Graph LSTM with Context-Gated Mechanism for Spoken Language Understanding

Linhao Zhang,[1] Dehong Ma,[1] Xiaodong Zhang,[1] Xiaohui Yan,[2] Houfeng Wang[1]
[1]MOE Key Lab of Computational Linguistics, Peking University, Beijing, 100871, China
[2]CBG Intelligence Engineering Dept, Huawei Technologies, China
{zhanglinhao, madehong, zxdcs, wanghf}@pku.edu.cn
yanxiaohui2@huawei.com

Abstract

Much research in recent years has focused on spoken language understanding (SLU), which usually involves two tasks: intent detection and slot filling. Since Yao et al.(2013), almost all SLU systems are RNN-based, which have been shown to suffer various limitations due to their sequential nature. In this paper, we propose to tackle this task with Graph LSTM, which first converts text into a graph and then utilizes the message passing mechanism to learn the node representation. Not only the Graph LSTM addresses the limitations of sequential models, but it can also help to utilize the semantic correlation between slot and intent. We further propose a context-gated mechanism to make better use of context information for slot filling. Our extensive evaluation shows that the proposed model outperforms the state-of-the-art results by a large margin.

Introduction

Spoken language understanding (SLU) is an essential part of dialog system. It usually involves two tasks: intent detection (ID) and slot filling (SF). Typically, ID is regarded as a semantic utterance classification problem, and different classification methods can be applied (Haffner, Tur, and Wright 2003; Tür et al. 2011; Deng et al. 2012). Meanwhile, SF is usually treated as a sequence labeling problem. Popular approaches to perform SF include support vector machines (SVMs) and conditional random fields (CRFs) (Lafferty, McCallum, and Pereira 2001).

Yao et al.(2013) adapted RNN language models to perform SLU, outperforming previous CRF-based models by a large margin. RNN-based methods (including LSTM and GRU) have since defined the state-of-the-art in SLU research (Mesnil et al. 2015; Liu and Lane 2016; Zhang and Wang 2016; Goo et al. 2018; Niu et al. 2019).

Despite their success, these RNN-based models have been shown to suffer various limitations. Firstly, their inherently sequential nature precludes parallelization within training examples (Vaswani et al. 2017). Secondly, local n-grams are not fully exploited in their models. In SLU, slots are not only determined by the associated items, but also local context.

Utterance	show	flights	from	Seattle	to	San	Diego
Slots	O	O	O	B-fromloc	O	B-toloc	I-toloc
Intent	Flight						

Figure 1: An example of SLU utterance with intent and annotated slots using the IOB scheme. The B- prefix before a tag indicates that the tag is the beginning of a slot, and an I- prefix before a tag indicates that the tag is inside a slot. An O tag indicates that a token belongs to no slot.

As shown in Figure 1, the corresponding slot label for *Seattle* is *B-fromloc*, but it could also be *B-toloc*, if the utterance is transformed into *show flights from San Diego to Seattle*. Thirdly, the sequential nature of RNN-based methods leads to weaker power in capturing long-range dependencies, which accounts for a large portion of SF errors (Tür, Hakkani-Tür, and Heck 2010).

In this paper, we propose to use Graph LSTM to tackle these problems. There are many variants of Graph LSTM (Liang et al. 2016; Peng et al. 2017; Zayats and Ostendorf 2018; Song et al. 2018; Zhang, Liu, and Song 2018). In this paper, we choose the S-LSTM (Zhang, Liu, and Song 2018) because it is ideally suited for this task.

The main idea of S-LSTM is to model the hidden states of all words simultaneously rather than sequentially, hence can solve the non-parallelization problem. Specifically, the S-LSTM views the whole sentence as a single graph, which consists of word-level nodes and a sentence-level node. These nodes are updated simultaneously through message passing mechanism. Since message passing is conducted between consecutive word-level nodes, and between sentence-level node and each word-level node, both local n-grams and long-range dependencies are better captured.

Compared to other variants of Graph LSTM, the S-LSTM has a special sentence-level node, making it ideally suited to utilize the semantic correlation between slot and intent. We note that intent and slot are not independent but intrinsically correlated. As the example shown in Figure 1, an utterance is more likely to contain departure and arrival cities if its intent is to find a flight, and vice versa. For joint ID and SF, we use the final word-level nodes of S-LSTM for slots

prediction and the final sentence-level node to predict intent (See Figure 2). The two kinds of nodes are updated based on the value of each other at each time step, and changes in the word-level nodes (slot nodes) would lead to a corresponding change in the sentence-level node (intent node), and vice versa. In this way, the semantic correlation between intent and slot is explicitly modeled by the S-LSTM through its message passing mechanism.

