Dual Learning for Semi-Supervised Natural Language Understanding

Su Zhu, Ruisheng Cao, and Kai Yu

***Abstract*—Natural language understanding (NLU) converts sentences into structured semantic forms. The paucity of annotated training samples is still a fundamental challenge of NLU. To solve this data sparsity problem, previous work based on semi-supervised learning mainly focuses on exploiting unlabeled sentences. In this work, we introduce a dual task of NLU, semantic-to-sentence generation (SSG), and propose a new framework for semi-supervised NLU with the corresponding dual model. The framework is composed of dual pseudo-labeling and dual learning method, which enables an NLU model to make full use of data (labeled and unlabeled) through a closed-loop of the primal and dual tasks. By incorporating the dual task, the framework can exploit pure semantic forms as well as unlabeled sentences, and further improve the NLU and SSG models iteratively in the closed-loop. The proposed approaches are evaluated on two public datasets (ATIS and SNIPS). Experiments in the semi-supervised setting show that our methods can outperform various baselines significantly, and extensive ablation studies are conducted to verify the effectiveness of our framework. Finally, our method can also achieve the state-of-the-art performance on the two datasets in the supervised setting.**

***Index Terms*—Natural language understanding, semi-supervised learning, dual learning, slot filling, intent detection.**

I. Introduction

RECENTLY, the development of mobile internet and smart devices has led to the tremendous growth of conversational dialogue systems, such as Amazon Alexa, Google Assistant, Apple Siri, and Microsoft Cortana. Natural language understanding (NLU) is a key component of these systems, parsing user's utterances into the corresponding semantic forms [1] for certain narrow domain (e.g., *booking hotel*, *searching flight*). Typically, the primary task of the NLU module in goal-oriented dialogue systems usually contains two sub-tasks: intent detection and slot filling [1], [2], [3], [4], [5], [6], [7]. The intent detection is typically treated as a sentence classification problem [8], [9], [10], while the slot filling is typically treated as a sequence labeling problem in which contiguous sequences of words are tagged with semantic labels (slots) [11], [12], [13], [14].

Deep learning has achieved great success for the intent detection and slot filling in NLU [2], [3], [4], [5], [6], [7], [14], [15], [16], [17], [18], [19], outperforming most traditional approaches [13], [20] in the field of supervised learning.

Su Zhu, Ruisheng Cao and Kai Yu are with the SpeechLab and MoE Key Lab of Artificial Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. (*Su Zhu and Ruisheng Cao contribute equally to this article.*) (*Corresponding authors*: Kai Yu.) (e-mail: paul2204@sjtu.edu.cn; 211314@sjtu.edu.cn; kai.yu@sjtu.edu.cn)

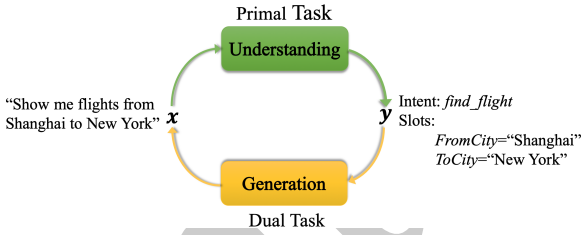

Fig. 1. A diagram of NLU and its dual task. The primal task is NLU, which converts an input sentence into the corresponding intent and slots. In the inverse direction, the dual task is semantic-to-sentence generation (SSG), which converts the intent and slots into a natural language sentence.

However, the deep learning method is notorious for requiring large labeled data, which limits the scalability of NLU models to new domains due to the annotation cost. Semi-supervised learning methods are adopted to solve this data sparsity problem of NLU, which utilize a large number of unannotated sentences to enhance the supervised NLU training [21], [22], [23], [24]. These semi-supervised learning methods focus on exploiting the unlabeled sentences to enhance input encoders or create additional samples with predicted pseudo-labels.

Apart from *pure* sentences (i.e., unannotated sentences), *pure* semantic forms (i.e., intents and slots without sentence expressions) can also be utilized in the semi-supervised NLU. Exploiting semantic forms could be more affordable and effective than collecting in-domain sentences, since they are well-structured and could be automatically created or synthesized under domain knowledge. However, the previous methods of semi-supervised NLU cannot utilize pure semantic forms data.

In this work, we introduce the dual task of intent detection and slot filling in NLU, as shown in Fig. 1. By incorporating the dual task, a novel framework of semi-supervised NLU is proposed, which can utilize not only pure sentences but also pure semantic forms (i.e., intents and slots). Our framework consists of two parts: a dual pseudo-labeling method and dual learning algorithm. 1) Besides using a primal model to generate pseudo labels [22] for unlabeled sentences, the dual pseudo-labeling method also utilizes a dual model to generate pseudo sentences for pure semantic forms. Next, we combine these pseudo-labeled samples with the labeled dataset to retrain both the primal and dual models iteratively. 2) Furthermore, the dual learning algorithm [25] is applied to train the primal and dual models jointly in a closed-loop of the two models. New validity rewards are proposed to validate potential sentences and semantic forms.

The main contributions of this paper are summarized:

- A dual model for joint intent detection and slot filling

in NLU is introduced to generate sentences based on structured semantic forms.
- We propose a novel framework for semi-supervised NLU by incorporating the dual model, which can better utilize unlabeled data.
- We present extensive experiments on ATIS [26] and SNIPS [27] datasets, which demonstrate the benefit of our proposed framework for semi-supervised NLU. It also achieves the state-of-the-art performance in the supervised setting.

The rest of the paper is organized as follows. The following section discusses related works. We introduce the intent detection and slot filling in NLU in Section III, then describe the details of the dual task in Section IV. A semi-supervised NLU framework with the dual task is proposed in Section V. Detailed experimental results and analysis are given in Section VI. Section VII summarizes this work and the future direction.

II. Related Work

This section describes previous literature of intent detection and slot filling in NLU as well as the semi-supervised NLU.

A. Intent Detection and Slot Filling in NLU

Recently, motivated by a number of successful neural network and deep learning methods in natural language processing, many neural network architectures have been applied in the intent detection and slot filling, such as vanilla recurrent neural network (RNN) [14], [28], [29], [30], convolutional neural network (CNN) [16], [2], [31], long short-term memory (LSTM) [15], [4], [32], [33], encoder-decoder [18], [3], [34], [17], capsule neural networks [35], transformers [36], etc. Several pre-trained language models are also applied to improve generalization, like ELMo [37] and BERT [6], [38]. Most of the previous work tends to share the encoders of the intent detection and slot filling while leaves their decoders (e.g., classification layers) independent. Besides, some investigations focus on interrelated modeling of intent detection and slot filling [19], [5], [7], [39], which is orthogonal to the semi-supervised learning of NLU.

B. Semi-supervised NLU

The traditional approaches of semi-supervised NLU utilize unlabeled sentences to improve NLU performances in two ways. 1) NLU model trained with the existing labeled sentences is exploited to predict pseudo-labels for unlabeled sentences, which can be used to retrain the NLU model [21], [22], [23]. 2) Except for the pseudo-labeling method, some prior works design several unsupervised tasks to make use of the unlabeled sentences, like language models [40], [41], [24], [37], sequence-to-sequence based sentence reconstruction [42], [43]. They share partial parameters between the unsupervised tasks and the NLU task. However, we are the first to exploit pure semantic forms (without sentence expressions) by developing a dual pseudo-labeling method.

The dual learning algorithm is first proposed for neural machine translation [25], where translation from the target language to source language (i.e., back-translation) is the dual task. The dual learning is also applied in semantic parsing [44], [45] and natural language understanding [45]. As the most similar work, Su et al. [45] propose a dual supervised learning method for natural language understanding and generation. However, their method is not compatible with a semi-supervised problem. Moreover, they simplify the NLU task into a multi-label classification problem, which is not scalable. We are the first to propose a dual task for intent detection and slot filling in NLU and utilize the dual task in semi-supervised NLU.

III. Intent Detection and Slot Filling in NLU

This section introduces the NLU task and describes the basic multitask framework of intent detection and slot filling.

A. NLU Task Formulation

Intent detection and slot filling are major tasks of NLU in task-oriented dialogue systems. An intent is a purpose or a goal that underlies a user-generated utterance [46]. Therefore, intent detection can be seen as a classification problem to determine the intent label of an input sentence. Slot filling aims to automatically extract a set of attributes or "slots", with the corresponding values. It is typically treated as a sequence labeling problem. An example of data annotation is provided in Fig. 2. The user's intent is to find flights. For slot annotation, it follows the popular inside/outside/beginning (IOB) schema, where *Boston* and *New York* are the departure and arrival cities specified as the slot values in the user's utterance, respectively. In this work, we use the word *tag* as an alias for *slot* to denote semantic labels in IOB schema.

Sentence	Show	me	flights	from	*Shanghai*	to	*New*	*York*
Slots	O	O	O	O	B-FromCity	O	B-ToCity	I-ToCity
Intent	Find_Flight							

Fig. 2. An example of intent and slot annotation (IOB format) in ATIS dataset.

