A Self-Attentive Model with Gate Mechanism for Spoken Language Understanding

Changliang Li[1], Liang Li[2], Ji Qi[1]
[1]Kingsoft AI Lab
[2]Tsinghua University
[1]{lichangliang, qiji1}@kingsoft.com
[2]liliang17@mails.tsinghua.edu.cn

Abstract

Spoken Language Understanding (SLU), which typically involves intent determination and slot filling, is a core component of spoken dialogue systems. Joint learning has shown to be effective in SLU given that slot tags and intents are supposed to share knowledge with each other. However, most existing joint learning methods only consider joint learning by sharing parameters on surface level rather than semantic level. In this work, we propose a novel self-attentive model with gate mechanism to fully utilize the semantic correlation between slot and intent. Our model first obtains intent-augmented embeddings based on neural network with self-attention mechanism. And then the intent semantic representation is utilized as the gate for labelling slot tags. The objectives of both tasks are optimized simultaneously via joint learning in an end-to-end way. We conduct experiment on popular benchmark ATIS. The results show that our model achieves state-of-the-art and outperforms other popular methods by a large margin in terms of both intent detection error rate and slot filling F1-score. This paper gives a new perspective for research on SLU.

1 Introduction

One long-term goal in artificial intelligence field is to build an intelligent human-machine dialogue system, which is capable of understanding human's language and giving smooth and correct responses. A typical dialogue system is designed to execute the following components: (i) automatic speech recognition converts a spoken query into transcription, (ii) spoken language understanding component analyzes the transcription to extract semantic representations, (iii) dialogue manager interprets the semantic information and decides the best system action, according to which the system response is further generated either as a natural language output(Jurafsky, 2000).

In this paper, we focus on spoken language understanding which is a core component of a spoken dialogue system. It typically involves two major tasks, intent determination and slot filling. Intent determination aims to automatically identify the intent of the user as expressed in natural language. Slot filling aims to extract relevant semantic constituents from the natural language sentence towards achieving a goal.

Usually, intent detection and slot filling are carried out separately. However, separate modeling of these two tasks is constrained to take full advantage of all supervised signals. Joint learning of intent detection and slot filling is worthwhile for three reasons. Firstly, the two tasks usually appear simultaneously in SLU systems. Secondly, the information of one task can be utilized in the other task to promote each other and a joint prediction can be made (Zhang and Wang, 2016). For example, if the intent of a utterance is to find a flight, it is likely to contain the departure and arrival cities, and vice versa. Lastly, slot tags and intents, as semantics representations of user behaviours, are supposed to share knowledge with each other.

Recently, joint model for intent detection and slot filling has achieved much progress. (Xu and Sarikaya, 2013) proposed using CNN based triangular CRF for joint intent detection and slot filling. (Guo et al., 2014) proposed using a recursive neural network that learns hierarchical representations of the input text for the joint task. (Liu and Lane, 2016b) describes a recurrent neural network (RNN) model that jointly performs intent detection, slot filling and language modeling. The neural network models keep updating the intent prediction as word in the transcribed utterance arrives and uses it as contextual features in the joint model.

In this work, we propose a novel model for

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3824–3833
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

joint intent determination and slot filling by introducing self-attention and gating mechanism. Our model can fully utilize the semantic correlation between slot and intent. To the best of our knowledge, this is the first attempt to utilize intent-augmented embedding as a gate to guide the learning of slot filling task. To fully evaluate the efficiency of our model, we conduct experiment on Airline Travel Information Systems (ATIS) dataset (Hemphill et al., 1990), which is popularly used as benchmark in related work. And empirical results show that our independent model outperforms the previous best result by 0.54% in terms of F1-score on slot filling task, and gives excellent performance on intent detection task. Our joint model further promotes the performance and achieves state-of-the-art results on both tasks.

The rest of our paper is structured as follows: Section 2 discusses related work, Section 3 gives a detailed description of our model, Section 4 presents experiments results and analysis, and Section 5 summarizes this work and the future direction.

2 Related Work

There is a long research history for spoken dialogue understanding, which emerged in the 1990s from some call classification systems (Gorin et al., 1997) and the ATIS project. In this section, we describe some typical works on intent classification and slot-filling, which are both core tasks of SLU (De Mori, 2007).

