Robust Spoken Language Understanding with RL-based Value Error Recovery

Chen Liu[⋆], Su Zhu[⋆], Lu Chen, and Kai Yu[⋆⋆]

MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{chris-chen,paul2204,chenlusz,kai.yu}@sjtu.edu.cn

Abstract. Spoken Language Understanding (SLU) aims to extract structured semantic representations (e.g., *slot-value* pairs) from speech recognized texts, which suffers from errors of Automatic Speech Recognition (ASR). To alleviate the problem caused by ASR-errors, previous works may apply input adaptations to the speech recognized texts, or correct ASR errors in predicted values by searching the most similar candidates in pronunciation. However, these two methods are applied separately and independently. In this work, we propose a new robust SLU framework to guide the SLU input adaptation with a rule-based value error recovery module. The framework consists of a slot tagging model and a rule-based value error recovery module. We pursue on an adapted slot tagging model which can extract potential slot-value pairs mentioned in ASR hypotheses and is suitable for the existing value error recovery module. After the value error recovery, we can achieve a supervision signal (reward) by comparing refined slot-value pairs with annotations. Since operations of the value error recovery are non-differentiable, we exploit policy gradient based Reinforcement Learning (RL) to optimize the SLU model. Extensive experiments on the public CATSLU dataset show the effectiveness of our proposed approach, which can improve the robustness of SLU and outperform the baselines by significant margins.

Keywords: Spoken Language Understanding · Robustness · RL

1 Introduction

The Spoken Language Understanding (SLU) module is a key component of Spoken Dialogue System (SDS), parsing user's utterances into structured semantic forms. For example, "*I want to go to Suzhou not Shanghai*" can be parsed into "{*inform(dest=Suzhou), deny(dest=Shanghai)*}". It can be usually formulated as a sequence labelling problem to extract values (e.g., *Suzhou* and *Shanghai*) for certain semantic slots (attributes, e.g., *inform-dest* and *deny-dest*).

[⋆] Chen Liu and Su Zhu contributed equally to this work.

[⋆⋆] Lu Chen and Kai Yu are the corresponding authors.

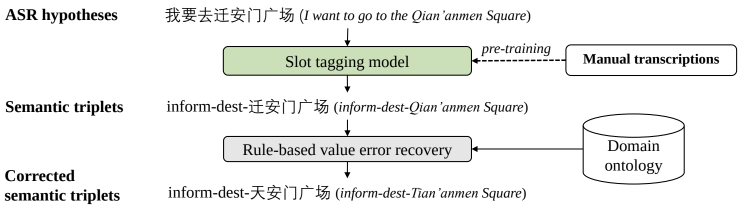

ASR hypotheses
我要去迁安门广场 (*I want to go to the Qian'anmen Square*)
Slot tagging model
pre-training
Manual transcriptions
Semantic triplets
inform-dest-迁安门广场 (*inform-dest-Qian'anmen Square*)
Rule-based value error recovery
Domain ontology
Corrected semantic triplets
inform-dest-天安门广场 (*inform-dest-Tian'anmen Square*)

Fig. 1. An overview of the robust SLU framework, which is composed of two main components: a slot tagging model and a rule-based value error recovery module. During evaluation, only ASR hypotheses are fed into the two modules to generate the final semantic form.

It is crucial for SLU to be robust to speech recognition errors, since ASR-errors would be propagated to the downstream SLU model. By ignoring ASR-errors, it promotes rapid development of natural language processing (NLP) algorithms for SLU [7, 16, 5, 2] where SLU models are trained and evaluated on manual transcriptions and even natural language texts. Once ASR hypotheses are used as input for evaluation, it will lead to a sharp decrease in SLU performance [20].

ASR-errors may give rise to two issues: 1) inputs for training and evaluation are mismatched; 2) Sequence labelling models extract values directly from ASR hypotheses, which may contain wrong words. Previous works try to overcome these problems in two ways: 1) Adaptive training approaches are introduced to transfer the SLU model trained on manual transcriptions to ASR hypotheses [18, 9]. 2) Other works adopt rule-based post-processing techniques to refine the predicted values with the most similar candidates in pronunciation [11, 13, 4]. However, this value error recovery module is usually fixed and independent of the SLU model.