Let $x = (x_1, \cdots, x_{|x|})$ denote an input sentence (word sequence), o^I denote its intent label, and $o^S = (o_1^S, \cdots, o_{|x|}^S)$ denote its output sequence of slot tags, where $|x|$ is the sequence length. Each $o_i^S \in \mathcal{T}$ and $o^I \in \mathcal{I}$, where $\mathcal{T}$ and $\mathcal{I}$ are the sets of all possible slot tags and intent labels respectively in the current domain. Therefore, the intent detection and slot filling in NLU are to estimate $p(o^I, o^S|x)$, the joint posterior probability of intent o^I and slot sequence o^S given input x. Usually, the two sub-tasks are modelled independently, i.e.,

$$p(\widetilde{y}|x) = p(o^I, o^S|x) = p(o^I|x)p(o^S|x) \tag{1}$$

where $\widetilde{y} = (o^I, o^S)$.

B. Preliminaries for Neural Network

Before providing details of the NLU model, we first introduce two basic NN modules for conciseness.

BLSTM: As mentioned before, many neural network architectures have been applied in intent detection and slot filling tasks. In this paper, bi-directional LSTM based RNN

(BLSTM) is adopted for sequence encoding. Given a sequence of feature vectors $(\mathbf{e}_1, \cdots, \mathbf{e}_L)$, hidden vectors are recursively computed at the i-th time step ($i \in \{1, \cdots, L\}$) via

$$\overrightarrow{\mathbf{h}}_i = \mathrm{f}_{\mathrm{LSTM}}(\mathbf{e}_i, \overrightarrow{\mathbf{h}}_{i-1}); \overleftarrow{\mathbf{h}}_i = \mathrm{f}_{\mathrm{LSTM}}(\mathbf{e}_i, \overleftarrow{\mathbf{h}}_{i+1}) \quad (2)$$

and $\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i$, where $\oplus$ denotes the vector concatenation and $\mathrm{f}_{\mathrm{LSTM}}$ is the LSTM function. For convenience, we rewrite the entire operation as a mapping BLSTM_Θ:

$$(\mathbf{h}_1, \cdots, \mathbf{h}_L) \leftarrow \mathrm{BLSTM}_\Theta(\mathbf{e}_1, \cdots, \mathbf{e}_L) \quad (3)$$

Attention Mechanism: Attention mechanism [47], [48] is usually used to obtain a sequence-level feature vector or context representations in encoder-decoder architectures. Given a sequence of feature vectors $(\mathbf{e}_1, \cdots, \mathbf{e}_L)$ and a query vector $\mathbf{q}$, the attention weight for $\mathbf{q}$ with each $\mathbf{e}_i$ ($i \in \{1, \cdots, L\}$) is $a_i = \exp(u_i)/\sum_{j=1}^{L} \exp(u_j)$, and

$$u_i = \mathbf{v}_a^\top \tanh(\mathbf{W}_a(\mathbf{q} \oplus \mathbf{e}_i)) \quad (4)$$

where $\mathbf{v}_a$ and $\mathbf{W}_a$ are learnable parameters. Finally, a context vector is computed as $\mathbf{z} = \sum_{i=1}^{L} a_i \mathbf{e}_i$. For brevity, we rewrite the entire operation as a mapping returning the context vector and attention weights:

$$\mathbf{z}, (a_1, \cdots, a_L) \leftarrow \mathrm{ATTN}_\Theta(\mathbf{q}, (\mathbf{e}_1, \cdots, \mathbf{e}_L)) \quad (5)$$

C. NLU Model Architecture

The basic multitask framework of intent detection and slot filling is comprised of three modules: sentence encoding, intent classification, and slot tagging.

Sentence Encoding: Every input word is mapped to a vector via $\mathbf{x}_i = \mathbf{W}_x \mathbf{o}(x_i)$, where $\mathbf{W}_x$ is an embedding matrix and $\mathbf{o}(x_i)$ a one-hot vector. An BLSTM encoder is applied to get hidden vectors $\mathbf{h}_i^{\mathrm{sen}} \in \mathbb{R}^{2n}$ (n is the hidden size, $i \in \{1, \cdots, |x|\}$):

$$(\mathbf{h}_1^{\mathrm{sen}}, \cdots, \mathbf{h}_{|x|}^{\mathrm{sen}}) \leftarrow \mathrm{BLSTM}_{\Theta_1}(\mathbf{x}_1, \cdots, \mathbf{x}_{|x|}) \quad (6)$$

Intent Classification: An attention model is applied to gather a sentence embedding, and then feed it into a linear output layer for intent classification:

$$\mathbf{z}^{\mathrm{sen}}, (a_1^{\mathrm{sen}}, \cdots, a_{|x|}^{\mathrm{sen}}) \leftarrow \mathrm{ATTN}_{\Theta_2}(\overleftarrow{\mathbf{h}}_1^{\mathrm{sen}}, (\mathbf{h}_1^{\mathrm{sen}}, \cdots, \mathbf{h}_{|x|}^{\mathrm{sen}})) \quad (7)$$

$$p(o^I|x) = \mathrm{softmax}_{o^I}(\mathbf{W}_1 \mathbf{z}^{\mathrm{sen}}) \quad (8)$$

where $\mathbf{W}_1 \in \mathbb{R}^{|\mathcal{I}| \times 2n}$ is trainable (bias is omitted).

Slot Tagging: Slot filling is considered as a sequence labeling problem which tags each input word sequentially. There are three typical methods for slot tagging concerning the time series dependence of slot tags, as shown below.

1) BLSTM-softmax: At each time step, a linear output layer is applied to predict slot tags independently, i.e.

$$p(o^S|x) = \prod_{i=1}^{|x|} p(o_i^S|\mathbf{h}_i^{\mathrm{sen}}) = \prod_{i=1}^{|x|} \mathrm{softmax}_{o_i^S}(\mathbf{W}_2 \mathbf{h}_i^{\mathrm{sen}}) \quad (9)$$

where $\mathbf{W}_2 \in \mathbb{R}^{|\mathcal{T}| \times 2n}$ is trainable, and $\mathrm{softmax}_{o_i^S}(.)$ is precisely the o_i^S-th element of the distribution defined by the *softmax* function.

2) BLSTM-CRF: CRF output layer considers the correlations between tags in neighborhoods and jointly decode the best chain of tags for a given input sentence [49], [50], [51]. The posterior probability of slot sequence is computed via:

$$\psi(x, o^S) = \sum_{i=1}^{|x|} ([\mathbf{A}]_{o_{i-1}^S, o_i^S} + [\mathbf{W}_2 \mathbf{h}_i^{\mathrm{sen}}]_{o_i^S}) \quad (10)$$

$$p(o^S|x) = \frac{\exp(\psi(x, o^S))}{\sum_{o^{S'}} \exp(\psi(x, o^{S'}))} \quad (11)$$

where $\mathbf{A} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$ is a transition matrix, and its element $[\mathbf{A}]_{m,n}$ models the transition from the m-th to the n-th label for a pair of consecutive time steps.

3) BLSTM-focus: To consider the time series dependence of slot tags, several encoder-decoder architectures [17], [18], [3], [34] are also proposed for slot filling. With the focus mechanism [18], we utilize a uni-directional LSTM based decoder to model tag dependencies. The decoder's hidden vector at the i-th time step is computed by $\mathbf{h}_i^{\mathrm{tag}} = \mathrm{f}_{\mathrm{LSTM}}(\mathbf{h}_i^{\mathrm{sen}} \oplus \mathbf{o}_{i-1}^S, \mathbf{h}_{i-1}^{\mathrm{tag}})$, where $\mathbf{o}_{i-1}^S$ is the embedding of the previously predicted slot tag, and $\mathbf{h}_0^{\mathrm{tag}} = \overleftarrow{\mathbf{h}}_1^{\mathrm{sen}}$. Then we compute $p(o^S|x)$ via:

$$p(o^S|x) = \prod_{i=1}^{|x|} p(o_i^S|o_{<i}^S, x) = \prod_{i=1}^{|x|} \mathrm{softmax}_{o_i^S}(\mathbf{W}_3 \mathbf{h}_i^{\mathrm{tag}})$$

where $\mathbf{W}_3 \in \mathbb{R}^{|\mathcal{T}| \times n}$. Compared with *BLSTM-CRF*, this method can model longer-range dependence of slot tags.

Slot-Value Summary: Though we can get a sequence of slot tags after slot tagging, it is a semifinished representation that the value for each predicted slot is not revealed. With alignment between the predicted tag sequence and input sentence, we can extract a summary of slot-value pairs easily. For instance, the list of slot-value pairs for the sample in Fig. 2 is *(FromCity=Shanghai, ToCity=New York)*. Let o^C denote the list of slot-value pairs, and then we can get the final semantic form y of the input x:

$$y = (o^I, o^C) = (o^I, \mathrm{getSummary}(o^S, x)) \quad (12)$$

The loss function of the NLU model given x and y is

$$\mathcal{L}_{\mathrm{NLU}}(x, y) = -\log p(\widetilde{y}|x) = -\log p(o^I|x) - \log p(o^S|x)$$

IV. Dual Task of NLU

In this section, we will introduce the dual task of NLU, which is formulated as a semantic-to-sentence generation (SSG) task. It generates the corresponding sentence x given an intent o^I and a list of slot-value pairs $o^C = (o_1^C, \cdots, o_M^C)$ (M is the number of slot-value pairs). As a fact of the IOB annotation schema, each value in o^C must appear in x without overlapping. Thus, we choose to first generate a delexicalized form[1] $\widetilde{x}$ comprised of words and slots, and then fill the slots up with given values in o^C to get x. We wish to estimate

$$p(\widetilde{x}|y) = p(\widetilde{x}|o^I, o^C) \quad (13)$$

, the conditional probability of delexicalized form $\widetilde{x}$ given intent o^I and slot-value pairs o^C.