For intent detection task, the early traditional method is to employ n-grams as features with generic entities, such as locations and dates (Zhang and Wang, 2016). This type of method is restricted to the dimensionality of the input space. Another line of popular approaches is to train machine learning models on labeled training data (Young, 2002; Hahn et al., 2011). For example, SVM (Haffner et al., 2003) and Adaboost (Schapire and Singer, 2000) have been explored to improve intent detection. Approaches based on neural network architecture have shown good performance on intent detection task. Deep belief networks (DBNs) have been first used in call routing classification (Deoras and Sarikaya, 2013). More recently, RNNs have shown excellent performance on the intent classification task (Ravuri and Stolcke, 2015).

For slot-filling task, traditional approaches are based on conditional random fields (CRF) architecture, which has strong ability on sequence labelling (Raymond and Riccardi, 2007). Recently, models based on neural network and its extensions have shown excellent performance on the slot filling task and outperform traditional CRF models. For example, (Yao et al., 2013) proposed to take words as input in a standard recurrent neural network language model, and then to predict slot labels rather than words on the output side. (Yao et al., 2014b) improved RNNs by using transition features and the sequence-level optimization criterion of CRF to explicitly model dependencies of output labels. (Mesnil et al., 2013) tried bi-directional and hybrid RNN to investigate using RNN for slot filling. (Yao et al., 2014a) introduced LSTM architecture for this task and obtained a marginal improvement over RNN. Besides, following the success of attention based models in the NLP field, (Simonnet et al., 2015) applied the attention-based encoder-decoder to the slot filling task, but without LSTM cells.

Recently, there has been some work on learning intent detection and slot filling jointly exploited by neural networks. Slot labels and intents, as semantics of user behaviors, are supposed to share knowledge with each other. (Guo et al., 2014) adapted recursive neural networks (RNNs) for joint training of intent detection and slot filling. (Xu and Sarikaya, 2013) described a joint model for intent detection and slot filling based on convolutional neural networks (CNN). The proposed architecture can be perceived as a neural network version of the triangular CRF model (TriCRF). (Hakkani-Tür et al., 2016) proposed a single recurrent neural network architecture that integrates the three tasks (domain detection, intent detection and slot filling for multiple domains) in a model. (Liu and Lane, 2016a) proposed an attention-based neural network model for joint intent detection and slot filling. Their joint model got the best performance of 95.98% slot filling F1-score and 1.57% intent error rate in the ATIS dataset.

Despite the great progress those methods have achieved, it is still a challenging and open task for intent detection and slot filling. Therefore, we are motivated to design a powerful model, which can improve the performance of SLU systems.

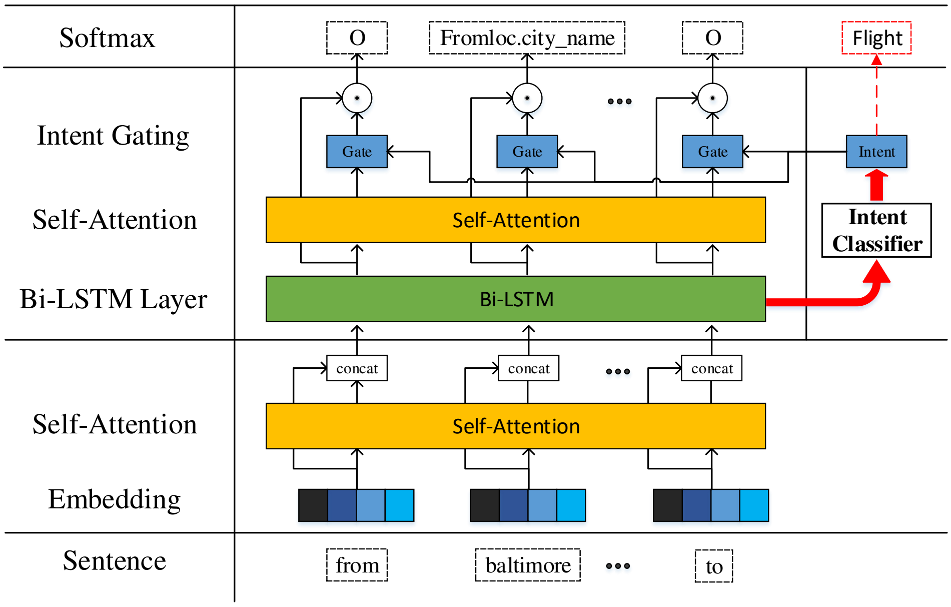

Figure 1: Illustration of our proposed model for joint intent detection and slot filling. Red arrows represent the intent classification task based on the weighted average of BiLSTM outputs. The embeddings coloured with different intensity values denote word-level and char-level embeddings (three kinds of convolution kernels).