To overcome the above problems, we propose a new robust SLU framework to guide the former SLU model trained with a rule-based value error recovery. As illustrated in Figure 1, it consists of a slot tagging model and a value error recovery module. The slot tagging model is pre-trained on manual transcriptions, which considers SLU as a sequence labelling problem. To alleviate the input mismatched issue, it is adaptively trained on ASR hypotheses. The value error recovery module is exploited to correct potential ASR-errors in predicted values of the slot tagging model, which is built upon a pre-defined domain ontology.

However, there are no word-aligned annotations for ASR hypotheses to fine-tune the slot tagging model. Thus, we indirectly guide the adaptive training of the slot tagging model on ASR hypotheses by utilizing supervisions of the value error recovery. Concretely, we can compute a reward by measuring predicted semantic forms after the value error recovery with annotations, and then optimize the slot tagging model by maximizing the expected reward. Since operations in the value error recovery are non-differentiable, we use a policy gradient [10] based reinforcement learning (RL) approach for optimizing.

Table 1. An example of user utterance (manual transcription and ASR hypothesis) and its semantic annotations.

$\hat{x}$	I want to go to Suzhou not Shanghai
x	I one goal to Suizhou not Shanghai
y	inform(dest="Suzhou"); deny(dest="Shanghai")
$\hat{o}$	I$_{[O]}$ want$_{[O]}$ to$_{[O]}$ go$_{[O]}$ to$_{[O]}$ Suzhou$_{[B\text{-inform-dest}]}$ not$_{[O]}$ Shanghai$_{[B\text{-deny-dest}]}$

We conduct an empirical study of our proposed method and a set of carefully selected state-of-the-art baselines on the 1st CATSLU challenge dataset [20], which is a large-scale Chinese SLU dataset collected from a real-world SDS application. Experiment results confirm that our proposed method can outperform the baselines significantly.

In summary, this paper makes the following contributions:

- To the best of our knowledge, this is the first work to train a slot tagging model guided by a rule-based value error recovery module. It tends to learn a robust slot tagging model for easier and more accurate value error recovery.
- We propose to optimize the slot tagging model with indirect supervision and RL approach, which does not require word-aligned annotations on ASR hypotheses. Ablation study confirms that RL training can give improvements even without the value error recovery module.

2 Proposed Method

In this section, we provide details of our proposed robust SLU framework, which consists of a sequence labelling based slot tagging model and a rule-based value error recovery (VER) module. To guide the training of the slot tagging model on ASR hypotheses with the VER, we propose an RL-based training algorithm.

Let $x = (x_1 \cdots x_{|x|})$ and $\hat{x} = (\hat{x}_1 \cdots \hat{x}_{|\hat{x}|})$ denote the ASR 1-best hypothesis and manual transcription of one utterance respectively. Its semantic representation (i.e., *act(slot=value)* triplets) is annotated on $\hat{x}$. Thus, it is easy to get the word-level tags on $\hat{x}$, $\hat{o} = (\hat{o}_1 \cdots \hat{o}_{|\hat{x}|})$, which is in Begin/In/Out (BIO) schema (e.g., O, B-inform-dest, B-deny-dest), as shown in Table 1.

2.1 Slot Tagging Model

For slot tagging, we adopt an encoder-decoder model with focus mechanism [19] to model the label dependency. A BLSTM encoder reads an input sequence $\hat{x}$, and generates the hidden states at the t-th time-step via

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}];\ \overrightarrow{\mathbf{h}_t} = \mathrm{LSTM}_\mathrm{f}(\overrightarrow{\mathbf{h}_{t-1}}, \phi(\hat{x}_t));\ \overleftarrow{\mathbf{h}_t} = \mathrm{LSTM}_\mathrm{b}(\overleftarrow{\mathbf{h}_{t+1}}, \phi(\hat{x}_t)) \qquad (1)$$

where $\phi(\cdot)$ is a word embedding function, $[\cdot;\cdot]$ denotes vector concatenation, LSTM_f and LSTM_b represent the forward and backward LSTMs, respectively.