[1]For example, the delexicalized form of the sentence in Fig. 2 is "*show me flights from* ⟨FromCity⟩ *to* ⟨ToCity⟩".

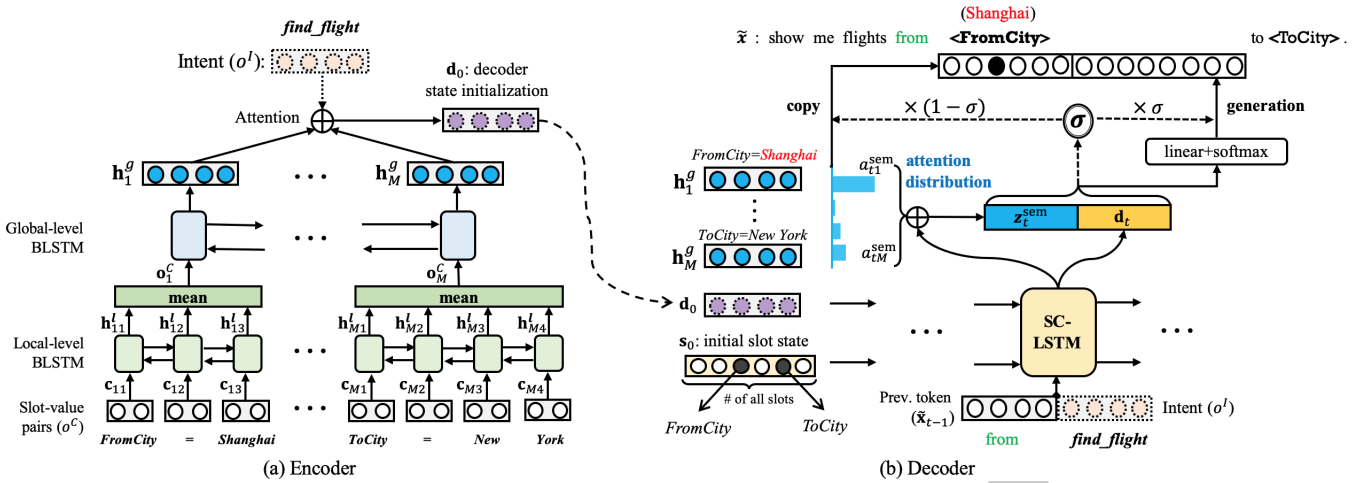

Fig. 3. The proposed architecture for the dual task of NLU, which is comprised of an encoder and a decoder. The encoder is a hierarchical BLSTM to obtain deep features for a list of slot-value pairs o^C. The decoder exploits a semantically controlled LSTM to precisely generate a delexicalized form $\widetilde{x}$, and then substitute the special slot tokens with the corresponding values in o^C.

Sequence-to-sequence based encoder-decoder architectures has achieved success in natural language generation such as machine translation [47], [48], dialogue generation [52] and text summarization [53]. However, it is non-trivial to apply the encoder-decoder architectures into SSG, since the input of SSG is not a sequence any more but a structured form (i.e., an intent and a list of slot-value pairs).

The proposed architecture for SSG is illustrated in Fig. 3. An *encoder* is exploited to encode the intent o^I and the list of slot-value pairs o^C into vector representations, and a *decoder* learns to generate the delexicalized form $\widetilde{x}$ depending on the encoding vectors. Finally, we replace the slots in $\widetilde{x}$ with the corresponding values in o^C.

Encoder: We exploit a hierarchical BLSTM to encode the list of slot-value pairs at local and global levels. Firstly, each slot-value pair is considered as a sub-sequence, i.e., $o^C_m = (c_{m1}, \cdots, c_{mT_m})$, where T_m is the sequence length, and $m \in \{1, \cdots, M\}$. For example, a slot-value pair, *ToCity=New York*, is tokenized as ("*ToCity*", "=", "*New*", "*York*").

1) Local-level: For each slot-value pair o^C_m, we use a shared BLSTM to get local representations independently:

$$(\mathbf{h}^l_{m1}, \cdots, \mathbf{h}^l_{mT_m}) \leftarrow \text{BLSTM}_{\Theta_3}(\mathbf{c}_{m1}, \cdots, \mathbf{c}_{mT_m}) \quad (14)$$

where $\mathbf{c}_{mj}$ is the embedding of j-th token[2] in o^C_m. The local representation of o^C_m is defined as $\mathbf{o}^C_m = \frac{1}{T_m}\sum_j \mathbf{h}^l_{mj}$, $\mathbf{o}^C_m \in \mathbb{R}^{2n}$.

2) Global-level: Upon the local representations of all slot-value pairs in o^C, another BLSTM is applied to get global hidden features, $\mathbf{h}^g_m \in \mathbb{R}^{2n}$, $m \in \{1, \cdots, M\}$:

$$(\mathbf{h}^g_1, \cdots, \mathbf{h}^g_M) \leftarrow \text{BLSTM}_{\Theta_4}(\mathbf{o}^C_1, \cdots, \mathbf{o}^C_M) \quad (15)$$

Decoder: In order to avoid generating redundant or missing slots in the prediction of sequence $\widetilde{x}$, the semantically controlled LSTM (SC-LSTM) [54] is applied. The hidden vector at the t-th time step is computed by $(\mathbf{d}_t, \mathbf{s}_t) = \text{f}_{\text{SC-LSTM}}(\widetilde{\mathbf{x}}_{t-1} \oplus \mathbf{o}^I, (\mathbf{d}_{t-1}, \mathbf{s}_{t-1}))$, where $\widetilde{\mathbf{x}}_{t-1}$ is the embedding of the previously predicted token, $\mathbf{o}^I$ is the embedding of the given intent, and $\mathbf{d}_t$ is the hidden vector. Compared with LSTM, the SC-LSTM contains a slot-value state $\mathbf{s}_t$ which plays the role of sentence planning. $\mathbf{s}_t$ manipulates the slot-value features during the generation process in order to produce a hidden vector which accurately encodes the input semantics. The slot-value state is initialized with the original slots 1-hot vector $\mathbf{s}_0$ where each element is zero except for the slots in o^C. Additional regularization term will be added to the final loss function for each sample ($\mathbf{s}_{|\widetilde{x}|}$ is the final slot state vector)

$$\mathcal{L}_{\text{SC}} = ||\mathbf{s}_{|\widetilde{x}|}||_2 + \sum_{t=1}^{|\widetilde{x}|} \eta \xi^{||\mathbf{s}_t - \mathbf{s}_{t-1}||_2}$$

where $\eta = 10^{-4}, \xi = 100$, $||\cdot||_2$ is l_2 norm. The first term is used to penalise generated sequences that failed to render all the required slots, while the second term discourages the decoder from turning more than one slot off in a single time step.

The hidden vector is initialized by the aggregated encoding vectors, i.e. $\mathbf{d}_0 = \mathbf{W}_0\mathbf{z}^{\text{sem}}_0$, where $\mathbf{W}_0 \in \mathbb{R}^{n \times 2n}$, and $\mathbf{z}^{\text{sem}}_0 \in \mathbb{R}^{2n}$ is an attention vector of the encoder hiddens, i.e.,

$$\mathbf{z}^{\text{sem}}_0, (a^{\text{sem}}_{01}, \cdots, a^{\text{sem}}_{0M}) \leftarrow \text{ATTN}_{\Theta_5}(\mathbf{o}^I, (\mathbf{h}^g_1, \cdots, \mathbf{h}^g_M)) \quad (16)$$

An output layer with the attention mechanism [48] and the copying mechanism [53] is applied on the SC-LSTM to predict tokens in $\widetilde{x}$. The attention weight for the current step t of the decoder with the m-th slot-value pair in the encoder ($m \in \{1, \cdots, M\}$) and the attention vector are computed via

$$\mathbf{z}^{\text{sem}}_t, (a^{\text{sem}}_{t1}, \cdots, a^{\text{sem}}_{tM}) \leftarrow \text{ATTN}_{\Theta_6}(\mathbf{d}_t, (\mathbf{h}^g_1, \cdots, \mathbf{h}^g_M)) \quad (17)$$

Then we compute the vocabulary distribution

$$p_{\text{gen}}(\widetilde{x}_t | \widetilde{x}_{<t}, y) = \text{softmax}_{\widetilde{x}_t}(\mathbf{W}_o(\mathbf{d}_t \oplus \mathbf{z}^{\text{sem}}_t)) \quad (18)$$

, where $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}_{\widetilde{x}}| \times 3n}$, and $|\mathcal{V}_{\widetilde{x}}|$ is the output vocabulary size. Generation terminates once an end-of-sequence token "EOS" is emitted.

[2]Every slot is also mapped to a trainable embedding vector.