3 Model

In this section, we present our model for the joint learning of intent detection and slot filling. Figure 1 gives an overview of our model.

The first layer maps input sequence into vectors by concatenating its word-level embeddings and character-level embeddings (obtained by convolution). And we use these vectors as merged embeddings in downstream layers. In many situations, contextual information is useful in sequence labelling. In this paper, we introduce an approach that leverages context-aware features at each time step. In particular, we make use of self-attention to produce context-aware representations of the embeddings. Then a bidirectional recurrent layer takes as input the embeddings and context-aware vectors to produce hidden states. In the last step, we propose to exploit the intent-augmented gating mechanism to match the slot label. The gate for a specific word is obtained by taking a linear transformation of the intent embedding and another contextual representation of this word computed by self-attention. We apply element-wise dot-product between the gate and each BiLSTM output.

Finally, a softmax layer is added to classify the slot labels on top of the gate layer. For simplicity, we only take the weighted average of BiLSTM outputs to predict the intent label.

The design of this structure is motivated by the effectiveness of multiplicative interaction among vectors and by self-attention mechanism which has been used successfully in a variety of tasks (Cheng et al., 2016; Vaswani et al., 2017; Lin et al., 2017). It also typically corresponds to our finding that the intent is highly correlated with slot label in some cases, so the semantics of intent should be useful for detecting the slot labels.

3.1 Embedding Layer

We first convert the indexed words $\mathbf{w} = (w_1, w_2, ..., w_T)$ to word-level embeddings $E^w = [e^w_1, e^w_2, ..., e^w_T]$, and character-level embeddings $E^c = [e^c_1, e^c_2, ..., e^c_T]$. Although word embeddings are sufficient for many NLP task, provided by a well-pretrained glove[1] or word2vec[2], character-level information provides some more prior knowledge (e.g. morphemes) to the embedding learning procedure. Some morphemic correlated words are more close in vector space, which is useful for identifying the slot labels. Character embeddings also alleviate the out-of-vocabulary (OOV) problem in the testing phase. In this paper we focus on a character-aware convolution layer used in (Kim et al., 2016) for words. The

[1] http://nlp.stanford.edu/projects/glove/

[2] https://code.google.com/p/word2vec/

character-level embeddings are generated by convolution over characters in the word with multiple window size to extract n-gram features.

Let $\mathcal{C}$ be the vocabulary of characters, $\mathcal{V}$ be the vocabulary of words. The dimensions of character-level embedding and word-level embedding are denoted as d_c and d_w, respectively. For each word $w_t \in \mathcal{V}$, characters in w_t constitute the matrix $C^t \in \mathbb{R}^{d_c \times l}$, where the columns corresponds to l character embeddings.

A narrow convolution is applied between C^t and a filter (or kernel) $H \in \mathbb{R}^{d_c \times w}$. Here we suppose the filter width is w. After that, we obtain a feature map $\mathbf{f}^t \in \mathbb{R}^{l-w+1}$ by adding a nonlinearity activation. The final n-gram features is generated by taking the max-over-time:

$$\mathbf{f}^t[i] = relu(H \cdot C^t[:, i : i + w - 1] + b) \quad (1)$$

$$c^t = \max_i \mathbf{f}^t[i] \quad (2)$$

where $C^t[:, i : i + w - 1]$ is the i-to-(i+w-1)-th column of C^t, and the character-level embedding e_t^c is made up of multiple c^t generated by different convolution kernels.