Then, an LSTM decoder updates its hidden states at the t-th time-step recursively by $\mathbf{s}_t = \mathrm{LSTM}(\mathbf{s}_{t-1}, [\psi(\hat{o}_{t-1}); \mathbf{h}_t])$, where $\psi(\cdot)$ is a label embedding function, and $\mathbf{s}_0$ is initialized with $\overleftarrow{\mathbf{h}}_1$. Finally, the slot tag $\hat{o}_t$ is generated by

$$P(\hat{o}_t|\hat{o}_{<t}; \hat{x}) = \mathrm{Softmax}(\mathbf{W}\mathbf{s}_t + \mathbf{b}) \tag{2}$$

where $\mathbf{W}$ and $\mathbf{b}$ are parameters for the linear output layer.

Following the BIO schema, we can restructure the predicted slot-tag sequence aligned with the input sequence to obtain a set of *act(slot=value)* triplets.

2.2 Value Error Recovery (VER) Module

During evaluation, ASR hypotheses are fed into the slot tagging model to get the *act(slot=value)* triplets, which may retain ASR errors in the values. Thus, a value error recovery (VER) module based on a pre-defined domain ontology [1] is applied to refine wrong values. Li et al. [4] search through the ontology to find the most similar candidate with minimum edit distance. While the calculation of the minimum edit distance is time-consuming and hard to be parallelized, we exploit an N-gram based cosine distance to accelerate this process.

Generally, we define the n-gram set of a word sequence $w = (w_1, \cdots, w_T)$ as $\mathrm{Ngram}(w, n) = \{(w_i, \cdots, w_{i+n-1}) \mid i = \{1, \cdots, T-n+1\}\}$. The n-grams of all values in the domain ontology $\mathcal{O}$ constitute a vocabulary, denoted as $\mathrm{Ngram}(\mathcal{O}, n)$. Given a predicted semantic triplet $a(s = v)$, where the value v is a word sequence $v = (v_1, \cdots, v_T)$. Then, we get a binary-valued feature vector $\mathbf{d}'(v) = (d_1^v, \cdots, d_L^v)$ for the predicted value v, where $L = |\,\mathrm{Ngram}(\mathcal{O}, n)|$ and $d_j^v = \mathbb{1}_{\{\mathrm{Ngram}(\mathcal{O},n)_j \in \mathrm{Ngram}(v,n)\}}$. Finally, we normalize it to be a unit vector, $\mathbf{d}(v) = \mathbf{d}'(v)/||\mathbf{d}'(v)||$.

Based on the domain ontology, there is a value candidate set [2] corresponding to the act a and slot s, $\mathcal{V}_{a,s} = (\bar{v}^1, \cdots, \bar{v}^M)$, where M is the number of possible values. Therefore, the value candidate set can be represented as an $L \times M$ feature matrix $\mathbf{D}(\mathcal{V}_{a,s})$, the k-th column of which is $\mathbf{d}(\bar{v}^k)$. We believe that the more n-grams two values share, the more similar they are. Thus, we calculate the cosine similarity score between $\mathbf{d}(v)$ and each column vector in $\mathbf{D}(\mathcal{V}_{a,s})$ as:

$$\mathbf{sim}_{\mathrm{word}}(v, \mathcal{V}_{a,s}) = \mathbf{D}(\mathcal{V}_{a,s})^\top \mathbf{d}(v), \ \in \mathbb{R}^M \tag{3}$$

Since ASR tends to produce words similar in pronunciation, we convert word sequences of values into pronunciation sequences with a pre-defined pronunciation dictionary. For example, “上海 (Shanghai)” can be converted into “*sh ang h ai*”. Therefore, we get another similarity vector by considering the pronunciation n-grams, denoted as $\mathbf{sim}_{\mathrm{pron}}(v, \mathcal{V}_{a,s})$. The final similarity vector is obtained by averaging the two vectors, i.e.,

$$\mathbf{sim}(v, \mathcal{V}_{a,s}) = \lambda \mathbf{sim}_{\mathrm{word}}(v, \mathcal{V}_{a,s}) + (1-\lambda)\mathbf{sim}_{\mathrm{pron}}(v, \mathcal{V}_{a,s}) \tag{4}$$

[1] All possible value candidates of each slot are provided in the domain ontology.

