an average of 1.47 points. We also replace word embeddings in Bi-LSTM+CRF with those from fine-tuned BERT$_{LARGE}$ but its results are worse than the proposed model, by 0.07

[2]https://nlp.stanford.edu/software/lex-parser.shtml.
[3]https://github.com/jiesutd/NCRFpp.
[4]https://github.com/LiyuanLucasLiu/LM-LSTM-CRF.
[5]https://github.com/Adaxry/GCDT.
[6]Few results are not available for comparison as Qin et al. (2019); Liu et al. (2019b) are for mult-task learning of intent detection and slot filling.

Method	SNIPS
Bi-LSTM + CRF + LCR + DEI	**97.20**
w/o LCR	94.37
w/o VAE, w/ LSTM-LM	96.02
w/o OOV masking	95.63
w/o DEI	95.82
w/o LCR, DEI (Bi-LSTM + CRF)	93.37

Table 3: Ablation experiments for local context reconstruction (LCR) and delexicalized entity identification (DEI). LCR includes VAE and OOV masking.

points, 1.05 points and 0.14 points, respectively, on ATIS, SNIPS, and CoNLL-03.

6 Analysis

It is noteworthy that the substantial improvements by the model are obtained without using external resources nor large pre-trained models. Keys to its success are local context reconstruction and delexicalized entity identification. This section reports our analysis of these modules.

6.1 Ablation Study

Local Context Reconstruction (LCR) We first examine the impact bought by the LCR process. In Table 3, we show that removing LCR (w/o LCR) hurts performance significantly on SNIPS. We then study if constructing local context in LCR using a traditional deterministic encoder-decoder can be equally effectively as using VAE. We make a good faith attempt of using LSTM-based language model (Sundermeyer et al., 2012) to generate local context directly from local context representation (w/o VAE, w/ LSTM-LM). This does improve results over that without LCR at all, indicating the information from reconstructing local context is indeed useful. However, its F1 score is still far worse than that of using VAE. This confirms that VAE is superior to deterministic model in dealing with the inherently one-to-many generation of local context from entities. Lastly, we examine the impact of OOV masking and observe that F1 score without it (w/o OOV masking) drops about 1.6 point below the model. We attribute this improvement from OOV masking to mitigating the entity distribution discrepancy between training and test.

Delexicalized Entity Identification (DEI) Removing delexicalized entity identification (w/o DEI) performs worse than the model, with large drop of 1.38 point on SNIPS.

Method	CoNLL-03	
	OOV	LF
LM-LSTM-CRF	2049	1136
GCDT	2073	1149
Bi-LSTM + CRF	2041	1135
w/ external resource‡	2052	1143
w/ BERT fine-tuned embedding*	2084	1153
Bi-LSTM + CRF + LCR	**2112**	1139
Bi-LSTM + CRF + DEI	2043	**1169**
Bi-LSTM + CRF + LCR + DEI	**2124**	**1181**
Total	2509	1363

Table 4: Numbers of OOV and LF entities that are correctly labeled. * refers to fine tuning on pretrained large models. ‡ refers to adopting external resources.

These results show that both local context reconstruction and delexicalized entity identification contribute greatly to the improved performance by the proposed model. Because both LCR and DEI share the same RNN-encoder as the baseline Bi-LSTM, the information from reconstructing local context and fooling the discriminator of delexicalization is useful for the Bi-LSTM to better predict sequence labels.

6.2 Rare Entity Handling

In this section, we compare models specifically by the numbers of OOV and LF entities they can recall correctly. Such comparison reveals the capability of each model in handling rare entities.

Results are presented in Table 4. In the case of without using any external resource and pre-trained models, the proposed model recalls 3.66% more OOV entities and 3.96% more LF entities than LM-LSTM-CRF. This gain is similar when comparing against Bi-LSTM+CRF. Furthermore, the proposed model also recalls more rare entities than GCDT, a recent state-of-the-art model in NER. Separately using LCR or DEI improves performance over baseline Bi-LSTM+CRF. Their gains are complementary as results show that jointly applying LCR and DEI obtains the best performance. These results demonstrate convincingly the capability of local context reconstruction and delexicalized entity identification in rare entities.