Except for directly generation, the decoder also includes the copying mechanism to improve model generalization, which copies slots from the slot-value pairs in o^C. We use sigmoid gate function σ to make a soft decision between generation and copy at each step t, i.e. $g_t = \sigma(\mathbf{v}_g^\top(\mathbf{d}_t \oplus \mathbf{z}_t^{\text{sem}}))$ and

$$p(\widetilde{x}_t|\widetilde{x}_{<t}, y) = g_t p_{\text{gen}}(\widetilde{x}_t|\widetilde{x}_{<t}, y) + (1-g_t)p_{\text{copy}}(\widetilde{x}_t|\widetilde{x}_{<t}, y)$$

where $g_t \in [0,1]$ is the balance score, $\mathbf{v}_g$ is a weight vector. Distribution $p_{\text{copy}}(\cdot|\cdot)$ is defined over M slots in $(o_1^C, \cdots, o_M^C)$:

$$p_{\text{copy}}(\widetilde{x}_t|\widetilde{x}_{<t}, y) = \begin{cases} a_{tm}^{\text{sem}}, & \widetilde{x}_t \text{ is the slot of } o_m^C, m \in [1, M] \\ 0, & \text{otherwise} \end{cases}$$

Afterward, we can get the final sentence x by substituting each slot in $\widetilde{x}$ with the value in the corresponding slot-value pair. The loss function of SSG model given x and y is

$$\mathcal{L}_{\text{SSG}}(y, x) = -\sum_{t=1}^{|\widetilde{x}|} \log p(\widetilde{x}_t|\widetilde{x}_{<t}, y) + \mathcal{L}_{\text{SC}}$$

V. Dual Semi-supervised NLU

In this section, we will describe our dual semi-supervised framework for the NLU task, which contains two methods: dual pseudo-labeling and dual learning. Algorithm 1 gives an overview of the dual semi-supervised NLU. Besides the labeled data $\mathcal{D}_{\text{xy}}^L$, there are two kinds of unlabeled data here: pure (unlabeled) sentences set $\mathcal{D}_{\text{x}}^U$, and pure (unexpressed) semantic forms (i.e., intents and lists of slot-value pairs without corresponding sentences) set $\mathcal{D}_{\text{y}}^U$.

A. Dual Pseudo-Labeling Method

Pseudo-Label are target labels for unannotated data as if they were true labels [22]. We first pre-train the NLU and SSG models in a supervised fashion with labeled data. Given an unlabeled sentence x, we can use the NLU model to generate pseudo label y'. Symmetrically, we can also obtain x' by utilizing the SSG model given an unexpressed semantic form y. In other words, we can create pseudo training samples (x, y') and (x', y) in addition to the existing labeled data, as shown in Algorithm 1 (part 1).

Except for the supervised training stage with the labeled data, the NLU and SSG models can also be fine-tuned with the pseudo-samples by respectively minimizing losses $w_i(\mathcal{L}_{\text{NLU}}(x, y') + \mathcal{L}_{\text{NLU}}(x', y))$ and $w_i(\mathcal{L}_{\text{SSG}}(y, x') + \mathcal{L}_{\text{SSG}}(y', x))$ at the i-th iteration, where w_i is an important coefficient. To prevent models stuck in poor local minima, we slowly increase w_i such that greater confidence is assigned to pseudo-samples as training goes on. Concretely, $w_i = \frac{i}{N}$, where N is the maximum number of iterations.

Besides unlabeled x and unexpressed y, we also generate pseudo-samples starting from a sentence x and a semantic form y in the labeled data. Because it may rectify annotation noise and create various expressions for the same semantic form.

Previous work [22], [23], [55] related to the pseudo-labeling method only proposes to generate pseudo-labels y' for unlabeled inputs x. To the best of our knowledge, we are the first to propose the dual pseudo-labeling method, which shares pseudo-samples between the NLU and SSG tasks and optimizes the NLU and SSG models iteratively.

Algorithm 1 Dual Semi-supervised NLU.

Require: Labeled training set $\mathcal{D}_{\text{xy}}^L$; pure (unlabeled) sentences set $\mathcal{D}_{\text{x}}^U$; pure (unexpressed) semantic forms (i.e., intents and lists of slot-value pairs) set $\mathcal{D}_{\text{y}}^U$; beam search size K; weight factor δ; maximum number of iterations N.

Train sentence side language model LM$(\cdot)$ on data $\mathcal{D}_{\text{xy}}^L \cup \mathcal{D}_{\text{x}}^U$. Build lexicon database DB$(\cdot)$ and intent-slot co-occurence matrix COM on $\mathcal{D}_{\text{xy}}^L \cup \mathcal{D}_{\text{y}}^U$.
Pre-train NLU$(\cdot|\Theta_{\text{NLU}})$ and SSG$(\cdot|\Theta_{\text{SSG}})$ models on $\mathcal{D}_{\text{xy}}^L$ by respectively minimizing the cross-entropy losses $\sum_{(x,y)\in\mathcal{D}_{\text{xy}}^L} \mathcal{L}_{\text{NLU}}(x, y)$ and $\sum_{(x,y)\in\mathcal{D}_{\text{xy}}^L} \mathcal{L}_{\text{SSG}}(y, x)$.
for $i = 1$ to N **do**
  repeat
    Sample sentence $x \sim \mathcal{D}_{\text{xy}}^L \cup \mathcal{D}_{\text{x}}^U$
    Sample semantic form $y \sim \mathcal{D}_{\text{xy}}^L \cup \mathcal{D}_{\text{y}}^U$
    ▷ **Part 1: Dual Pseudo-Labeling Method**
    Use the current NLU model to generate pseudo labels for x, i.e., $y' = \text{NLU}(x|\Theta_{\text{NLU}})$.
    Use the current SSG model to generate pseudo sentences for y, i.e., $x' = \text{SSG}(y|\Theta_{\text{SSG}})$.
    Update Θ_{NLU} and Θ_{SSG} on the generated pseudo-samples by minimizing $w_i(\mathcal{L}_{\text{NLU}}(x, y') + \mathcal{L}_{\text{NLU}}(x', y))$ and $w_i(\mathcal{L}_{\text{SSG}}(y, x') + \mathcal{L}_{\text{SSG}}(y', x))$ respectively.
    ▷ **Part 2: Dual Learning Method**
    Produce K semantic forms $y'^1, \cdots, y'^K$ using beam search according to NLU$(x|\Theta_{\text{NLU}})$.
    For each y'^k, generate $x'^k = \text{SSG}(y'^k|\Theta_{\text{SSG}})$.
    For k-th sample, compute the total reward $r_1^k(x)$.
    Compute gradients $\nabla_{\Theta_{\text{NLU}}}\psi_1(x)$ and $\nabla_{\Theta_{\text{SSG}}}\psi_1(x)$.
    Produce K sentences $x'^1, \cdots, x'^K$ using beam search according to SSG$(y|\Theta_{\text{SSG}})$.
    For each x'^k, generate $y'^k = \text{NLU}(x'^k|\Theta_{\text{NLU}})$.
    For k-th sample, compute the total reward $r_2^k(y)$.
    Compute gradients $\nabla_{\Theta_{\text{NLU}}}\psi_2(y)$ and $\nabla_{\Theta_{\text{SSG}}}\psi_2(y)$.
    Update Θ_{NLU} with $\delta\nabla_{\Theta_{\text{NLU}}}\psi_1(x) + (1-\delta)\nabla_{\Theta_{\text{NLU}}}\psi_2(y)$; update Θ_{SSG} with $\delta\nabla_{\Theta_{\text{SSG}}}\psi_1(x) + (1-\delta)\nabla_{\Theta_{\text{SSG}}}\psi_2(y)$.
    ▷ **Part 3: Raw Supervised Training**
    Sample training pair $(x, y) \sim \mathcal{D}_{\text{xy}}^L$;
    Update Θ_{NLU} and Θ_{SSG} by minimizing $\mathcal{L}_{\text{NLU}}(x, y)$ and $\mathcal{L}_{\text{SSG}}(y, x)$ respectively.
  until all samples in $\mathcal{D}_{\text{xy}}^L, \mathcal{D}_{\text{x}}^U, \mathcal{D}_{\text{y}}^U$ are sampled.
end for

B. Dual Learning Method

Besides the pseudo-labeling method where the NLU and SSG models are updated separately, we also propose to apply the dual learning method by jointly training the two models. We use one agent to represent the model of the primal task (NLU) and another agent to represent the model of the dual task (SSG). Then a two-agent game is designed in a closed loop which can provide quality feedback to the primal and dual models even if only sentences or semantic forms are available. As the feedback rewards are non-differentiable, reinforcement learning algorithm [56] based on policy gradient [57] is applied for optimization.

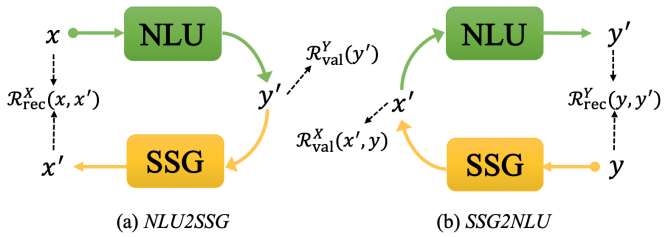

Fig. 4. An overview of dual learning method. The NLU and SSG models can form a closed cycle, which contains two directed loops *NLU2SSG* and *SSG2NLU* starting from a sentence x and semantic form y respectively.