[2] E.g., the value candidate set for slot *address* can be all available addresses saved in the database of a dialogue system.

Algorithm 1 Training algorithm

```
Input: Manual transcriptions with word-aligned labels $\mathcal{D}_{tscp} = \{(\hat{x}, \hat{o})\}$; ASR hypotheses with utterance-level labels $\mathcal{D}_{hyp} = \{(x, y)\}$; reward function $R(\cdot)$.
Output: Robust slot tagging model $\Theta$
Initialize $\Theta$ randomly;
repeat                                                        ▷ Pre-training stage
    Sample $(\hat{x}, \hat{o})$ from $\mathcal{D}_{tscp}$;
    Update the model: $\Theta \leftarrow \Theta - \eta_1 \nabla_{\Theta} L_{tag}(\Theta)$;
until convergence
repeat                                                        ▷ RL-training stage
    Sample $(x, y)$ from $\mathcal{D}_{hyp}$;
    $K$ groups of semantic labels $\tilde{y}^1, ..., \tilde{y}^K$ are generated after beam search and value error recovery by feeding $x$;
    for $k = 1, ..., K$ do
        Compute reward $R(x, y, \tilde{y}^k)$ by Eqn.(6);
    end for
    Compute policy gradient $\nabla_{\Theta} \hat{E}[R]$ by Eqn.(7);
    Update the model: $\Theta \leftarrow \Theta + \eta_2 \nabla_{\Theta} \hat{E}[R]$;
    Sample $(\hat{x}, \hat{o})$ from $\mathcal{D}_{tscp}$;
    Update the model: $\Theta \leftarrow \Theta - \eta_1 \nabla_{\Theta} L_{tag}(\Theta)$;
until convergence
```

where λ is a balancing parameter (0.5 in our experiments). So far, we can easily find the best alternative value $\bar{v}^k$, where $k = \text{argmax}(\mathbf{sim}(v, \mathcal{V}_{a,s}))$.

Although some slots have numerous possible values in the domain ontology, it is much efficient by simply performing matrix multiplication. We also set a threshold (0.5 in this paper) to reject a bad error recovery.

2.3 Training Procedure

We propose to guide the adaptive training of the slot tagging on ASR hypotheses with the value error recovery module. It takes two advantages: 1) mitigating the input mismatch problem of training and testing; 2) not requiring word-aligned annotations on ASR hypotheses. Meanwhile, it tends to learn a robust slot tagging model suitable for the value error recovery.

We apply the policy gradient based reinforcement learning (RL) algorithm to handle non-differentiable operations. To prune the large search space, the model is pre-trained with annotated transcriptions to bootstrap the RL-training. The whole training procedure contains two stages, as described below.

Pre-training Let $\mathcal{D}_{tscp} = \{(\hat{x}, \hat{o})\}$ denote manual transcriptions with aligned labels. The slot tagging model (let Θ refer to the model parameters) is trained by minimizing a negative log-likelihood loss:

$$L_{tag}(\Theta) = -\sum_{(\hat{x},\hat{o}) \in \mathcal{D}_{tscp}} \log P(\hat{o}|\hat{x}; \Theta). \tag{5}$$

RL-training ASR hypotheses coupled with unaligned labels, $\mathcal{D}_{hyp} = \{(x, y)\}$, are utilized for the adaptive training. The slot tagging model, $P(o|x; \Theta)$, samples via beam search to produce K tag sequences, and then K sets of *act(slot=value)* triplets. Finally, corrected semantic triplets $\{\tilde{y}^k\}_{k=1}^K$ are generated after VER module. For each beam, the reward is considered at both triplet-level and utterance-level:

$$R(x, y, \tilde{y}^k) = R_{\text{triplet}} + R_{\text{utt}} = \left(1 - \frac{\text{FP}(y, \tilde{y}^k) + \text{FN}(y, \tilde{y}^k)}{\mid y \mid}\right) + \mathbb{1}_{\{y=\tilde{y}^k\}} \tag{6}$$

where the first term punishes false-positives (FP) and false-negatives (FN) of *act(slot=value)* triplets, and the second term is a binary value indicating whether the entire triplets of one utterance is predicted correctly.

