ABSTRACT

Spoken language understanding (SLU) usually requires human semantic annotation on collected data, but the process is expensive. In order to make better use of unlabeled data for robust SLU, we propose an adversarial multi-task learning method by merging a bidirectional language model (BLM) and a slot tagging model (STM). As a secondary objective, the BLM is used to learn generalized and unsupervised knowledge with abundant unlabeled data and improve the performance of STM on unseen data samples. We construct a shared space for both tasks and independent private spaces for each task respectively. Additional adversarial task discriminator is also used to obtain more task-independent sharing information. Experiments show that the proposed approaches achieve the state-of-the-art performance on the small scale ATIS benchmark and significantly improve the semi-supervised performance on a large-scale dataset.

Index Terms— Spoken language understanding, Multi-task learning, Semi-supervised learning, Adversarial task discriminator

1. INTRODUCTION

The Spoken Language Understanding (SLU) module is a key component of the goal-oriented spoken dialogue system (SDS), parsing the users’ utterances into the corresponding semantic concepts. For example, the sentence “Show me flights from Boston to New York” can be parsed into (fromloc.city.name=Boston, toloc.city.name=New York)[1]. Typically, it is regarded as a slot filling task, assigning one predefined semantic slot tag to each word in the utterance [2].

Recent research about statistical slot filling in SLU has focused on recurrent neural network (RNN) [3] and its extensions, such as long-short memory networks (LSTM) [4], encoder-decoder model [5, 6, 7], etc. These traditional methods require large amounts of labeled data to achieve a good performance. However, it is difficult to get sufficient in-domain labeled data for training because the data annotation is labor-intensive and time-consuming [8]. When an existing domain expands or a new one is created, only limited data are available for supervised learning. In recent years, more and more applications of SDS have been released along with the development of the mobile internet, e.g. Apple Siri, Amazon Alexa, Google Home, Microsoft Cortana etc. Lack of supervised data generally results in a locally optimal solution. To alleviate the localization, semi-supervised learning can be used to access to those unseen inputs efficiently.


Inspired by the success of shared-private models [13], we propose an adversarial multi-task learning method for SLU which learns generalized and unsupervised knowledge and regularizes the slot tagging model. The motivation is to tune the slot tagging model by integrating general language information from unlabeled data. Specifically, a bidirectional language model (BLM) and a slot tagging model (STM) are combined by a shared space and two task-specific private spaces. The BLM learns underlying general patterns of semantic and syntactic composition [11] with abundant unsupervised data, while the STM obtains supervised knowledge with limited labeled data. The shared space is trained for both tasks. Furthermore, we investigate an adversarial task discriminator as a rival to the shared space. The aim of the task discriminator is to figure out which task are the shared features trained for at each time. In order to confuse the task discriminator, the shared space is forced to extract task-invariant knowledge and jettison task-specific information. The task discriminator

The corresponding author is Kai Yu. This work has been supported by the National Key Research and Development Program of China under Grant No.2017YFB1002102, and the China NSFC projects (No. 61573241). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.
is applied on word-level or sentence-level. Unlike Chen et al who trained their models only for Chinese word segmentation task by supervised learning on multiple segmentation criteria with the same data source, we adopt distinctive training objectives, methods and data sources for each task.

The experiments are conducted on the standard ATIS corpus and a large-scale dataset which contains about 30-thousand utterances from three different domains. For the small dataset, the proposed methods obtain the state-of-the-art performance compared to published models. For the large dataset, the models are evaluated on semi-supervised learning performance with different amounts of labeled data. The results show that the proposed approaches perform better than previous methods.

We describe the proposed adversarial multi-task learning methods in Section 2 and the training procedure in Section 3. The experimental results and analysis are given in Section 4. Then Section 5 provides the conclusion of this paper.

2. ADVERSARIAL MULTI-TASK LEARNING

The slot filling is typically considered as a sequence labeling problem. Given an input sequence with \( n \) words \( w = \{ w_1, w_2, \cdots, w_n \} \), slot filling is to predict the output (slot-tag) sequence \( t = \{ t_1, t_2, \cdots, t_n \} \).

The traditional slot tagging model is only optimised based on the ground truth of labels. Because the number of each word in the input is not greater than one and the size of labeled data is limited, slot tags actually contribute little to a generalized SLU model. Inspired by taking language modeling as a supplementary objective [11], we integrate a unidirectional or bidirectional language model with the slot tagging model. The LM can learn more general patterns of the semantic and syntactic composition without any additional labeled data. The unidirectional LM (ULM) predicts the next word, while the bidirectional LM (BLM) consists of two separate ULMs, predicting the next word and the previous word without share-weighting.