To further improve SF performance, we employ a context-gated mechanism on top of the Graph LSTM. Specifically, we first employ a convolutional unit for modeling *local context*. Inspired by the Inception (Szegedy et al. 2016; Lin et al. 2018) architecture widely used in Computer Vision (CV), our convolutional unit is capable of extract n-grams of different sizes. Besides, a multi-head self-attention unit is used to model the *global context*. The self-attention mechanism has become an integral part of compelling sequence modeling in various tasks, allowing modeling of dependencies without regard to their distance. The two units are combined to compute a context gate, which is used to modify the slot vectors for the final prediction.

To sum up, our contributions are threefold:

1) To the best of our knowledge, we are the first to introduce the Graph LSTM structure into the SLU area. The proposed model outperforms state-of-the-art results by a large margin.

2) Our model offers a new approach to model the semantic correlation between intent and slot.

3) We propose a context-gated mechanism that considers both local and global context, which substantially contributes to SF performance.

Model

Figure 2 gives an overview of our proposed model. Given a sequence of words $(w_1, ..., w_n)$ in an utterance, we first transform them into word embeddings, $(e_1, ..., e_n)$, and then input these embeddings into the Graph LSTM at each time step. After T steps, we use the word-level nodes to predict slot labels and the sentence-level node for intent prediction. Throughout the paper, we will use the two pairs of items, *word-level nodes* and *slot nodes*, and *sentence-level node* and *intent node*, interchangeably. For SF, we further add a context gate to improve our model's ability to utilize local and global context information. The output of our model is a sequence of slots labels $(y_1^s, ..., y_n^s)$, plus the intent y^u of the whole utterance. A detailed description is given below.

Input Encoding

To obtain robust word representations, we map each word to a vector using the pretrained ELMo embeddings (Peters et al. 2018), which have shown superior performance in a wide range of NLP tasks. The ELMo embeddings are totally character-based, allowing the network to utilize morphological clues to form robust representations for out-of-vocabulary (OOV) words, hence are especially suited for this low data regime.

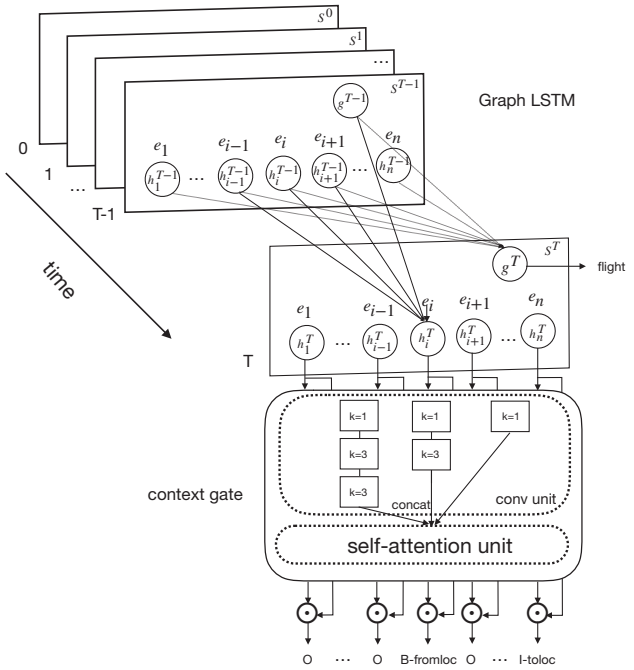

Figure 2: An overview of our proposed model. The Graph-LSTM views the whose sentence as a single graph S and uses message passing mechanism to learn the node representation. After T time steps, we use the final word-level node h_i^T to predict the i_{th} slot label and the sentence-level node g^T to predict intent (where T is hyperparameter). For SF, a context gate, which consists of a convolution unit and a self-attention unit, is further added on top of the Graph LSTM to capture local and global context information.

Graph LSTM

There are many variants of Graph LSTM. In this paper, we choose the S-LSTM (Zhang, Liu, and Song 2018) as the backbone of our model. We first describe the basic structure of S-LSTM and then analyze the strengths of using S-LSTM for this task.

S-LSTM views the whole sentence as a single graph S, which consists of word-level nodes h_i and a sentence-level node g. Formally, the graph state at time step t is represented as:

$$S^t = \langle h_1^t, h_2^t, \ldots, h_n^t, g^t \rangle \quad (1)$$

The S-LSTM updates its node representation using message passing mechanism. As show in Figure 2, the graph state transition from S^{t-1} to S^t consists of node state transitions from h_i^{t-1} to h_i^t and from g^{t-1} to g^t. At each time step, the value of h_i^t is updated according to $e_i, h_{i-1}^{t-1}, h_i^{t-1}$ h_{i+1}^{t-1} and g^{t-1}, together with their corresponding cell values. Formally:

$$
\begin{aligned}
\xi_i^t &= [h_{i-1}^{t-1}, h_i^{t-1}, h_{i+1}^{t-1}] \\
\hat{i}_i^t &= \sigma(W_i \xi_i^t + U_i e_i + V_i g^{t-1} + b_i) \\
\hat{l}_i^t &= \sigma(W_l \xi_i^t + U_l e_i + V_l g^{t-1} + b_l) \\
\hat{r}_i^t &= \sigma(W_r \xi_i^t + U_r e_i + V_r g^{t-1} + b_r) \\
\hat{f}_i^t &= \sigma(W_f \xi_i^t + U_f e_i + V_f g^{t-1} + b_f) \\
\hat{s}_i^t &= \sigma(W_s \xi_i^t + U_s e_i + V_s g^{t-1} + b_s) \\
o_i^t &= \sigma(W_o \xi_i^t + U_o e_i + V_o g^{t-1} + b_o) \\
i_i^t, l_i^t, r_i^t, f_i^t, s_i^t &= \mathrm{softmax}(\hat{i}_i^t, \hat{l}_i^t, \hat{r}_i^t, \hat{f}_i^t, \hat{s}_i^t) \\
c_i^t &= l_i^t \odot c_{i-1}^{t-1} + f_i^t \odot c_i^{t-1} + r_i^t \odot c_{i+1}^{t-1} \\
&\quad + s_i^t \odot c_g^{t-1} + i_i^t \odot u_i^t \\
h_i^t &= o_t^i \odot \tanh(c_i^t)
\end{aligned} \tag{2}
$$

where W_x, U_x, V_x and b_x are model parameters ($x \in \{i, o, l, r, f, s, u\}$).

Note that there are more control gates in S-LSTM than in LSTM. These gates control information flow from ξ_i^t and e_i to c_i^t . Specifically, i_i^t controls information from the input e_i; l_i^t and r_i^t control information from the left context cell c_{i-1}^{t-1} and the right context c_{i+1}^{t-1} respectively. f_i^t, as in LSTM, is the forget gate that control information flow from c_i^{t-1}. s_i^t is the sentence gate that controls information from sentence cell value c_g^{t-1}. The values of these control gates are normalized by a softmax function and then used as weights to obtain the new cell value c_i^t and finally the word-level node h_i^t (Zhang, Liu, and Song 2018).

The sentence-level node g^t is updated based on all these word-level nodes:

$$
\begin{aligned}
\overline{h} &= \mathrm{avg}(h_1^{t-1}, h_1^{t-1}, \ldots, h_n^{t-1}) \\
\hat{f}_g^t &= \sigma(W_g g^{t-1} + U_g \overline{h} + b_g) \\
\hat{f}_i^t &= \sigma(W_f g^{t-1} + U_f h_i^{t-1} + b_f) \\
o^t &= \sigma(W_o g^{t-1} + U_o \overline{h} + b_o) \\
f_1^t, \ldots, f_n^t, f_g^t &= \mathrm{softmax}(\hat{f}_1^t, \ldots, \hat{f}_n^t, \hat{f}_g^t) \\
c_g^t &= f_g^t \odot c_g^{t-1} + \sum_i f_i^t \odot c_i^{t-1} \\
g^t &= o^t \odot \tanh(c_g^t)
\end{aligned} \tag{3}
$$

where W_x, U_x and b_x are model parameters ($x \in \{g, f, o\}$). Note that for conciseness, we here abuse the notation by utilizing the same notations as Equation 2, though the sentence-level node uses a different set of parameters from the word-level nodes.

Since the message passing between word-level nodes is only between neighbouring ones, we can regard this setting as adopting a 1-word window. A wider window size χ may help to capture larger n-grams as well as expedite information flow. To achieve this, we may modify Equation 2, integrating additional context words to ξ_i^t, with extended gates and cells. For example, with a window size $\chi = 2$, $\xi_i^t = [h_{i-2}^{t-1}, h_{i-1}^{t-1}, h_i^{t-1}, h_{i+1}^{t-1}, h_{i+2}^{t-1}]$.

After T time step, we use the word-level nodes h^T and the sentence-level node g^T to predict slot and intent respectively. At each time step, the slot nodes (h^T) and intent node (g^T) are updated based on the value of each other. Changes in slot nodes would lead to a corresponding change in the intent node, and vice versa. In this way, the semantic correlation between slot and intent is explicitly modeled.

By employing S-LSTM for SLU, we also improve the model's ability to extract n-grams. From Equation 2, we can see that the S-LSTM allows bi-directional information flow at each word simultaneously. At each time step, each slot vector captures an increasing larger n-gram context. Besides, since the word-level nodes also get information from the sentence-level node, which stores global information, the model's ability to capture long-range dependencies gets improved.