The model is optimized by maximizing the expected cumulative rewards using policy gradient descent. The policy gradient can be calculated as:

$$\nabla_\Theta \hat{E}[R] = \frac{1}{K} \sum_{k=1}^{K} \Big[R(x, y, \tilde{y}^k) - B(r)\Big] \cdot \nabla_\Theta \log P(\tilde{y}^k | x; \Theta) \tag{7}$$

where $B(r) = \frac{1}{K} \sum_{k=1}^{K} R(x, y, \tilde{y}^k)$ is a baseline for reducing the variance of gradient estimation, obtained by averaging the rewards inside a beam.

In order to stabilize the training process, it is beneficial to train batches with $\mathcal{D}_{tscp}$ and $\mathcal{D}_{hyp}$ iteratively. The training framework is shown in Algorithm 1.

3 Experiments

3.1 Experimental Setup

We conduct our experiments on the 1st Chinese Audio-Textual Spoken Language Understanding Challenge (CATSLU)[3] dataset containing four dialogue domains (*map*, *music*, *video*, *weather*). The statistics of the CATSLU dataset are demonstrated in detail in Zhu et al. [20].

Slot tagging is modeled at Chinese character level. The 200-dim char embedding is initialized by pre-training LSTM based bidirectional language models (biLMs) with zhwiki [4] corpus. LSTMs are single-layer with 256 hidden units. In the training process, parameters are uniformly sampled within the range of $(-0.2, 0.2)$. Dropout with a probability of 0.5 is applied to non-recurrent layers. We choose Adam [3] as our optimizer. For the learning rate, we set η_1=1e-3 and η_2=5e-4 fixed during training. The maximum norm for gradient clipping is set to 5. In the RL-training stage, the beam search sampling size K is set to 10. In the decoding stage, the beam size is 5. The best model is selected according to the performance on the validation set, and we measure both F$_1$-score of *act(slot=value)* triplets and utterance-level accuracy.

[3] https://sites.google.com/view/CATSLU

[4] https://dumps.wikimedia.org/zhwiki/latest

3.2 Baselines

We compare the proposed method with strong baselines for robust SLU:

- **HD**: Hierarchical Decoding model proposed in Zhao et al. [17], which performs slot filling in a generative way with only unaligned data ($\mathcal{D}_{hyp}$).
- **Focus**: BLSTM-Focus model as described in section 2.1 for slot tagging.
- **UA**: Unsupervised Adaptation method [18] utilizes the language modelling task to transfer the slot tagging model from manual transcriptions to ASR hypotheses.
- **DA**: Data Augmentation methods are also involved to predict pseudo labels aligned with ASR hypotheses; thus the pseudo samples can be exploited to train a robust slot tagging model. (1) Gen: ASR hypotheses are fed into the pre-trained slot tagging model to generate pseudo labels. (2) Align: ASR hypotheses are aligned with manual transcriptions via achieving minimum edit-distance, and then the aligned labels of words in transcriptions can be assigned to the corresponding words in ASR hypotheses.

3.3 Main Results

In this section, the main results on the test set compared with the baselines are demonstrated in Table 2. In the evaluation stage of all baselines and our approach, VER is applied for post-processing. Lexicon features are added as additional input features, same as what Li et al. [4] did.

Overall, the SLU models perform better *with* auxiliary lexicon features. For basic slot filling models, Focus performs much better than HD, showing that the sequence labelling based slot tagging model is more generalizable than a generative model. The results of the oracle experiments suggest that ASR hypotheses largely degrade the performance. By adapting to ASR hypotheses, UA performs slightly better on some domains, but the average result drops instead. With lexicon features, by augmenting the training data with pseudo aligned hypotheses (DA), both Gen and Align can beat the basic model, indicating that DA methods are beneficial for improving robustness to ASR hypotheses.