As illustrated in Fig. 4, two agents, NLU and SSG, participate in the collaborative game with two directed loops. 1) *NLU2SSG* loop starts from a sentence, generates a possible semantic form by agent NLU and tries to reconstruct the original sentence by SSG. 2) *SSG2NLU* loop starts from the opposite side. Each agent will obtain quality feedback depending on reward functions defined in the directed loops. The NLU and SSG models are pre-trained on the labeled data. Let Θ_{NLU} and Θ_{SSG} denote all the parameters of the NLU and SSG models respectively. A brief description of the dual learning algorithm is provided in Algorithm 1 (part 2), comprised of the two directed loops:

1) Loop NLU2SSG: We sample a sentence x from the union of labeled and unlabeled data randomly. Given x, the NLU model could produce K possible semantic form $y'^1, \cdots, y'^K$ via beam search (K is beam size). For each y'^k, we can obtain a validity reward $\mathcal{R}^Y_{\text{val}}(y'^k)$ (a scalar) which reflects the likelihood of y'^k being a valid semantic form. Afterwards, we pass y'^k into the SSG model and get an output x'^k by greedy decoding. Finally, we get a reconstruction reward $\mathcal{R}^X_{\text{rec}}(x, x'^k)$ which forces the generated sentence x'^k as similar to x as possible. The rewards will be elucidated in Section V-B3. A coefficient $\alpha \in [0,1]$ is exploited to balance these two rewards in $r^k_1(x) = \alpha\mathcal{R}^Y_{\text{val}}(y'^k) + (1-\alpha)\mathcal{R}^X_{\text{rec}}(x, x'^k)$.

By minimizing the negative expected reward $\psi_1(x) = \mathbb{E}[-\frac{1}{K}\sum_{k=1}^K r^k_1(x)]$ via policy gradient [57], the stochastic gradients of Θ_{NLU} and Θ_{SSG} are computed as:

$$\nabla_{\Theta_{\text{NLU}}}\psi_1(x) = \frac{1}{K}\sum_{k=1}^K r^k_1(x)\nabla_{\Theta_{\text{NLU}}}\mathcal{L}_{\text{NLU}}(x, y'^k)$$

$$\nabla_{\Theta_{\text{SSG}}}\psi_1(x) = \frac{1-\alpha}{K}\sum_{k=1}^K \mathcal{R}^X_{\text{rec}}(x, x'^k)\nabla_{\Theta_{\text{SSG}}}\mathcal{L}_{\text{SSG}}(y'^k, x)$$

2) Loop SSG2NLU: Symmetrically, we sample a semantic form y from the labeled and unlabeled data randomly. Given y, the SSG model could generate K possible sentences $x'^1, \cdots, x'^K$ via beam search. For each x'^k, we can obtain a validity reward $\mathcal{R}^X_{\text{val}}(x'^k, y)$ which reflects whether the sampled natural language sentence x'^k is well-formed and fluent. Afterwards, we feed x'^k into the NLU model, and get the top-hypothesis y'^k. Finally, we get a reconstruction reward $\mathcal{R}^Y_{\text{rec}}(y, y'^k)$ which forces y'^k as similar to y as possible. The rewards will be explained in Section V-B3. A coefficient $\beta \in [0,1]$ is exploited to balance these two rewards in $r^k_2(y) = \beta\mathcal{R}^X_{\text{val}}(x'^k, y) + (1-\beta)\mathcal{R}^Y_{\text{rec}}(y, y'^k)$.

By minimizing the negative expected reward $\psi_2(y) = \mathbb{E}[-\frac{1}{K}\sum_{k=1}^K r^k_2(y)]$ via policy gradient [57], the stochastic gradients of Θ_{SSG} and Θ_{NLU} are computed as:

$$\nabla_{\Theta_{\text{SSG}}}\psi_2(y) = \frac{1}{K}\sum_{k=1}^K r^k_2(y)\nabla_{\Theta_{\text{SSG}}}\mathcal{L}_{\text{SSG}}(y, x'^k)$$

$$\nabla_{\Theta_{\text{NLU}}}\psi_2(y) = \frac{1-\beta}{K}\sum_{k=1}^K \mathcal{R}^Y_{\text{rec}}(y, y'^k)\nabla_{\Theta_{\text{NLU}}}\mathcal{L}_{\text{NLU}}(x'^k, y)$$

To the best of our knowledge, we are the first to apply the dual learning algorithm to the intent detection and slot filling in NLU. Compared with the dual learning for neural machine translation [25] with only language models based reward, we introduce new validity and reconstruction rewards for structured data of NLU.

3) Reward Design: Here we will give some details about two validity and two reconstruction rewards introduced in the two directed loops above.

Validity reward of $\mathcal{R}^Y_{\text{val}}(y')$ measures whether a possible semantic form y' is valid. It can be jointly evaluated on intents and slots from two relations:

- *Slot-value*: whether a given slot-value pair is valid, e.g. "boston" is a valid city name for slot `FromCity`.
- *Slot-intent*: whether a slot is likely to co-occur with respect to the predicted intent, e.g. `FromCity` often co-occurs with intent `find_flight`.

To this end, a lexicon **d**ata**b**ase DB$(\cdot)$ is created from the training set, which specifies any possible value v for each slot s. A **co**-**o**ccurrence **m**atrix (COM) is leveraged from the training set, where COM$(i, s) \in \{0, 1\}$ indicates whether the slot s co-occurs with the user intent i. Concretely, $\mathcal{R}^Y_{\text{val}}(y')$ is defined as

$$\text{score}(s, v) = \max_{e \in \text{DB}(s)} (1 - \text{Edit\_Distance}(e, v)/|v|)$$

$$r_{\text{sv}}(o^{C\prime}) = \begin{cases} \frac{1}{|o^{C\prime}|}\sum_{(s,v)\in o^{C\prime}} \text{score}(s, v), & \text{if } |o^{C\prime}| \neq 0 \\ 1.0, & \text{otherwise} \end{cases}$$

$$r_{\text{si}}(o^{I\prime}, o^{C\prime}) = \begin{cases} \frac{1}{|o^{C\prime}|}\sum_{(s,v)\in o^{C\prime}} \text{COM}(o^{I\prime}, s), & \text{if } |o^{C\prime}| \neq 0 \\ 1.0, & \text{otherwise} \end{cases}$$

$$\mathcal{R}^Y_{\text{val}}(y') = \lambda \cdot r_{\text{sv}}(o^{C\prime}) + (1-\lambda) \cdot r_{\text{si}}(o^{I\prime}, o^{C\prime})$$

where $y' = (o^{I\prime}, o^{C\prime})$ contains an intent $o^{I\prime}$ and a list of slot-value pairs $o^{C\prime}$, Edit_Distance(e, v) calculates a word-level edit distance between two values, and λ is a weight factor.

Validity reward of $\mathcal{R}^X_{\text{val}}(x', y)$ measures whether a generated natural language sentence x' is well-formed and fluent. We also evaluate it from two aspects:

- *Semantic integrity*: whether x' expresses all slots in the input y precisely. This can be measured by a metric of slot accuracy, i.e. SlotAcc$(x', y) = 1 - \frac{p+q}{m}$, where m is the total number of slot-value pairs in y, p and q are the number of omitted and redundant slots in the delexicalized form of x' respectively.
- *Word fluency*: the probability of x' to be a natural language sentence. We train a LSTM based language model [58] with sentences of both the labeled and

TABLE I
DATASET STATISTICS.

Dataset	Vocab Size	Train	Valid	Test	#Slot	#Intent
ATIS	950	4478	500	893	83	18
SNIPS	14349	13084	700	700	39	7

unlabeled data to evaluate the quality of x'. Length-normalization [59] is applied to make a fair competition between short and long sentences, i.e. $\frac{1}{|x'|}\log \text{LM}(x')$.

A weight factor γ is used to combine these two aspects:

$$\mathcal{R}^X_{\text{val}}(x', y) = \gamma \cdot \text{SlotAcc}(x', y) + (1-\gamma)\cdot \frac{1}{|x'|}\log \text{LM}(x')$$

Reconstruction reward of $\mathcal{R}^X_{\text{rec}}(x, x')$ measures the similarity score between the finally generated sentence x' and the raw input x. The BLEU score [60] is utilized:

$$\mathcal{R}^X_{\text{rec}}(x, x') = \text{BLEU}(x, x')$$

Reconstruction reward of $\mathcal{R}^Y_{\text{rec}}(y, y')$ reflects the similarity between the finally produced semantic form $y' = (o^{I\prime}, o^{C\prime})$ and the raw input $y = (o^I, o^C)$. We use the slot-value F_1 score and intent accuracy to measure it, i.e.

$$\mathcal{R}^Y_{\text{rec}}(y, y') = \omega \mathbb{I}\{o^I = o^{I\prime}\} + (1-\omega)\text{F}_1(o^C, o^{C\prime})$$

where $\mathbb{I}$ is the indicator function and η is a weight factor.