Context-Gated Mechanism

As we have mentioned, slots are determined not only by the associated items, but also by context. Although the Graph LSTM already has certain ability to capture this information, we find that adding a context gate to explicitly utilize it leads to better performance. The context gate is composed of a convolution unit and self-attention unit. These two units help us to utilize local and global context, respectively.

Convolution Unit - Similar to images, language also contains local correlation, such as the internal correlation of phrase structure. To capture this information, we employ 1-d convolution unit on top of the S-LSTM. Convolutions create representations for local n-grams, which is equal in size to the filters. Formally, a convolution block is computed as:

$$
conv_i = \mathrm{ReLU}\left(W_c \left[h_{i-k/2}^T, \ldots, h_{i+k/2}^T\right] + b_c\right) \tag{4}
$$

where W_c and b_c are parameters; k is the filter size.

However, choosing the right filter size for the convolution operation is hard. Inspired by the Inception architecture (Szegedy et al. 2016; Lin et al. 2018), we have filters of multiple sizes operate on the same level to capture different local n-grams, yielding three different representations of the previous layer, noted as $conv^1$, $conv^2$ and $conv^3$ respectively.

These vectors are concatenated and then linearly transformed to the same dimension as the previous layer. Formally,

$$
z_i = [conv_i^1; conv_i^2; conv_i^3] W_z \tag{5}
$$

where W_z is model parameter.

Self-Attention Unit - Sometimes the disambiguating words are out of the current n-gram context. For example, in the following utterance (Tür, Hakkani-Tür, and Heck 2010):

Find flights to New York arriving in no later than next Saturday

A 6-gram context is required to determine that *Saturday* is an *arrival date*. Many of the erroneous cases of SF are caused by this kind of long-range dependencies, as pointed out by (Tür, Hakkani-Tür, and Heck 2010).

To tackle this challenge, we further implement a self-attention (Vaswani et al. 2017) unit on top of the convolution unit. An attention function maps a query and a set of

key-value pairs to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Formally,

$$Attn(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (6)$$

where Q, K and V are packed queries, keys and values, and d_k is the dimensions of keys.

For self-attention, all of the keys, values and queries come from the same place, in our cases, the S-LSTM's final word-level nodes h^T. In this way, dependencies in the utterance can be modeled without regard to their distance.

Vaswani et al.(2017) further proposed a multi-head self-attention mechanism, which can learn diverse representations of the input. Following their approach, we first project the d_{model}-dimensional queries, keys and values h times with different linear projections to d_k, d_k and d_v dimensions respectively. We then perform the attention function on each of these projected vectors, resulting in d_v-dimensional output values, which are concatenated and once again projected, yielding the final values. Formally,

$$Multi(Q, K, V) = [head_1, \ldots, head_{\text{h}}]W^O \quad (7)$$

$$head_i = Attn(QW_i^Q, KW_i^K, VW_i^V) \quad (8)$$

where the projections are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}, W_i^K \in R^{d_{model} \times d_k}, W_i^V \in R^{d_{model} \times d_v}$ and $W^O \in R^{hd_v \times d_{model}}$.

Finally, the self-attention unit takes as input the output of the convolution unit and results in the final context gate value $\tilde{z}$, which is used to modify h_i^T for the final prediction of slot labels:

$$\tilde{z} = Multi(z, z, z) \quad (9)$$

$$\tilde{h_i}^T = h_i^T \odot \sigma(\tilde{z_i}) \quad (10)$$

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

Task Learning

The softmax function is applied to $\tilde{h^T}$ and g^T with linear transformations to give the probability distribution y_i^s over the i-th slot labels and the distribution y^u over the intent labels. Formally,

$$\begin{aligned} y_i^s &= \text{softmax}(W_m \tilde{h}_i^T + b_m) \\ y^u &= \text{softmax}(W_n g^T + b_n) \end{aligned} \quad (11)$$

where W_m, W_n, b_m, b_n are model parameters.

The next step is to define the loss function for our networks. We use U to denote the utterance, l^s and l^u to denote the ground truth label of slot and intent. The cross-entropy loss for ID and SF is computed as:

$$\mathcal{L}^u = -l^u \log y^u \quad (12)$$

$$\mathcal{L}^s = -\sum_{i=1}^{n} l_i^s \log y_i^s \quad (13)$$

	Snips	ATIS
Vocabulary size	11241	722
Training set	13084	4478
Validation set	700	500
Test set	700	893
# Slot	72	120
# Intent	7	21

Table 1: Statistics of Snips and ATIS datasets

where n is the length of each utterance, which can vary among the training samples.

The training target of the network is to minimize a united loss function:

$$\mathcal{L} = \Sigma_{(l^s, l^u, U) \in \mathcal{D}}(\alpha \mathcal{L}^s + \mathcal{L}^u) \quad (14)$$

where D denotes the whole dataset; α is a weight factor to adjust the importance of the two tasks.