With lexicon features, our proposed method outperforms the Focus model in all domains significantly, achieving an average improvement of 0.8% in F_1-score and 1.5% in joint-accuracy, which reveals the benefit of VER guided training. Our model also surpasses the best baseline (DA-Align), indicating that it is less effective to merely augment the data with pseudo aligned ASR texts.

We also attempt to employ only transcriptions in training (the second-to-last row), that is, replace $\mathcal{D}_{hyp}$ (on line 7) with $\mathcal{D}_{tscp}$ in Algorithm 1. The consistent improvements in all domains compared with Focus prove that the RL loss benefits the slot tagging. On this basis, adaptively involving the ASR hypotheses in training further improves the robustness of the SLU model.

We only compare our model with the “System 1” in Li et al. [4] (the top solution in CATSLU challenge), because their other systems add the validation set in training and leverage audio information for better results. As shown in the table, our proposed method achieves higher average F_1-score and joint-accuracy.

Table 2. Main results *with* or *without* lexicon features. F_1-score(%)/joint-accuracy(%) on the test set of each domain are reported. In the table, *tscp* means manual transcriptions while *hyp* means ASR hypotheses. Our results that significantly outperform the best baseline are marked by † ($p < 0.05$) and ‡ ($p < 0.01$). * denotes oracle experiments, in which manual transcriptions are evaluated.

(a) *with* lexicon features

Models	**train**	**test**	**map**	**music**	**video**	**weather**	**avg.**
HD	hyp	hyp	87.8/84.2	90.5/82.1	88.7/76.1	89.6/82.4	89.2/81.2
Focus*	tscp	tscp	96.4/93.8	97.6/93.3	94.1/85.6	95.5/90.6	95.9/90.8
Focus	tscp	hyp	89.0/84.9	92.8/84.8	91.5/80.9	92.6/86.7	91.5/84.3
UA	tscp+hyp	hyp	88.5/85.2	91.8/83.6	91.2/81.2	91.8/84.9	90.9/83.7
DA-Gen	tscp+hyp	hyp	88.9/85.4	92.2/84.6	92.0/81.4	93.1/87.1	91.5/84.6
DA-Align	tscp+hyp	hyp	89.1/85.5	93.1/85.7	91.5/80.8	93.1/87.0	91.7/84.7
Li et al. [4]	tscp	hyp	87.9/83.8	92.7/85.1	**92.3**/**82.6**	93.0/86.8	91.5/84.6
Proposed	tscp	hyp	89.0/85.3	93.1/85.7	91.9/81.6	93.1/87.6	91.8/85.0
	tscp+hyp	hyp	**90.0**/**86.7**†	**93.8**/**87.0**†	92.0/82.0	**93.4**/**87.7**	**92.3**/**85.8**†

(b) *without* lexicon features

Models	**train**	**test**	**map**	**music**	**video**	**weather**	**avg.**
HD	hyp	hyp	87.9/83.5	87.0/74.1	84.0/64.9	89.2/80.0	87.0/75.6
Focus*	tscp	tscp	96.4/93.9	96.3/89.4	92.8/81.6	94.7/88.7	95.0/88.4
Focus	tscp	hyp	89.4/86.1	92.2/83.6	90.2/77.8	92.4/85.7	91.0/83.3
UA	tscp+hyp	hyp	89.3/86.4	91.9/83.6	90.0/78.4	91.8/85.0	90.8/83.3
DA-Gen	tscp+hyp	hyp	88.7/85.7	91.4/82.5	90.3/78.2	92.2/85.8	90.6/83.1
DA-Align	tscp+hyp	hyp	89.3/85.8	**92.3**/83.6	90.6/77.6	91.8/85.1	91.0/83.0
Proposed	tscp	hyp	89.5/86.3	91.8/82.8	90.8/79.0	92.2/86.1	91.1/83.6
	tscp+hyp	hyp	**89.6**/**86.8**†	92.2/**83.7**	**91.2**/**79.7**‡	**92.6**/**86.2**†	**91.4**/**84.1**‡