Importantly, results in the last two rows reveal that potentially large improvements can be potentially achieved since there are still 15.34% of OOV entities and 13.35% of LF entities not recalled.

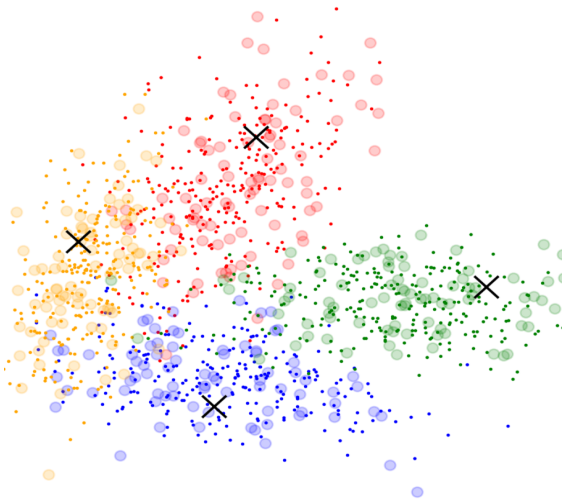


Figure 3: Visualization of learned representations on CoNLL-03 test dataset. Entity types are represented in different shapes with red for PER, blue for ORG, green for LOC and orange for MISC. Rare entities are represented using bigger points. The points with "X" are for the delexicalized entities.

6.3 Representation for Delexicalized Entity Identification

We visualize the learned representation at Eq. (15) using t-SNE (Maaten and Hinton, 2008) in Figure 3. It shows 2-dimensional projections of randomly sampled 800 entities on CoNLL-03 dataset.

Figure 3 clearly shows separability of entities by their entity types but no separations among low-frequency and frequent entities. This observation is consistent to the mini-max objective in Eq. (17) to learn entity-type-specific and frequency-agnostic representations.

6.4 Handling Data Scarcity

This section investigates the proposed model on data scarcity. On ATIS, the percentage of training samples are reduced down to 20% of the original size, with a reduction size of 20%. This setting is challenging and few previous works have experimented. Results in Figure 3 show that the proposed model consistently outperforms other models, especially in low-resource conditions. Furthermore, reductions of performance from the proposed model are much smaller, in comparison to other models. For instance, at percentage 40%, the proposed model only lose 1.17% of its best F1 score whereas GCDT loses 3.62% of its F1 score. This suggests that the proposed model is more robust to low resource than other models.

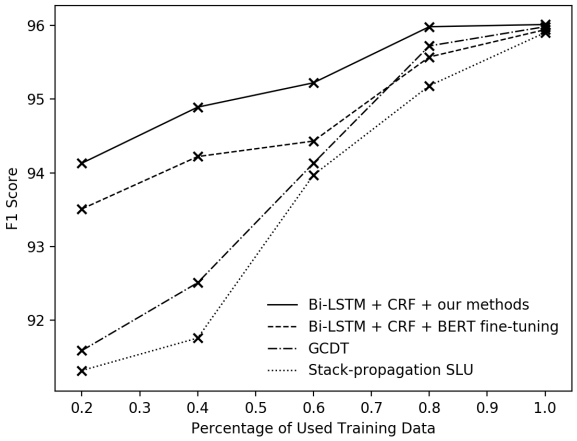


Figure 4: Comparisons with respect to different percentage of training data on ATIS.

7 Related Work

Neural sequence labeling has been an active field in NLP, and we briefly review recently proposed approaches related to our work.