Experiments

Datasets

To fully evaluate the proposed model, we conducted experiments on the Snips and ATIS datasets. The statistics of these two datasets are shown in Table 1.

Snips - The Snips dataset was created by *snips.ai* (Coucke et al. 2018). It is in the domain of personal assistant commands. Compared to the ATIS corpus, it is more complicated in terms of vocabulary size and the diversity of intent and slots. There are 13,084 utterances in the training set and 700 utterances in the test set, with a development set of 700 utterances. There are 72 slot labels and 7 intent types. The diversity of intent and slots is an important feature of Snips dataset. Intents in ATIS are all about fight information, while Snips contains intents like *RateBook* and *GetWeather* that come from totally different topics. Goo et al.(2018) first exploited this dataset and reported results for models that had achieved state-of-the-art performance on ATIS, along with their own results.

ATIS - The Airline Travel Information Systems (ATIS) (Hemphill, Godfrey, and Doddington 1990) dataset has long been used as benchmark in SLU. There are some variants of the ATIS dataset. In this work, we use the same one as used in Goo et al.; Niu et al.(2018; 2019). There are 4,478 utterances in the training set, 500 in the valid set and 893 in the test set, with a total of 120 distinct slot labels and 21 different intent types.

Evaluation Metrics

In this paper, we adopt three mainstream evaluation metrics:

SF - We evaluate the system's performance on SF using the F1 score, which is defined as the harmonic average of the precision and recall.

ID - The metric for ID is classification accuracy. Some utterances in the ATIS dataset have more than one intent labels. Some researchers (Liu and Lane 2016; Li, Li, and Qi 2018) count an utterance as a correct classification if any ground truth label is predicted. Others (Goo et al. 2018;

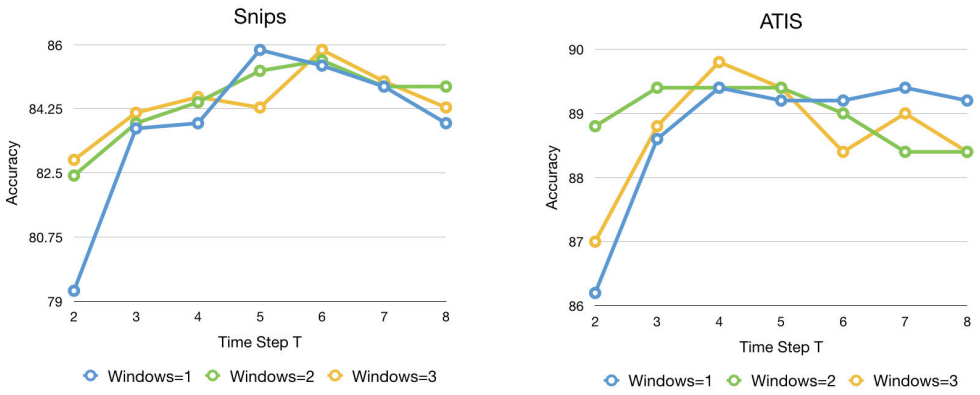

Figure 3: Sentence-level accuracies with various window sizes and time steps on the development sets.

Niu et al. 2019) require that all of these intent labels have to be correctly predicted if an utterance is to be counted as a correct classification. Since we use the same datasets as Goo et al.; Niu et al.(2018; 2019), we adopt the latter setting for a fair comparison.

Sent. - Following Goo et al.; Niu et al.(2018; 2019), we also report the sentence-level accuracy, which considers both SF and ID performance. A sentence is counted as correct if all its slot labels and intent are correctly predicted.

Implementation Details

The model was implemented in PyTorch and trained on a single NVIDIA GeForce GTX 1080 GPU,

The dimensions of S-LSTM hidden state is set to 150. The dimensions of ELMo embeddings are 1024. We do not update ELMo representations during training to reduce training time. The batch size is set to 32. Dropout (Hinton et al. 2012) layers are applied on both input and output vectors during training for regularization, with a dropout rate of 0.5.

We use Adam (Kingma and Ba 2015) for the training process to minimize the cross-entropy loss, with learning rate = 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$.

Following previous work (Zhang and Wang 2016; Li, Li, and Qi 2018), we convert each number in the utterance to the string *DIGIT* for ATIS. For Snips dataset, we set words that appear less than two times to *UNK*.

We assign equal attention to ID and SF; that is, we set α in Equation 14 to be 1.

Systems for Comparison

We compared our model against the following baselines:

Joint Seq - (Hakkani-Tür et al. 2016) proposed a approach to jointly models SF and ID in a single bi-directional RNN with LSTM cells.