3.2 Self-Attention

Attention mechanism is usually used to guide the forming of sentence embedding, extra knowledge is also used to weigh the CNN or LSTM hidden states (i.e. document words sometimes attend to question information). However in slot filling task, the input to our model is just one sequence. So the attention mechanism used here is called self-attention, that is to say, the word at each time step attends to the whole words in this sentence. And it helps to determine which region is likely to be a slot. Since the embedding at each time step consists of multiple parts (i.e. word embedding and character embeddings of different kernel width), each part has its own semantic meaning. As shown in Figure 2, we divide the embedding into multiple parts and the attention of each part is processed within its corresponding dimension. In this approach, we restrict the interaction among different aspects of the embedding. We hypothesize that different semantic parts are relatively independent and play different roles in our network.

Suppose $M \in \mathbb{R}^{d_m \times T}$ to be the matrix containing sentence hidden vectors $[m_1, ..., m_T]$, where d_m is the dimension of these T vectors. Considering the characteristics of slot filing task, our aim is to encode each hidden vector into a context-aware representation. We achieve that by using attention over all the sentence hidden vectors M. Firstly, We linearly map all the vectors in M to three feature spaces by different projection parameters W_a, W_b and W_c, so the resulting vectors are expressed as M_a, M_b and M_c with the same shape as M. These matrices are shared across all time steps. Considering the structure of embedding which consists of K different parts (we use 4 kinds of embeddings with the same dimension), these transformed matrices are equally split into K parts. Furthermore, the attention weight is computed by dot product between M_a and M_b. Lastly, the attention output is a weighted sum of M_c.

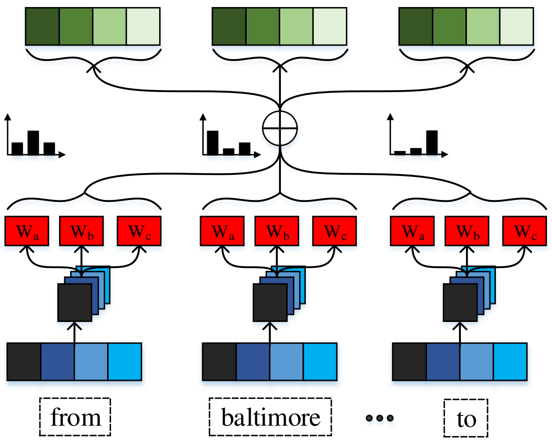

Figure 2: The structure of self-attention layer. Red coloured rectangles stand for matrices which map the input to different subspaces. These transformed vectors are divided into multiple parts for computing self-attention.

Specifically, we consider different K parts in detail for $k = 1, .., K$:

$$\begin{bmatrix} M_a \\ M_b \\ M_c \end{bmatrix} = \begin{bmatrix} W_a M \\ W_b M \\ W_c M \end{bmatrix} \quad (3)$$

$$\alpha_{k,t} = \text{softmax}(m_{k,a,t}^{\mathrm{T}} M_{k,b}) \quad (4)$$

$$\text{S-Att}(m_t, M) = [M_{1,c}\alpha_{1,t}^{\mathrm{T}}, ..., M_{K,c}\alpha_{K,t}^{\mathrm{T}}] \quad (5)$$

where $M_{k,a} \in \mathbb{R}^{(d_m/K) \times T}$ is the k-th part of M_a which is transformed from M by W_a. Index t is word position ranging over T time steps and $m_{k,a,t} \in \mathbb{R}^{d_m/K}$ is the t-th column of $M_{k,a}$. $\alpha_{k,t}$

is the attention weights over $M_{k,c}$. The output of self-attention module generated at time step t is the concatenation of K parts by using Equation 5.

3.3 BiLSTM

Character embeddings and word embeddings are both important features in our task. To further utilize these features, we associate each embedding with a context-aware representation which is typically implemented by self-attention mechanism. For current word w_t, the input of the recurrent layer at time step t is represented as x_t:

$$e_t^a = \text{S-Att}([e_t^c, e_t^w], E) \tag{6}$$
$$x_t = [e_t^c, e_t^w, e_t^a] \tag{7}$$

e_t^a is the context-aware vector of w_t which is obtained by applying self-attention mechanism on the concatenated embeddings $E = [e_1^c \parallel e_1^w, ..., e_T^c \parallel e_T^w]$.