Slot Filling and NER Neural sequence labeling has been applied to slot filling (Mesnil et al., 2014; Zhang and Wang, 2016; Liu and Lane, 2016; Qin et al., 2019) and NER (Huang et al., 2015; Strubell et al., 2017; Liu et al., 2018; Devlin et al., 2018; Liu et al., 2019a). For slot filling, multi-task learning for joint slot filling and intent detection has been dominating in the recent literature, for example (Liu and Lane, 2016). The recent work in (Liu et al., 2019b) employs a collaborative memory network to further model the semantic correlations among words, slots and intents jointly. For NER, recent works use explicit architecture to incorporate information such as global context (Liu et al., 2019a) or conduct optimal architecture searches (Jiang et al., 2019). The best performing models have been using pre-training models on large corpus (Baevski et al., 2019) or incorporating fine-tuning on existing pre-trained models (Liu et al., 2019a) such as BERT (Devlin et al., 2018).

External Resource This approach to handle rare entities includes feature engineering methods such as incorporating extra knowledge from part-of-speech tags (Huang et al., 2015) or character embeddings (Li et al., 2018). Extra knowledge also includes tags from public tagger (Manning et al., 2014). Multi-task learning has been effective in incorporating additional label information through multiple objectives. Joint slot filling and intent detection have been used in (Zhang and Wang, 2016;

Qin et al., 2019; Zhang et al., 2019). Joint part-of-speech tagging and NER have been used in (Lin et al., 2018).

Transfer Learning This approach refers to methods that transfer knowledge from high-resources to low-resources (Zhou et al., 2019) or use models pretrained on large corpus to benefit downstream tasks (Devlin et al., 2018; Liu et al., 2019a). The most recent work in (Zhou et al., 2019) applies adversarial training that uses a resource-adversarial discriminator to improve performances on low-resource data.

8 Conclusion

We have presented local context reconstruction for OOV entities and delexicalized entity identification for low-frequency entities to address the *rare entity problem*. We adopt variational autoencoder to learn a stochastic reconstructor for the reconstruction and adversarial training to extract frequency-agnostic and entity-type-specific features. Extensive experiments have been conducted on both slot filling and NER tasks on three benchmark datasets, showing that sequence labeling using the proposed methods achieve new state-of-the-art performances. Importantly, without using external knowledge nor fine tuning of large pretrained models, our methods enable a sequence labeling model to outperform models fine-tuned on BERT. Our analysis also indicates large potential of further performance improvements by exploiting OOV and LF entities.

Acknowledgments

This work was done while the first author did internship at Ant Financial. We thank anonymous reviewers for valuable suggestions.

References

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. 2019. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Issam Bazzi. 2002. *Modelling out-of-vocabulary words for robust speech recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. pages 2493–2537.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, pages 4171–4186.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In *Advances in neural information processing systems*, pages 1334–1345.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

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

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

Yufan Jiang, Chi Hu, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2019. Improved differentiable architecture search for language modeling and named entity recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3576–3581.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Diederik P Kingma and Max Welling. 2015. Auto-encoding variational bayes. *ICLR*.

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

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. *ACL*, pages 799–809.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019a. GCDT: A global context enhanced deep transition architecture for sequence labeling. *ACL*, pages 2431–2441.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019b. CM-Net: A novel collaborative memory network for spoken language understanding. *arXiv preprint arXiv:1909.06937*.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2015. Distributional smoothing with virtual adversarial training.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv preprint arXiv:1909.02188*.

Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative mod- els. *ICML*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language- independent named entity recognition. *arxiv preprint cs/0306050*.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673–2681.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate sequence labeling with iterated dilated convolutions. *EMNLP*, 138:2670–2680.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in ATIS? *IEEE Spoken Language Technology Workshop (SLT)*, pages 19–24.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2306–2315.

Tsung-Hsien Wen, Milica Gasic, Dongho Kim, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*.

Yingwei Xin, Ethan Hart, Vibhuti Mahajan, and Jean-David Ruvini. 2018. Learning better internal structure of words for sequence labeling.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu. 2019. Joint slot filling and intent detection via capsule neural networks. *ACL*, pages 5259–5267.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. *ACL*, pages 3461–3471.