Atten. Based - (Liu and Lane 2016) proposed attention-based bidirectional RNN for for joint SF and ID. They explored different strategies in incorporating this alignment information to the encoder-decoder framework.

Slot-gated - (Goo et al. 2018) proposed a slot-gated mechanism to learn slot-intent relations. They applied the intent information for SF but the slot information was not used for ID.

SF-ID Net. - (Niu et al. 2019) proposed an SF-ID network to establish direct connections for ID and SF to help them promote each other, with a new iteration mechanism inside the SF-ID network. They also used a CRF layer to further improve performance.

The results of baselines are taken from Goo et al.(2018) and Niu et al.(2019) because we use the same datasets and evaluation metrics.

Results

Development Experiments

We first use the development sets to find the optimal window size χ and time step T of the S-LSTM. We test the sentence-level accuracy, which considers both SF and ID performance. The results are shown in Figure 3[1].

We can see that when the number of time steps increases from 2 to 8, the sentence-level accuracies generally increase, before reaching a maximum value. This shows the effectiveness of the message passing mechanism in S-LSTM state transition. To achieve the best performance, we set χ = 3 and T = 6 for the Snips dataset. For the ATIS, we set χ = 3 and T = 4.

Overall Performance

The overall performance on the Snips and ATIS datasets are demonstrated in Table 2.

On the Snips dataset, our model achieves the top performance by a significant margin. Specifically, for SF, our model achieves an F1 score of 95.30%, a significant improvement over previous state-of-the-art (3.07% absolute). For ID, we get an absolute 1% improvement over the best-reported score. For the sentence-level accuracy, our improvement is more striking, with over 9% absolute improvement. This is because the sentence-level accuracy benefits more from the modeling of the bi-directional correlation between slots and intent.

[1]Results on the development sets were obtained without ELMo embeddings

Model	Snips			ATIS		
	SF	ID	Sent.	SF	ID	Sent.
Joint Seq (Hakkani-Tür et al. 2016)	87.30	96.90	73.20	94.30	92.60	80.70
Atten.-Based (Liu and Lane 2016)	87.80	96.70	74.10	94.20	91.10	78.90
Slot-Gated (Goo et al. 2018)	89.27	96.86	76.43	95.42	95.41	83.73
SF-ID, SF first (Niu et al. 2019)	91.43	97.43	80.57	95.75	**97.76**	86.79
SF-ID, ID first (Niu et al. 2019)	92.23	97.29	80.43	95.80	97.09	86.90
Our Model	**95.30**	**98.29**	**89.71**	**95.91**	97.20	**87.57**
Our Model - ELMo	93.81	97.71	85.57	95.84	96.41	86.23
Our Model - Context	94.56	97.29	88.00	95.72	97.09	86.11

Table 2: Results on the Snips and ATIS datasets (%).

Although the ATIS dataset has been well studied for decades and the scores of the state-of-the-art methods are very high, we still achieve noticeable improvement. We achieve new start-of-the-art performance with an F1 score of 95.91% and sentence-level accuracy of 87.57%, with ID result slightly lower than the SF-First model of Niu et al.(2019). This is understandable, because their SF-First model actually gives high priority to ID, at the cost of SF performance.

As such, empirical results demonstrate the effectiveness of our proposed model on both datasets.

We also notice that the performance improvement is more significant on the Snips dataset than on the ATIS dataset. We suggest that this is because the Snips dataset is more diverse, and thus the intent can provide more useful information for slots prediction, and vice versa. As a result, our joint model can benefit more from the modeling of the correlation between slot and intent on this dataset.

Ablation and Analysis

This subsection aims to demonstrate the relative effectiveness of different components of our proposed model. The results are also shown in Table 2.

ELMo - We first replaced the character-based ELMo embeddings with non-pretrained word embeddings. On the Snips dataset, we observed a 1.49% drop for SF and 0.58% drop for ID. On the ATIS dataset, the performance also deteriorates.

The results support our claim that pretrained, character-based embeddings are beneficial to SLU systems. Both the Snips and ATIS datasets are small, and if we train word embeddings from scratch the embeddings could be poorly trained, especially for rare words. By depending solely on word embeddings, it is also hard to exploit character-level features like prefix and suffix. The character-based ELMo representations, on the other hand, can make full use of the morphological clues to form robust representations for OOV tokens, hence are especially beneficial for this low data regime.

We are more interested in the comparison between our ELMo-free model and baselines. As can be seen, our model is highly competitive even without the ELMo representations. On the Snips dataset, our ELMo-free model still achieves state-of-the-art performance by a large margin, with over 5% absolute improvement on the sentence-level accuracy. On the ATIS dataset, we also achieve the best F1 score, with intent accuracy lower than Niu et al.(2019). These results suggest the superior performance of our model lies more in the model design than in the pretrained ELMo embeddings.