Instead of simply adding an additional parallel output layer, we investigate an adversarial multi-task model using the shared-private framework as illustrated in Fig.1. Each task has its own private space (STM in green, LM in blue) and shares a joint space (in yellow). The first step is to map the current word \( w_t \) to a word embedding \( e_t \). A BLSTM is adopted as the hidden layer for the shared, STM-specific and BLM-specific space, and an LSTM for ULM-specific space. Each LSTM takes as input the hidden state from the previous time step and the word embedding at the current step \( i \):

\[
\begin{align*}
\vec{h}_i^k &= LST M_k(e_t, \vec{h}_{i-1}^k) \\
\hat{h}_i^k &= LST M_k(e_t, \hat{h}_{i+1}^k)
\end{align*}
\]

where \( k \in \{ t, l, s \} \), \( t \) refers to slot tagging space, \( l \) refers to language modeling space, and \( s \) refers to the shared space.

The task-specific output layer estimates the probability of the slot tag or word respectively at time frame \( i \):

\[
\begin{align*}
\hat{o}_i^t &= \sigma(\vec{W}^t[\vec{h}_i^t; \hat{h}_i^t; \vec{h}_i^s]) \\
\hat{o}_i^l &= \sigma(\vec{W}^l[\vec{h}_i^l; \vec{h}_i^s]) \\
\hat{o}_i^s &= \sigma(\vec{W}^s[\vec{h}_i^s])
\end{align*}
\]

where \([\cdot]\) is the concatenation operator, \( \vec{W}^t, \vec{W}^l, \vec{W}^s \) are independent weight matrices and \( \sigma \) denotes the softmax layer which predicts a normalized distribution over all possible tags or words. Then the model can be trained by minimizing the cross-entropy loss between predictive distribution \( \hat{o}_i \) and the ground truth label (slot tag \( t_i \), next word \( w_{i+1} \) or previous word \( w_{i-1} \)).

Inspired by the impressive success of adversarial methods in deep generative modeling [14], cross-lingual task [15], and domain adaptation [16], we investigate a task discriminator to make sure the shared space only contains task-invariant features. Specifically, the task discriminator takes as input shared features and predicts whether slot tagging task or language modeling task the inputs are trained for. To confuse the discriminator, the shared space is encouraged to extract task-invariant features. To make a strong rival to the shared model, we analyze the task discriminator on word-level and sentence-level.

The word-level discriminator \( D^{(w)} \) calculates the average of the shared feature \( \vec{h}_i^s = [\vec{h}_i^t; \vec{h}_i^l] \) at each time-frame \( i \) after linear transformation, while the sentence-level discriminator \( D^{(s)} \) selects the most salient feature from the sequence of shared features. Then they predict the probability of the task indication \( t_i \), which equals 1 when \( w \) is used for STM training and equals 0 for LM training.

\[
P_d(y|w; \theta^d, \theta^s) = \begin{cases} 
\frac{1}{n} \sum_{i=1}^{n} \sigma(\vec{W}^d \vec{h}_i^s) & \text{for } D^{(w)}; \\
\sigma(\vec{W}^d \max_{1 \leq i \leq n} \vec{h}_i^s) & \text{for } D^{(s)} 
\end{cases}
\]
where $W^d$ is a weighting matrix in the task discriminator space, $\theta^d$ and $\theta^s$ are the parameters in the discriminator and shared space respectively.

3. TRAINING PROCEDURE

We first present the training objective for each component, then show the overall training algorithm. The training objective of $D$ is to maximize the probability of correctly distinguishing the task which input features are used for, while the shared space attempts to confuse the task discriminator:

$$\max_{\theta^d}\mathcal{L}^d = \sum_{(w,y)\in\text{data}} \log P_d(y|w; \theta^d, \theta^s)$$  \hspace{2cm} (6)$$

$$\min_{\theta^s}\mathcal{L}^s = \sum_{(w,y)\in\text{data}} \log P_d(y|w; \theta^d, \theta^s)$$  \hspace{2cm} (7)$$

where the dataset data could be labeled or unlabeled.

For slot tagging task and language modeling task, the objective functions can be computed as:

$$\max_{\theta^*, \theta^t} \mathcal{L}^i = \sum_{(w,t)\in\text{data}^i} \sum_{i=1}^{\mid w \mid} \log P_i(t_i|w_i; \theta^*, \theta^t)$$  \hspace{2cm} (8)$$

$$\max_{\theta^*, \theta^t} \mathcal{L}^3 = \sum_{w\in\text{data}} \sum_{i=1}^{\mid w \mid} \log P_i(w_{i+1}|w_i; \theta^*, \theta^t)$$  \hspace{2cm} (9)$$

$$\max_{\theta^*, \theta^t} \mathcal{L}^4 = \sum_{w\in\text{data}} \sum_{i=1}^{\mid w \mid} \log P_i(w_{i-1}|w_i; \theta^*, \theta^t)$$  \hspace{2cm} (10)$$

where data$^i$ is the labeled part of data, in which each word $w_i$ is annotated with a slot tag $t_i$. $P_i(\cdot|w_i)$ is the probability over slot tags, and $P_i(\cdot|w_{i+1})$ is that over the vocabulary. $w_0$ and $w_1$ are assigned to the sentence start $<s>$ and the sentence end $</s>$ respectively.