It was difficult to train RNNs to capture long-term dependencies because the gradients tend to either vanish or explode. Therefore, some more sophisticated activation functions with gating units were designed. We use LSTM (Hochreiter and Schmidhuber, 1997) in this work:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{8}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{9}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{10}$$
$$\widetilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{11}$$
$$c_t = i_t \odot \widetilde{c}_t + f_t \odot c_{t-1} \tag{12}$$
$$h_t = o_t \odot tanh(c_t) \tag{13}$$

Where $\odot$ denotes element-wise product of two vectors. To consider both the previous history and the future history, we use BiLSTM as encoder in advance. The bi-directional LSTM (BiLSTM), a modification of the LSTM, consists of a forward and a backward LSTM. The encoder reads the input vectors $\mathbf{x} = (x_1, x_2, ..., x_T)$ and generates T hidden states by concatenating the forward and backward hidden states of BiLSTM:

$$\overrightarrow{h}_t = \overrightarrow{LSTM}(x_t, \overrightarrow{h}_{t-1}) \tag{14}$$
$$\overleftarrow{h}_t = \overleftarrow{LSTM}(x_t, \overleftarrow{h}_{t+1}) \tag{15}$$
$$h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t] \tag{16}$$

where $\overleftarrow{h}_t$ is the hidden state of backward pass in BLSTM and $\overrightarrow{h}_t$ is the hidden state of forward pass in BLSTM at time t.

3.4 Intent-Augmented Gating Mechanism

As described above, intent information is useful for slot filling task. To measure the probability of words in target slots and attend to the ones relevant to the intent, we add a gate to the output of BiLSTM layer. Let $H \in \mathbb{R}^{2d \times T}$ be a matrix consisting of hidden vectors $[h_1, ..., h_T]$ produced by BiLSTM. For each word, we use self-attention mechanism to form another context-aware representation, the gate vector h_t^* is calculated by linearly transforming the concatenation of the context-aware representation and the intent embedding vector v^{int} with a multi-layer perceptron (MLP) network. The intent label is provided by correct label during training phase, and by the output from intent classification layer in the test phase. Specifically, for $t = 1, ...T$:

$$s_t = \text{Self-Attention}(h_t, H) \tag{17}$$
$$h_t^* = \text{MLP}([s_t, v^{int}]) \tag{18}$$
$$o_t = h_t \odot h_t^* \tag{19}$$

We use element-wise multiplication to model the interaction between BiLSTM outputs and the gate vector.

3.5 Task Learning

The bidirectional recurrent layer converts a sequence of words $\mathbf{w} = (w_1, w_2, ..., w_T)$ into hidden states $H = [h_1, ..., h_T]$ which are shared by two tasks. We use simple attention pooling function denoted as f_{att} over H to get an attention-sum vector for intent label classification. The classified label y^{int} is transformed to an embedding v^{int} by matrix E^{int} for gate computing.

$$h^{int} = f_{att}(H) \tag{20}$$
$$y^{int} = \text{softmax}(W^{int} h^{int} + b^{int}) \tag{21}$$

During the training phase, model parameters are updated w.r.t. a cross-entropy loss between the predicted probabilities and the true label. The label with maximum probability will be selected as the predicted intent during the testing phase.

For another task, the hidden states processed by our gating layer are used for predicting slot labels.

$$y_t^{slot} = \text{softmax}(W^{slot} o_t + b^{slot}) \tag{22}$$

Slot filling can be defined as a sequence labelling problem which is to map a utterance sequence $\mathbf{w} = (w_1, ..., w_T)$ to its corresponding slot label sequence $\mathbf{y} = (y_1, ..., y_T)$. The objective is to maximize the likelihood of a sequence:

$$P(\mathbf{y}^{slot}|\mathbf{w}) = \prod_{t=1}^{T} P(y_t^{slot}|\mathbf{w}) \quad (23)$$

It is equal to minimize Negative Log-likelihood (NLL) of the correct labels for the predicted sequence $\mathbf{y}^{slot}$.

4 Experiments

4.1 Dataset

In order to evaluate the efficiency of our proposed model, we conduct experiments on ATIS (Airline Travel Information Systems) dataset, which is widely used as benchmark in SLU research (Price, 1990).