VI. EXPERIMENTS

In this section, we first introduce the datasets and baselines with details of the experimental setup. Then, we compare the performance of our proposed methods with the baselines. Finally, extensive ablation studies are conducted for analysis.

A. Datasets

We evaluate our proposed methods on two public datasets: Airline Travel Information Systems (ATIS) dataset [26] and SNIPS Natural Language Understanding benchmark (SNIPS) [27]. ATIS is a widely used dataset in spoken language understanding, where audio recordings of people making flight reservations are collected. SNIPS contains natural language corpus collected in a crowdsourced fashion to benchmark the performance of voice assistants. The statistical information on the two datasets are illustrated in Table I.

B. Baselines

We compare the proposed dual semi-supervised NLU with other alternatives:

- *Supervised NLU* only exploits labeled data ($\mathcal{D}^L_{\text{xy}}$) for supervised learning, e.g. *BLSTM-softmax*, *BLSTM-CRF* and *BLSTM-focus* methods described in Section III.
- *Multi-task learning with unsupervised task* can exploit RNN-based language modelling [40], [41], [24], [37] and sequence-to-sequence based sentence auto-encoder [42], [43] to additionally utilize unlabeled sentences ($\mathcal{D}^U_{\text{x}}$). We implement the *sentence auto-encoder* in our experiments.
- The traditional *pseudo-labeling (PL) method* without the dual task [21], [22], [23] creates pseudo-samples for unlabeled sentences ($\mathcal{D}^U_{\text{x}}$) to perform data augmentation, using a pre-trained NLU model.
- *Template synthesis* method first extracts templates by converting each input sentence of the labeled data into its delexicalized form (e.g. "*show me flights from* ⟨FromCity⟩ *to* ⟨ToCity⟩"). Afterwards, we synthesize additional labeled samples for the supervised training by replacing slot types in each template with the corresponding values provided in unexpressed semantic forms ($\mathcal{D}^U_{\text{y}}$).

C. Experimental Setup

1) Training Details: The word embeddings with 400 dimensions are initialized by concatenating pre-trained Glove embeddings[3] [61] and character embeddings [62], which can be updated during training. The hidden size n is 256. Hyper-parameters $\alpha, \beta, \gamma, \omega, \delta$ are set to 0.5, and λ is 0.25 empirically. For the dual learning, the beam size K is set to 5. The network parameters are randomly initialized under the uniform distribution $[-0.2, 0.2]$, except for the pre-trained word embeddings. We use optimizer Adam [63] with learning rate 0.001 for all experiments. The *dropout* with a probability of 0.5 is applied to the non-recurrent connections during the training stage. The batch size is 16 for all datasets. The maximum norm for gradient clipping is set to 5, and we use l_2 norm regularization on all weights with factor $1e$-5 to avoid over-fitting. We keep the learning rate for 50 epochs and save the parameters that give the best performance on the validation set. Finally, we report the intent accuracy and F_1-score of slot-value pairs on the test set with parameters that have achieved the best average of intent accuracy and slot F_1-score on the validation set. The F_1-score is calculated using CoNLL evaluation script[4].

Besides using pre-trained word embeddings, some advanced pre-trained language models (e.g., ELMo [64], BERT [65]) can also be used to get input embeddings. It is investigated in the following ablation studies. We employ the pre-trained BERT model (*bert-base-cased*) with 12 layers of 768 hidden units and 12 self-attention heads[5]. We update all the parameters using the Adam with a learning rate $5e$-5.

2) Data settings for semi-supervised learning: To evaluate the effectiveness and efficiency of different methods for semi-supervised NLU, we discuss the experimental configuration for semi-supervised settings below. In order to simulate the annotation scarcity problem in the real world, a part of the training set is kept as fully labeled data ($\mathcal{D}^L_{\text{xy}}$), and the rest is left as unpaired sentences and semantic forms ($\mathcal{D}^U_{\text{x}}$ and $\mathcal{D}^U_{\text{y}}$ respectively) to simulate unlabeled data. For the part of labeled data, we randomly select 5, 10, 15, 20, 30 and 50 percent of the training set in each dataset for experiments.

3) Significance Test: We use McNemar's test to establish the statistical significance of a method over another ($\text{p} < 0.05$).

D. Overall Results

We first compare different methods on ATIS and SNIPS datasets with the simulated semi-supervised settings as well as

[3]http://nlp.stanford.edu/data/glove.840B.300d.zip
[4]https://www.clips.uantwerpen.be/conll2000/chunking/output.html
[5]https://github.com/google-research/bert#pre-trained-models

TABLE II
SLOT F_1 SCORES AND INTENT ACCURACIES OF DIFFERENT METHODS ON ATIS AND SNIPS DATASETS. WE RANDOMLY SELECT $\{5,10,20,30,50\}\%$ OF THE TRAINING SET AS LABELED DATA, AND LEAVE THE REST TO SIMULATE UNLABELED DATA. THE RESULTS IN BOLD BLACK ARE THE BEST SLOT F_1 SCORES AND INTENT ACCURACIES. ‡ INDICATES OUR RESULTS THAT SIGNIFICANTLY OUTPERFORM THE BEST BASELINE.

Method \ Slot F_1 (%)		ATIS						SNIPS					
		5%	10%	20%	30%	50%	100%	5%	10%	20%	30%	50%	100%
supervised	**BLSTM-focus** (backbone)	82.92	89.48	92.66	93.54	95.45	95.79	87.89	91.23	93.54	94.45	94.92	**96.44**
semi-supervised	+ sentence auto-encoder	83.16	89.65	92.74	94.52	95.36	95.87	87.83	90.29	93.20	94.68	94.78	95.89
	+ pseudo-labeling (PL)	84.75	90.08	94.07	94.91	95.52	95.75	90.67	91.89	93.89	94.29	95.06	96.00
	+ template synthesis	86.10	90.62	94.35	94.94	95.27	-	90.40	92.94	93.94	94.22	94.37	-
+ dual task (ours)	+ dual PL	**89.58**‡	93.49‡	94.88‡	**95.90**‡	96.02‡	95.82	93.86‡	94.46‡	**95.53**‡	95.23‡	95.29	96.22
	+ dual learning	88.92‡	93.40‡	95.09‡	95.50‡	95.70	96.00	93.85‡	94.18‡	95.31‡	95.08‡	**95.45**‡	95.86
	+ dual PL + dual learning	**89.58**‡	**93.53**‡	**95.37**‡	95.85‡	**96.14**‡	**96.37**‡	**94.00**‡	**94.51**‡	95.22‡	**95.34**‡	95.25	96.11

Method \ Intent Acc (%)		ATIS						SNIPS					
		5%	10%	20%	30%	50%	100%	5%	10%	20%	30%	50%	100%
supervised	**BLSTM-focus** (backbone)	89.03	92.61	94.40	94.85	**98.54**	98.43	97.86	98.14	98.00	98.29	98.71	**99.14**
semi-supervised	+ sentence auto-encoder	88.80	92.50	95.18	94.62	98.32	98.32	97.57	97.86	97.86	98.00	98.71	98.86
	+ pseudo-labeling (PL)	89.47	92.50	95.18	94.85	98.32	98.32	97.57	98.00	98.14	98.00	99.00	**99.14**
	+ template synthesis	90.05	92.05	94.06	94.51	98.10	-	97.86	98.00	**98.57**	98.14	98.86	-
+ dual task (ours)	+ dual PL	90.37	93.28	94.62	**96.08**‡	98.43	**98.66**	**98.57**‡	98.14	98.43	98.43	98.71	**99.14**
	+ dual learning	89.81	93.28	**95.30**	95.86‡	**98.54**	98.54	98.29	98.14	98.29	**98.57**	98.57	98.86
	+ dual PL + dual learning	**90.48**	**93.51**‡	95.18	95.30	**98.54**	98.54	98.29	**98.43**	**98.57**	98.14	**99.14**	98.86

full training samples, using the pre-trained word embeddings. Table II shows slot F_1 scores and intent accuracies of baselines and our methods on ATIS and SNIPS, then we can find that:

1) Intent detection is a much easier sub-task than slot filling. We can see that the performance gap of intent accuracy between using 5% and 100% labeled data is lower than that of slot F_1 score, especially on the SNIPS dataset. Meanwhile, there is little difference among various methods with respect to intent accuracy, whereas our methods can achieve the best in most cases.
2) For *supervised* NLU, we choose BLSTM-focus as our backbone model of NLU, rather than BLSTM-softmax and BLSTM-CRF. As shown in Table III, the BLSTM-focus model can achieve the best performance on ATIS and SNIPS with full training data.
3) Three baselines of *semi-supervised* learning NLU can improve performances by exploiting unlabeled data in most cases, where "+ *sentence auto-encoder*" adds a sequence-to-sequence based sentence reconstruction task, and "+ *pseudo-labeling (PL)*" uses the existing NLU model to generate pseudo-labels for unannotated sentences. "+ *template synthesis*" exploits unexpressed semantic forms and the labeled data to synthesize more labeled samples for training.
4) Compared with the traditional pseudo-labeling method (+ *PL*) without the dual task, our proposed dual pseudo-labeling method (+ *dual PL*) can get improvements by taking advantage of unexpressed semantic forms.
5) The proposed dual learning-based method can also make improvements over the baselines. Different from the dual pseudo-labeling method, it involves validity reward and reconstruction reward to estimate (soft) importances of generated sentences or semantic forms.
6) Finally, we combine the two proposed methods (as shown in Algorithm 1) and obtain further improvements. In most cases, the combination (+ *dual PL* + *dual learning*) can obtain the best performances especially on slot F_1 scores.
7) Our methods can even get improvements with 100% labeled data (i.e. no unlabeled data), e.g., we get 96.37% slot F_1 score on ATIS. However, our methods do not outperform the purely supervised method with 100% labeled data (slot F_1 is 96.44%) on SNIPS. The reason may be that the test set of ATIS contains more unseen delexicalized forms, while the test set of SNIPS includes more unseen slot-value pairs, as shown in Table IV. Our methods applied to fully labeled data are likely to generate varied natural language expressions (sentences) for existing semantic forms. Therefore, our methods fail to get improvements on SNIPS due to lots of unseen slot-value pairs.