Algorithm 1 shows the overall adversarial training procedure. The discriminator and shared model conduct a minimax competition through Eq.(6) and Eq.(7) which improve each other until their feature representations are close enough. The shared model is encouraged to extract the generalized features from abundant raw utterances. In addition, Eq.(9) and Eq.(10) learn underlying semantic and syntactic knowledge. Eq.(8) as a traditional supervised learning objective drives the slot tagging model to perform well on labeled data and transfer the supervised information to unlabeled data.

4. EXPERIMENTS

The proposed model and other methods are first evaluated on the Air Travel Information System (ATIS) benchmark. Then we demonstrate the effectiveness of the proposed model on semi-supervised learning with the different numbers of labeled utterances from a large-scale dataset. The experimental results show that our methods substantially improve over traditional semi-supervised methods in the slot filling task.

4.1. Experimental Setup

For all architectures, the dimensions of word embeddings and BLSTM hidden units are set to 100. At each time-frame, the SLU model takes the current word as input without any context words. For training, the network parameters are randomly initialized in accordance with the uniform distribution (-0.2, 0.2) and updated by stochastic gradient descent (SGD). The dropout with a probability of 0.5 is applied to the non-recurrent connections for regularization. Different learning rates are tried by grid-search in the range of [0.008, 0.03] and keep it for 100 epochs. The F1-scores of slot filling on the test set whose corresponding models perform best on validation are reported.

For adversarial training, the task discriminator and the multi-task model are optimized with the minibatch size of 10. At each iteration, the slot tagging model is trained on labeled data by the supervised algorithm, and the language model is trained on labeled and unlabeled data by self-supervising. Meanwhile, the shared model and the task discriminator are trained by a minimax game.

4.2. ATIS Experiment

ATIS includes 4,978 training sentences and 893 test ones from the only air travel domain. Because a slot may be mapped to several continuous words, we follow the popular In/Out/Begin (IOB) representation. The number of different slot tags is 84 (127 if IOB prefixes are considered). We randomly select 80% from the training sentences as the training set while the rest as validation [17]. The following methods are investigated:

STM: It is a simple supervised model, using BLSTM as the hidden layer for slot filling task [4].

STM+LM: It pre-trains a language model first, then initializes the word embeddings of an STM by those of the
well-trained LM. The word embeddings are updated during the training process for SLU.

MTL$e$: It exploits the multi-task learning for STM and LM. These two tasks share the embedding layer.

MTL$e$+$h$: Similar to [11], STM and LM share the embedding and hidden layer.

**SPM**: It uses the shared-private model for multi-task learning. Compared to MTL$e$, it adds a shared hidden space to improve the performance. Compared to MTL$e$+$h$, it adds private hidden spaces for each task. The output layer inputs both shared and private features. Unidirectional SPM (USPM) contains an STM and a forward LM while bidirectional SPM (BSM) has an additional backward LM (the dashed blocks in Fig.1).

**SPM$^+$D**: It is the exact model illustrated in Fig.1. Compared to SPM, a task discriminator is added to the framework.

**SPM$^+$D**: Compared to SPM$^+$D, it eliminates the LM-specific space and remains unchanged.

Table 1 shows the performance of these methods on ATIS dataset. The superscript of $F$ indicates the LM in the model is unidirectional ($F^1$) or bidirectional ($F^B$).

The results are illustrated in Table 2. Only bidirectional methods are shown, which have been proved to be more effective on ATIS. As a successful method, PSEUDO performs a pipeline in three stages: training an STM with the labeled data, generating pseudo labels for the unlabeled data by the pre-trained model, and retain an STM with labeled and pseudo-labeled data simultaneously [9, 10].

From Table 2, we can see the proposed BSM and BSM$^+$D constantly achieve better performance than other methods for different labeled sets. Our method improves significantly (99.9%) over STM on all datasets. Compared with MTL$e$, our method has a significant level of 99.9% on 5k set, and of 99.5% on 10k set. However, the improvement is not significant on 15k set. Similarly, our method improves significantly (99.8%) over PSEUDO on 5k and 10k set, but not significantly (over 95%) in 15k set.

The experiments indicate that our semi-supervised learning model is more efficient when the labeled data is limited and the data for LM is more sufficient. While BLM exploits the unsupervised knowledge, the shared-private framework and adversarial training make the slot tagging model more generalized and perform better on unseen samples.

### 5. Conclusion

In this paper, we propose an adversarial multi-task learning method for semi-supervised training on SLU, which alleviates the dependence on labeled data. A bidirectional language model is intersected with the slot tagging model by sharing the joint space and monopolizing a private LM space. Therefore, the slot tagging model acquires generalized language knowledge from the shared space and obtains supervised information from its private STM space. In addition, a task discriminator is used to force the shared space to discard task-specific information. The proposed methods achieve the state-of-the-art performance on ATIS benchmark, and significantly outperform previous models with limited labeled data on the large-scale dataset.
6. REFERENCES