Model	Snips		ATIS	
	SF	ID	SF	ID
SF-model	94.28	-	95.79	-
ID-model	-	97.71	-	96.64
Joint-model	**95.30**	**98.29**	**95.91**	**97.20**

Table 3: Comparison between joint model and separate models (%).

Context-Gated Mechanism - We then removed the context gate from our model. As can be seen, the F1 scores drop 0.74% and 0.19% for Snips and ATIS datasets, respectively. Interestingly, we found that the intent accuracies also deteriorated on both datasets. We suggest this might be due to the semantic correlation between slot and intent, and the deterioration of SF performance leads to a corresponding drop of ID accuracy. We can also observe that the sentence-level accuracy drops significantly, with 1.71% and 1.46% absolute drop on the Snips and ATIS datasets, respectively.

The results buttress our claim that the context gate mechanism is beneficial to SLU performance. We further note that both the convolution unit and self-attention unit are non-sequential, which match well with the Graph LSTM.

Joint Model vs Separate Model

One of the main claims of this paper is that the correlation of slot and intent is explicitly modeled by our model and hence contributes to both tasks. To further examine this claim, experiments were designed to compare the joint model with the separate models. The joint model is our proposed model in Figure 2, while the separate models follow the basic structure, but focus on only one task. For example, the ID-model learns only ID and does not predict slot labels. In this way, the separate models cannot benefit from the modeling of the semantic correlation between intent and slot. The results are demonstrated in Table 3

We can see that the joint model generally performs better than the two separate models. Specifically, on Snips dataset, the joint model beats the SF-model by 1.02% absolute F1

Model	F1
DBN (Deoras and Sarikaya 2013)	95.30
CRF (Mesnil et al. 2015)	95.16
RNN (Mesnil et al. 2015)	96.29
GRU-CRF (Zhang and Wang 2016)	96.89
Our model	**97.28**

Table 4: F1 scores on the ATIS dataset with named entity (NE) feature.

score, and ID-model by 0.58%. On the ATIS dataset, our joint model outperforms the SF-model by 0.12% F1 score and the ID-model by 0.56%.

Not only the results back our claim, they also suggest that the more diverse dataset can benefit more from the modeling of the semantic correlation between slot and intent, because ID and SF can provide more useful information for each other. This shows the potential of our model to be applied in open-domain area.

Named Entity Feature

The ATIS dataset also has extra named entity (NE) feature marked via table lookup, with which the slot labels are much easier to predict. These features were utilized by some of the previous researchers (Deoras and Sarikaya 2013; Mesnil et al. 2015; Zhang and Wang 2016). To make a comprehensive comparison, we also report the SF results with NE features for ATIS.

As can be seen in Table 4, our model beats all the reported results under this setting, with 0.39% absolute improvement. Compared with results from Table 2, we can see that the named entity feature does significantly improve SF performance, with F1 score arising from 95.91% to 97.28%.

Related Work

SLU - In recent years, RNN-based methods have defined the state-of-the-art in SLU research. Yao et al.(2013) adapted RNN language models to perform SLU, outperforming previous CRF result by a large margin. They attributed the superior performance to the task-specific word representations learned by the RNN. Mesnil et al.(2015) investigated different kinds of RNNs for SF and showed that Elman RNN performed better than Jordan RNN. Yao et al.(2014) used a deep LSTM architecture and investigated the relative importance of each gate in the LSTM by setting other gates to a constant and only learning particular gates.

Joint Method - There have been many attempts to learn ID and SF jointly. Xu and Sarikaya(2013) first proposed a joint model for ID and SF based on convolutional neural network (CNN). Liu and Lane(2016) proposed an attention-based neural network model and beat the state-of-the-art on both tasks. Zhang and Wang(2016) used a GRU-based model and max-pooling method to jointly learn these two tasks. Hakkani-Tür et al.(2016) adopted a multi-domain, multi-task sequence tagging approach. Kim, Lee, and Stratos(2017) proposed the *OneNet* to jointly perform domain, intent and slot prediction. Wang, Shen, and Jin(2018) proposed a bi-model that enables interaction between ID and SF.

Despite their success, most of these models did not explicitly model the interaction between ID and SF and only tied these two tasks through a joint loss function. Goo et al.(2018) pointed this problem out and tackled it with a gating mechanism, leveraging intent vector to influence slots prediction. This idea was later followed by other researchers (Li, Li, and Qi 2018).

However, in their models, the influence is only unidirectional, meaning that they only use the intent vector to influence slot prediction, but not the other way around. By contrast, our model enables bidirectional interaction between ID and SF via the S-LSTM. The similar idea has also been proposed by Niu et al.(2019). However, to model the correlation between ID and SF, their model relies on two addition subnets, while our model can handle it on the fly thanks to the message passing mechanism of S-LSTM.