TABLE III
COMPARISON AMONG BLSTM-SOFTMAX, BLSTM-CRF AND BLSTM-FOCUS FOR SUPERVISED NLU ON ATIS AND SNIPS DATASETS.

Method	ATIS		SNIPS	
	Slot F_1	Intent Acc	Slot F_1	Intent Acc
BLSTM-softmax	95.50	98.21	94.96	98.86
BLSTM-crf	95.62	98.32	96.34	98.86
BLSTM-focus	**95.79**	**98.43**	**96.44**	**99.14**

TABLE IV
DATA ANALYSIS OF TEST SETS COMPARED WITH TRAINING SETS.

Dataset	#Unseen delexicalized form	#Unseen slot-value pairs
ATIS	680	169
SNIPS	421	522

E. Analysis

In Section VI-D, significant improvements of two metrics have been witnessed on the two datasets. However, we would like to figure out the potential factors for the improvement. In this sub-section, we will show ablation studies on the SSG model, the dual pseudo-labeling and dual learning methods to reveal the effects of different components. Finally, we analyze the effect of BERT in our framework.

TABLE V
EVALUATIONS OF THE SSG MODEL FOR THE DUAL TASK OF NLU, WHICH IS SUPERVISED BY FULL TRAINING SETS ON ATIS AND SNIPS RESPECTIVELY.

Model	ATIS		SNIPS	
	BLEU	Slot Acc	BLEU	Slot Acc
supervised SSG	47.17	97.72	39.18	100.00
(-) w/o feeding intent	44.86	98.26	38.15	99.70
(-) w/o global BLSTM	41.14	96.15	31.65	98.95
(-) w/o copy mechanism	44.08	96.58	38.67	99.98
(-) w/o SC-LSTM	46.78	97.25	37.99	100.00

TABLE VI
EVALUATIONS OF THE SSG MODEL IN THE PROPOSED DUAL SEMI-SUPERVISED NLU.

Method	ATIS (10%)		SNIPS (5%)	
	BLEU	Slot Acc	BLEU	Slot Acc
supervised SSG	39.53	87.49	29.94	94.85
+ dual PL	40.28	93.90	35.26	98.85
+ dual learning	38.84	91.30	32.19	98.10
+ dual PL + dual learning	41.85	95.10	36.61	99.43

1) Ablation studies of the SSG model: To verify the effectiveness of the SSG model for the dual task of NLU, we apply ablation studies of supervised SSG on ATIS and SNIPS datasets, as shown in Table V. BLEU score [60] is exploited to measure the similarity between generated sentences and references in the test set. We also utilize the slot accuracy mentioned in Section V-B3 to measure the semantic integrity of generated sentences. From the result of "*(-) w/o feeding intent*" row, we can observe that the intent is essential for BLEU scores, whereas intent detection is much easy in NLU. The *global-level BLSTM* in the encoder, *copy mechanism* and *SC-LSTM cell* in the decoder are also important components of the SSG model.

Besides the supervised training, we may further want to know whether the SSG model will be improved in our proposed dual semi-supervised NLU. As shown in Table VI, the *dual pseudo-labeling* and *dual learning* methods can also improve the performance of the SSG model as well as the NLU model, where 10% and 5% of the training sets are selected as labeled data in ATIS and SNIPS respectively.

2) Ablation studies of the dual pseudo-labeling method: The dual pseudo-labeling method creates pseudo-samples in two ways: a) obtaining predicted semantic forms of sentences with the NLU model and b) generating sentences with the SSG model given intents and slot-value pairs. Experiments are conducted to make a comparison of these two ways, as shown in Table VII. From the results, we can find that *pseudo-samples created from SSG* are more essential. The reason may be that the SSG model tends to generate sentences semantically consistent with the given semantic forms though the sentences are not natural enough. However, the NLU model may predict wrong semantic labels. Meanwhile, without *pseudo-samples from SSG model*, the dual pseudo-labeling method reduces to the traditional pseudo-labeling without dual task.

From the result of "*(+)* w_i=1" row in Table VII, we can see that the increasing coefficient (w_i) at each iteration helps. We believe that the NLU and SSG models updated with more iterations could provide more qualified pseudo-samples.

TABLE VII
ABLATION STUDIES OF THE DUAL PSEUDO-LABELING METHOD.

Method	SNIPS (5%)	
	Slot F_1	Intent Acc
+ dual PL	93.86	98.57
(-) w/o pseudo-samples from NLU model	93.51	98.00
(-) w/o pseudo-samples from SSG model	90.67	97.57
(+) $w_i = 1$	93.65	98.14
(-) w/o iterative generation	91.49	97.71

TABLE VIII
ABLATION STUDIES OF THE DUAL LEARNING METHOD.

Method	SNIPS (5%)	
	Slot F_1	Intent Acc
+ dual learning	93.85	98.29
(-) w/o unlabeled sentences	92.96	98.14
(-) w/o unexpressed semantic forms	91.41	97.71
(-) w/o validity rewards	91.55	97.71
(-) w/o reconstruction rewards	93.74	97.86

Therefore, the performance decreases if we keep using pseudo-samples generated at the first iteration, as shown in "*(-) w/o iterative generation*" row.

3) Ablation studies of the dual learning method: Several experiments are conducted to show the effects of different components in the dual learning method, as illustrated in Table VIII. From the results of "*(-) w/o unlabeled sentences*" and "*(-) w/o unexpressed semantic forms*" rows, we can find that unexpressed semantic forms are more important, which is consistent with the findings in the ablation studies of the dual pseudo-labeling method. It may facilitate semi-supervised NLU, since semantic forms are well-structured and could be easily synthesized under domain knowledge. The last two rows show that two kinds of rewards are essential, while the validity rewards impact more on the slot F_1 score.

4) Effect of BERT: Besides using pre-trained word embeddings, the BERT model can also be used to get input embeddings[6]. However, it is orthogonal to the investigation of semi-supervised NLU. Table IX shows results on ATIS and SNIPS, where 10% and 5% of the training sets are selected as labeled data in respective. The results show that a pre-trained BERT model can further enhance our dual semi-supervised NLU as well as the baseline. Although BERT embeddings can bridge the gap between our method and the baseline, the dual semi-supervised NLU still outperforms the baseline significantly.

F. Compared with the Previous Results of the Supervised NLU

Finally, we make a comparison with the previous results on ATIS and SNIPS datasets using full training sets, as illustrated in Table X. Our proposed method (+ *dual PL* + *dual learning*) can achieve the state-of-the-art performances on the two datasets, but not significantly outperforming the previous state-of-the-art. We can also find that BERT boosts the performance of ATIS less than SNIPS, which may occur due to the much smaller vocabulary of ATIS. Our method

[6]We only consider BERT embeddings of the first subword if a word is broken into multiple subwords.

TABLE IX
SLOT F_1 SCORES AND INTENT ACCURACIES OF BERT-BASED MODELS ON THE TWO DATASETS.

Method	with BERT	**ATIS** (10%)		**SNIPS** (5%)	
		Slot	Intent	Slot	Intent
BLSTM-focus	✗	89.48	92.61	87.89	97.86
+ dual PL + dual learning	✗	93.53	93.51	94.00	98.29
BLSTM-focus	✓	91.41	93.51	91.53	98.14
+ dual PL + dual learning	✓	94.14	94.29	95.58	98.43

TABLE X
COMPARISON WITH PREVIOUS RESULTS OF NLU ON ATIS AND SNIPS.

	Method	**ATIS**		**SNIPS**	
		Slot	Intent	Slot	Intent
w/o BERT	Joint Seq. [32]*	94.3	92.6	87.3	96.9
	Attention BiRNN [3]*	94.2	91.1	87.8	96.7
	Slot-Gated [5]	95.2	94.1	88.8	97.0
	Self-Attentive Model [19]*	95.1	96.8	90.0	97.5
	Bi-Model [66]*	95.5	96.4	93.5	97.2
	CAPSULE-NLU [35]	95.2	95.0	91.8	97.3
	ELMo-Light for SLU [37]	95.4	97.3	93.3	98.8
	SF-ID Network [39]	95.8	97.1	92.2	97.3
	Stack-Propagation [7]	95.9	96.9	94.2	98.0
	our method	**96.4**	98.5	96.1	98.9
w/ BERT	Multi-ling. Joint BERT [38]	95.7	97.8	96.2	99.0
	Joint BERT SLU [6]	96.1	97.5	97.0	98.6
	Stack-Prop. + BERT [7]	96.1	97.5	97.0	99.0
	our method + BERT	96.0	**99.1**	**97.1**	**99.1**

* indicates a result borrowed from Qin et al. [7].

enhanced with BERT gets a decrease of slot F_1 score (from 96.4% to 96.0%) and an increase of intent accuracy (from 98.5% to 99.1%) on ATIS, while it achieves a better average score. It shows that our proposed method can also work well in fully supervised settings.