Figure 3 gives one example of sentence in ATIS dataset. The words are labelled with their value according to certain semantic frames. The slot labels of the words are represented in an In-Out-Begin (IOB) format and the intent is highlighted with a box surrounding it.

Sentence	Show	flights	from	Boston	to	New	York	today
Slots	O	O	O	B-FromCity	O	B-ToCity	I-ToCity	B-Date

Figure 3: Example of sentence annotated by slots sampled from ATIS corpus, the black boxed word indicates the intent.

In this paper, we use the ATIS corpus setting following previous related works (Liu and Lane, 2016a; Mesnil et al., 2015; Liu and Lane, 2015; Xu and Sarikaya, 2013; Tur et al., 2010). The training set contains 4978 utterances from ATIS-2 and ATIS-3 datasets, and test set contains 893 utterances from ATIS-3 NOV93 and DEC94 datasets. The number of slot labels is 127 and the intent has 18 different types.

4.2 Metrics

The performance of slot filling task is measured by the F1-score, while intent detection task is evaluated with prediction error rate that is the ratio of the incorrect intent of the test data.

4.3 Training Details

We preprocess the ATIS following (Yao et al., 2013; Liu and Lane, 2016a). To deal with unseen words in the test set, we mark those words that appear only once in the training set as ⟨UNK⟩, and use this label to represent those unseen words in the test set. Besides, each number is converted to the string DIGIT.

The model is implemented in the Tensorflow framework (Abadi et al., 2016). At training stage, we use LSTM cell as suggested in (Sutskever et al., 2014) and the cell dimension d is set to be 128 for both the forward and backward LSTM.

We set the dimension of word embedding d_w to be 64 and the dimension of character embedding d_c to be 128. We generate three character-level embeddings using multiple widths and filters (the convolution kernel width $w \in \{2, 3, 4\}$ with 64 filters each) followed by a max pooling layer over time. Then, the dimension of concatenated embeddings is 256. We make the dimensions of each parts equal for the convenience of dimension splitting during the self-attention in later stage. All the parameters in the network are randomly initialized with uniform distribution (Sussillo and Abbott, 2014) which are fine-tuned during training. We use the stochastic gradient descent algorithm (SGD) for updating parameters. And the learning rate is controlled by Adam algorithm (Kingma and Ba, 2014). The model is trained on all the training data with mini-batch size of 16. In order to enhance our model to generalize well, the maximum norm for gradient clipping is set to 5. We also apply layer normalization (Ba et al., 2016) on the self-attention layer after we add a residul connection between the output and input. Meanwhile, dropout rate 0.5 is applied on recurrent cell projection layer (Zaremba et al., 2014) and on each attention activation.

4.4 Independent Learning

The results of separate training for slot filling and intent detection are reported in Table 1 and Table 2 respectively. On the independent slot filling task, we fixed the intent information as the ground truth labels in the dataset. But on the independent intent detection task, there is no interaction with slot labels.

Table 1 compares F1-score of slot filling between our proposed architecture and some previous works. Our model achieves state-of-the-art results and outperforms previous best model by 0.56% in terms of F1-score. We attribute the improvement of our model to the following reasons: 1) The attention used in (Liu and Lane, 2016a) is vanilla attention, which is used to compute the de-

Methods	F1-score
CRF (Mesnil et al., 2013)	92.94
simple RNN (Yao et al., 2013)	94.11
CNN-CRF (Xu and Sarikaya, 2013)	94.35
LSTM (Yao et al., 2013)	94.85
RNN-SOP (Liu and Lane, 2015)	94.89
Deep LSTM (Yao et al., 2013)	95.08
RNN-EM (Peng et al., 2015)	95.25
Bi-RNN with Ranking Loss (Vu et al., 2016)	95.47
Encoder-labeler Deep LSTM (Kurata et al., 2016)	95.66
Attention BiRNN (Liu and Lane, 2016a)	95.75
BLSTM-LSTM (focus) (Zhu and Yu, 2017)	95.79
Our Model	**96.35**

Table 1: Results of independent training for slot filling in terms of F1-score.

coding states. It is not suitable for our model since the embeddings are composed of several parts. Self-attention allows the model to attend to information jointly from different representation parts, so as to better understand the utterance. 2) intent-augmented gating layer connects the semantics of sequence slot labels, which captures complex interactions between the two tasks.