Graph LSTM - There are many variants of Graph LSTM. Liang et al.(2016) proposed a Graph LSTM network to address the semantic object parsing task. It used the confidence-driven scheme to adaptively select the starting node. Zayats and Ostendorf(2018) adapted Tree-LSTM to graph. However, in their model, each node in the graph has at most two incoming edges. Peng et al.(2017) proposed another variant of the Graph LSTM based on the relation extraction task. Zhang, Liu, and Song(2018) proposed the S-LSTM to improve text encoding.

Of all these variants of Graph LSTM, we believe the S-LSTM (Zhang, Liu, and Song 2018) is especially suited for this task. The most prominent feature of S-LSTM is that it has a special sentence-level node, which stores the information of the whole sentence, making it especially suited to utilize the bi-directional correlation between slot and intent. To the best of our knowledge, we are the first to introduce the Graph structure into the SLU area.

Conclusion

In this paper, we propose a new model to jointly learn SF and ID based on the Graph LSTM. To the best of our knowledge, we are the first to introduce such a graph-based model into this area. Besides, our model effectively utilizes the semantic correlation between slot and intent. To further enhance our model's ability to utilize local and global context, we employ a context gate on top of the S-LSTM, which is composed of a convolution unit and a self-attention unit. Extensive evaluation shows that our model beats the state-of-the-art model by a large margin.

Acknowledgment

Our work is supported by National Natural Science Foundation of China under Grant No.61433015 and the National Key Research and Development Program of China under Grant No.2017YFB1002101. The corresponding author of this paper is Houfeng Wang.

References

Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; Primet, M.; and Dureau, J. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR* abs/1805.10190.

Deng, L.; Tur, G.; He, X.; and Hakkani-Tur, D. 2012. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, 210–215.

Deoras, A., and Sarikaya, R. 2013. Deep belief network based semantic taggers for spoken language understanding. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2713–2717.

Goo, C.-W.; Gao, G.; Hsu, Y.-K.; Huo, C.-L.; Chen, T.-C.; Hsu, K.-W.; and Chen, Y.-N. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *NAACL-HLT*, volume 2, 753–757.

Haffner, P.; Tur, G.; and Wright, J. H. 2003. Optimizing svms for complex call classification. In *ICASSP*, volume 1, I–I.

Hakkani-Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y.-N.; Gao, J.; Deng, L.; and Wang, Y.-Y. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, 715–719.

Hemphill, C. T.; Godfrey, J. J.; and Doddington, G. R. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580.

Kim, Y.-B.; Lee, S.; and Stratos, K. 2017. Onenet: Joint domain, intent, slot prediction for spoken language understanding. *ASRU* 547–553.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Lafferty, J.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML* 8.

Li, C.; Li, L.; and Qi, J. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *EMNLP*.

Liang, X.; Shen, X.; Feng, J.; Lin, L.; and Yan, S. 2016. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, 125–143. Springer.

Lin, J.; Sun, X.; Ma, S.; and Su, Q. 2018. Global encoding for abstractive summarization. In *ACL*.

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

Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM TASLP* 23(3):530–539.

Niu, P.; Haihong, E.; Chen, Z.; and Song, M. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 5467–5471.

Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; and Yih, W.-t. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics* 5:101–115.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Song, L.; Zhang, Y.; Wang, Z.; and Gildea, D. 2018. N-ary relation extraction using graph state lstm. *arXiv preprint arXiv:1808.09101*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Tür, G.; Hakkani-Tür, D. Z.; Heck, L. P.; and Parthasarathy, S. 2011. Sentence simplification for spoken language understanding. *ICASSP* 5628–5631.

Tür, G.; Hakkani-Tür, D. Z.; and Heck, L. P. 2010. What is left to be understood in atis? *2010 IEEE Spoken Language Technology Workshop* 19–24.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, Y.; Shen, Y.; and Jin, H. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. *arXiv preprint arXiv:1812.10235*.

Xu, P., and Sarikaya, R. 2013. Convolutional Neural Network Based Triangular Crf for Joint Intent Detection and Slot Filling. 78–83.

Yao, K.; Zweig, G.; Hwang, M.-Y.; Shi, Y.; and Yu, D. 2013. Recurrent neural networks for language understanding. In *Interspeech*, 2524–2528.

Yao, K.; Peng, B.; Zhang, Y.; Yu, D.; Zweig, G.; and Shi, Y. 2014. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 189–194.

Zayats, V., and Ostendorf, M. 2018. Conversation modeling on reddit using a graph-structured lstm. *Transactions of the Association for Computational Linguistics* 6:121–132.

Zhang, X., and Wang, H. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, 2993–2999.

Zhang, Y.; Liu, Q.; and Song, L. 2018. Sentence-state lstm for text representation. In *ACL*.