VII. CONCLUSION

This paper has introduced a dual task for SLU, which is semantic-to-sentence generation (SSG). It is incorporated in our proposed dual semi-supervised NLU to utilize unexpressed semantic forms as well as unlabeled sentences. The dual semi-supervised NLU includes the dual pseudo-labeling and dual learning methods which can learn the NLU and SSG models iteratively in the closed-loop of the primal and dual tasks. The proposed approaches are evaluated on two public datasets (ATIS and SNIPS). From the experimental results, we find that the dual semi-supervised NLU involving SSG could significantly improve the performances over traditional semi-supervised methods. We also provide extensive ablation studies to verify the effectiveness of our methods. Meanwhile, our methods can also achieve the state-of-the-art performance on the two datasets in the supervised setting.

The proposed framework of dual semi-supervised NLU shows promising perspectives of future improvements.

- Exploiting semantic forms for semi-supervised learning could be more affordable and effective than collecting in-domain sentences, since semantic forms are well-structured and could be automatically synthesized under domain knowledge.
- Validity and reconstruction rewards are important for softly validating pseudo-samples. We will explore appropriate rewards to improve the effectiveness and efficiency of the dual semi-supervised NLU in our future work.
- This work has shown the effectiveness of incorporating dual task in semi-supervised NLU. For developing a conversational dialogue system with wide application domains, the domain adaptation and transfer problems of the dual task will be an interesting future research direction.

REFERENCES

[1] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding–an introduction to the statistical framework," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[2] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *Proc. IEEE ASRU*, 2013, pp. 78–83.

[3] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proc. INTERSPEECH*, 2016, pp. 685–689.

[4] X. Zhang and H. Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *Proc. IJCAI*, 2016, pp. 2993–2999.

[5] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. NAACL*, 2018, pp. 753–757.

[6] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.

[7] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proc. EMNLP-IJCNLP*, 2019, pp. 2078–2087.

[8] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 50–58, 2008.

[9] G. Tur and L. Deng, "Intent determination and spoken utterance classification," *Spoken language understanding: systems for extracting semantic information from speech. Wiley, Chichester*, pp. 93–118, 2011.

[10] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Trans. on Audio, Speech and Lang. Process.*, vol. 22, no. 4, pp. 778–784, 2014.

[11] Y. Wang, L. Deng, and A. Acero, "Semantic frame-based spoken language understanding," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 41–91, 2011.

[12] Y. He and S. Young, "A data-driven spoken language understanding system," in *Proc. IEEE ASRU*, 2003, pp. 583–588.

[13] C. Raymond and G. Riccardi, "Generative and discriminative algorithms for spoken language understanding," in *Proc. INTERSPEECH*, 2007, pp. 1605–1608.

[14] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu *et al.*, "Using recurrent neural networks for slot filling in spoken language understanding," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 3, pp. 530–539, 2015.

[15] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *IEEE SLT Workshop*, 2014, pp. 189–194.

[16] N. T. Vu, "Sequential convolutional neural networks for slot filling in spoken language understanding," in *Proc. INTERSPEECH*, 2016, pp. 3250–3254.

[17] G. Kurata, B. Xiang, B. Zhou, and M. Yu, "Leveraging sentence-level information with encoder lstm for semantic slot filling," in *Proc. EMNLP*, 2016, pp. 2077–2083.

[18] S. Zhu and K. Yu, "Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding," in *Proc. ICASSP*, 2017, pp. 5675–5679.

[19] C. Li, L. Li, and J. Qi, "A self-attentive model with gate mechanism for spoken language understanding," in *Proc. EMNLP*, 2018, pp. 3824–3833.

[20] L. S. Zettlemoyer and M. Collins, "Online learning of relaxed ccg grammars for parsing to logical form," in *Proc. EMNLP-CoNLL*, 2007, pp. 678–687.

[21] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[22] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, 2013.

[23] A. Celikyilmaz, R. Sarikaya, D. Hakkani-Tür, X. Liu, N. Ramesh, and G. Tür, "A new pre-training method for training deep learning models with application to spoken language understanding." in *Proc. INTERSPEECH*, 2016, pp. 3255–3259.

[24] O. Lan, S. Zhu, and K. Yu, "Semi-supervised training using adversarial multi-task learning for spoken language understanding," in *Proc. ICASSP*, 2018, pp. 6049–6053.

[25] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *Proc. NeurIPS*, 2016, pp. 820–828.

[26] C. T. Hemphill, J. J. Godfrey, G. R. Doddington *et al.*, "The atis spoken language systems pilot corpus," in *Proc. the DARPA speech and natural language workshop*, 1990, pp. 96–101.

[27] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[28] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding." in *INTERSPEECH*, 2013, pp. 3771–3775.

[29] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding." in *INTERSPEECH*, 2013, pp. 2524–2528.

[30] N. T. Vu, P. Gupta, H. Adel, and H. Schütze, "Bi-directional recurrent neural network with ranking loss for spoken language understanding," in *Proc. ICASSP*, 2016, pp. 6060–6064.

[31] Y.-N. Chen, D. Z. Hakkani-Tür, and X. He, "Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models," in *Proc. ICASSP*, 2016, pp. 6045–6049.

[32] D. Hakkani-Tür, G. Tur, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional rnn-lstm," in *Proc. INTERSPEECH*, 2016, pp. 715–719.

[33] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *arXiv preprint arXiv:1707.06799*, 2017.

[34] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking." in *Proc. AAAI*, 2017, pp. 3365–3371.

[35] C. Zhang, Y. Li, N. Du, W. Fan, and S. Y. Philip, "Joint slot filling and intent detection via capsule neural networks," in *Proc. ACL*, 2019, pp. 5259–5267.

[36] L. Zhang and H. Wang, "Using bidirectional transformer-crf for spoken language understanding," in *Proc. NLPCC*, 2019, pp. 130–141.

[37] A. Siddhant, A. Goyal, and A. Metallinou, "Unsupervised transfer learning for spoken language understanding in intelligent agents," in *Proc. AAAI*, 2019.

[38] G. Castellucci, V. Bellomaria, A. Favalli, and R. Romagnoli, "Multi-lingual intent detection and slot filling in a joint bert-based model," *arXiv preprint arXiv:1907.02884*, 2019.

[39] E. Haihong, P. Niu, Z. Chen, and M. Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proc. ACL*, 2019, pp. 5467–5471.

[40] M. Rei, "Semi-supervised multitask learning for sequence labeling," in *Proc. ACL*, 2017, pp. 2121–2130.

[41] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," in *Proc. ACL*, 2017, pp. 1756–1765.

[42] Y.-B. Kim, K. Stratos, and D. Kim, "Adversarial adaptation of synthetic or stale data," in *Proc. ACL*, 2017, pp. 1297–1307.

[43] S. Zhu, O. Lan, and K. Yu, "Robust spoken language understanding with unsupervised asr-error adaptation," in *Proc. ICASSP*, 2018, pp. 6179–6183.

[44] R. Cao, S. Zhu, C. Liu, J. Li, and K. Yu, "Semantic parsing with dual learning," in *Proc. ACL*, 2019, pp. 51–64.

[45] S.-Y. Su, C.-W. Huang, and Y.-N. Chen, "Dual supervised learning for natural language understanding and generation," in *Proc. ACL*, 2019, pp. 5472–5477.

[46] C. Xia, C. Zhang, X. Yan, Y. Chang, and S. Y. Philip, "Zero-shot user intent detection via capsule neural networks," in *Proc. EMNLP*, 2018, pp. 3090–3099.

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

[48] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[49] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proc. ICASSP*, 2014, pp. 4077–4081.

[50] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[51] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proc. ACL*, 2016, pp. 1064–1074.

[52] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[53] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. ACL*, 2017, pp. 1073–1083.

[54] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned LSTM-based natural language generation for spoken dialogue systems," in *Proc. EMNLP*, 2015, pp. 1711–1721.

[55] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *arXiv preprint arXiv:1911.04252*, 2019.

[56] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[57] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. NeurIPS*, 2000, pp. 1057–1063.

[58] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, 2010.

[59] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[60] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002, pp. 311–318.

[61] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[62] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple NLP tasks," in *Proc. EMNLP*, 2017, pp. 1923–1933.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL*, 2018, pp. 2227–2237.

[65] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.

[66] Y. Wang, Y. Shen, and H. Jin, "A bi-model based rnn semantic frame parsing model for intent detection and slot filling," in *Proc. NAACL*, 2018, pp. 309–314.