Table 2 compares the performance of our proposed model to previously reported results on intent detection task. Our model gives good performance in terms of classification error rate, but not as good as Attention Encoder-Decoder (with aligned inputs) method (Liu and Lane, 2016a). As their published state-of-the-art result described in (Liu and Lane, 2016a), their attention-based model is based on word-level embeddings. While in our model, we introduce character-level embeddings to improve the performance of joint learning. But independent learning for intent classification aims at capturing the global information of an utterance, not caring much about the details of specific word. The character-level embeddings introduced in our model bring very little hurt to independent learning of intent detection, as a trade-off in performance between both criterion.

4.5 Joint Learning

We compare our model against the following baseline models based on joint learning:

Methods	Error(%)
Recursive NN (Guo et al., 2014)	4.60
Boosting (Tur et al., 2010)	4.38
Boosting + Simplified sentences (Tur et al., 2011)	3.02
Attention Enc-Dec (Liu and Lane, 2016a)	**2.02**
Our Model	2.69

Table 2: Results of independent training for intent detection in terms of error rate.

Methods	F1	Error(%)
Recursive NN (Guo et al., 2014)	93.22	4.60
Recursive NN+Viterbi (Guo et al., 2014)	93.96	4.60
Attention Enc-Dec (Liu and Lane, 2016a)	95.87	1.57
Attention BiRNN (Liu and Lane, 2016a)	95.98	1.79
Our Model	**96.52**	**1.23**

Table 3: Results of joint training for slot filling and intent detection.

- **Recursive NN:** (Guo et al., 2014) employed recursive neural networks for joint training of two tasks.

- **Recursive NN + Viterbi:** (Guo et al., 2014) applied the Viterbi algorithm on Recursive NN to improve the result on slot filling.

- **Attention Enc-Dec:** (Liu and Lane, 2016a) proposed Attention Encoder-Decoder (with aligned inputs) which introduced context vector as the explicit aligned inputs at each decoding step.

- **Attention BiRNN:** (Liu and Lane, 2016a) introduced attention to the alignment-based RNN sequence labeling model. Such attention provides additional information to the intent classification and slot label prediction.

Table 3 compares our joint model with reported results from previous works. We can see that our model achieves state-of-the-art results and outperforms previous best result by 0.54% in terms of F1-score on slot filling, and by 0.34% in terms of error rate on intent detection. This improvement is statistically significant. Besides, the joint learning

Methods	F1-Score	Error(%)
W/O char-embedding	96.30	1.23
W/O self-attention	96.26	1.34
W/O attention-gating	96.25	1.46
Full Model	**96.52**	**1.23**

Table 4: Feature ablation comparison of our proposed model on ATIS. slot filling and intent detection result are shown each row after after we exclude each feature from the full architecture

achieves better results than separate learning. It can be interpreted that the two tasks are highly correlated and boost the performance each other. The slot filling task enables the model to learn more meaningful representations which give more supervisory signals for the learning of shared parameters. Similarly, intent is also useful to determine the slot label.

4.6 Ablation Study

The ablation study is performed to evaluate whether and how each part of our model contributes to our full model. To further evaluate the advances of our gating architecture for joint learning, we ablate some techniques used in our model. We ablate three important components and conduct different approaches in this experiment. Note that all the variants are based on joint learning with intent-augmented gate:

- W/O char-embedding, where no character embeddings are added to the embedding layer. The embedding layer is composed of word embeddings only.

- W/O self-attention, where no self-attention is modelled after the embedding layer and in the intent-augmented gating layer. The intent gate is computed by the output of BiLSTM and intent embedding.

- W/O attention-gating, where no self-attention mechanism is performed in the intent-augmented gating layer. The gate is computed by the output of BiLSTM and intent embedding. But we still use the self-attention on top of embedding layer to augment the context information.

Table 4 shows the joint learning performance of our model on ATIS data set by removing one module at a time. We find that all variants of our model perform well based on our gate mechanism. As listed in the table, all features contribute to both slot filling and intent classification task.

If we remove the self-attention from the holistic model or just in the intent-augmented gating layer, the performance drops dramatically. The result can be interpreted that self-attention mechanism computes context representation separately and enhances the interaction of features in the same aspect. We can see that self-attention does improve performance a lot in a large scale, which is consistent with findings of previous work (Vaswani et al., 2017; Lin et al., 2017).

If we remove character-level embeddings and only use word-level embeddings, we see 0.22% drop in terms of F1-score. Though word-level embeddings represent the semantics of each word, character-level embeddings can better handle the out-of-vocabulary (OOV) problem which is essential to determine the slot labels.

5 Conclusion

In this paper, we propose a novel self-attentive model gated with intent for spoken language understanding. We apply joint learning on both intent detection and slot filling tasks. In our model, self-attention mechanism is introduced to better represent the semantic of utterance, and gate mechanism is introduced to make full use of the semantic correlation between slot and intent. Experiment results on ATIS dataset have shown efficiency of our model and outperforms the state-of-the-art approach on both tasks. Besides, our model also shows consistent performance gain over the independent training models. In future works, we plan to improve our model by introducing extra knowledge.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Renato De Mori. 2007. Spoken language understanding: A survey. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 365–376. IEEE.

Anoop Deoras and Ruhi Sarikaya. 2013. Deep belief network based semantic taggers for spoken language understanding. In *Interspeech*, pages 2713–2717.

Allen L Gorin, Giuseppe Riccardi, and Jeremy H Wright. 1997. How may i help you? *Speech communication*, 23(1):113–127.

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 554–559. IEEE.

Patrick Haffner, Gokhan Tur, and Jerry H Wright. 2003. Optimizing svms for complex call classification. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–I. IEEE.

Stefan Hahn, Marco Dinarelli, Christian Raymond, Fabrice Lefevre, Patrick Lehnen, Renato De Mori, Alessandro Moschitti, Hermann Ney, and Giuseppe Riccardi. 2011. Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583.

Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *INTERSPEECH*, pages 715–719.

Charles T Hemphill, John J Godfrey, George R Doddington, et al. 1990. The atis spoken language systems pilot corpus. In *Proceedings of the DARPA speech and natural language workshop*, pages 96–101.

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

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Gakuto Kurata, Bing Xiang, Bowen Zhou, and Mo Yu. 2016. Leveraging sentencelevel information with encoder lstm for natural language understanding. *arXiv preprint*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Bing Liu and Ian Lane. 2015. Recurrent neural network structured output prediction for spoken language understanding. In *Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions*.

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

Bing Liu and Ian Lane. 2016b. Joint online spoken language understanding and language modeling with recurrent neural networks. *arXiv preprint arXiv:1609.01462*.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539.

Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *Interspeech*, pages 3771–3775.

Baolin Peng, Kaisheng Yao, Li Jing, and Kam-Fai Wong. 2015. Recurrent neural networks with external memory for spoken language understanding. In *Natural Language Processing and Chinese Computing*, pages 25–35. Springer.

Patti J Price. 1990. Evaluation of spoken language systems: The atis domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Suman V Ravuri and Andreas Stolcke. 2015. Recurrent neural network and lstm models for lexical utterance classification. In *INTERSPEECH*, pages 135–139.

Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Eighth Annual Conference of the International Speech Communication Association*.

Robert E Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168.

Edwin Simonnet, Nathalie Camelin, Paul Deléglise, and Yannick Estève. 2015. Exploring the use of attention-based recurrent neural networks for spoken language understanding. In *Machine Learning for Spoken Language Understanding and Interaction NIPS 2015 workshop (SLUNIPS 2015)*.

David Sussillo and LF Abbott. 2014. Random walk initialization for training very deep feedforward networks. *arXiv preprint arXiv:1412.6558*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

Gokhan Tur, Dilek Hakkani-Tür, Larry Heck, and Sarangarajan Parthasarathy. 2011. Sentence simplification for spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5628–5631. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6060–6064. IEEE.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 78–83. IEEE.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014a. Spoken language understanding using long short-term memory neural networks. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 189–194. IEEE.

Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. 2014b. Recurrent conditional random field for language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4077–4081. IEEE.

Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. 2013. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528.

Steve J Young. 2002. Talking to machines (statistically speaking). In *INTERSPEECH*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

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

Su Zhu and Kai Yu. 2017. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5675–5679. IEEE.