3.4 Ablation Study

Ablation Study of the Slot Tagging Model For slot tagging, there are other popular methods like BLSTM and BLSTM-CRF [4]. Table 3 shows the comparison of different slot tagging models. Vanilla BLSTM performs the worst without modeling label dependencies. Focus can achieve the best results in most cases, thus we choose Focus as the backbone model. It should be noted that our proposed framework can be applied to other slot tagging models.

Effect of the Value Error Recovery (VER) Module We apply different post-processing ways for values to examine the effect of the VER module, as shown in Table 4. For HD, Focus and DA-Align, only the evaluation stage is affected by the post-processing. Results show that it is beneficial to delete invalid triplets (i.e., out of the domain ontology) while finding proper value alternative via the VER module brings further improvement. For our proposed method, both training and evaluation stages involve the VER module. Our proposed method makes consistent improvement regardless of the post-processing ways, whereas VER works best. Due to the additional triplet- and utterance-level policy losses

Table 3. Ablation experiments of the slot tagging models. Average F_1-score(%)/joint-accuracy(%) on the test set are reported with or without lexicon features.

Slot tagging models	**lexicon features**	
	✓	✗
BLSTM	90.84/83.92	89.34/80.78
BLSTM-CRF	91.31/**84.49**	90.30/82.43
Focus	**91.46**/84.31	**91.03/83.29**

Table 4. Ablation study of the value error recovery (VER) module. Lexicon features are used in all settings. “None” means no post-processing, “Delete” means simply deleting the triplets that are invalid according to the ontology, and “VER” means applying VER module. We report the average F_1-score(%)/joint-accuracy(%) on the test set.

Models	**post-processing settings**		
	None	Delete	VER
HD	82.47/73.90	87.30/76.05	89.16/81.22
Focus	88.75/81.81	91.10/83.34	91.46/84.31
DA-Align	88.53/82.12	91.25/83.59	91.69/84.74
Proposed	**89.61/83.33**	**91.68/84.17**	**92.28/85.83**

which help adapt the tagging model to ASR hypotheses, improvements are still observed even without the post-processing.

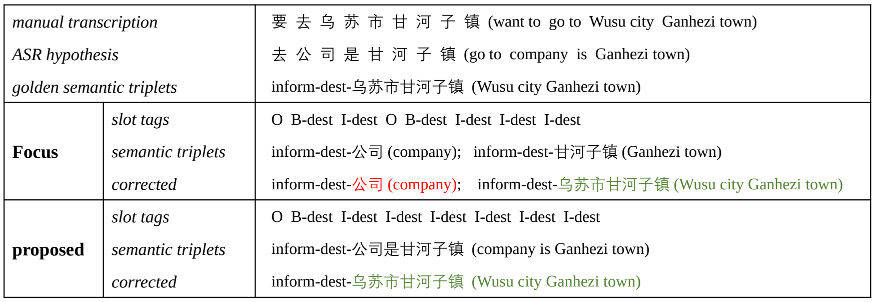
Ablation Study of the Training Procedure Table 5 considers (1) whether to pre-train the slot tagging model and (2) whether to utilize manual transcriptions in RL-training. Results indicate that pre-training using transcriptions helps to bootstrap the RL-training, and introducing transcriptions in RL-training stabilizes the training. Furthermore, the average performance decreases dramatically without these two procedures, which shows that the RL-training gets stuck in local optimum without any experiences about slot tagging from $\mathcal{D}_{tscp}$.

3.5 Analysis

Comparison with Baselines Traditional slot tagging models are supervised by BIO-tags, so ASR hypotheses without BIO-tag annotations cannot be used. In our method, the value error recovery module is applied to the output of the slot tagging model and provides feedback on the prediction. The feedback is utilized as a reward signal of RL-based training to finetune the slot tagging model. We give an example to illustrate how slot tagging benefits from VER guided training in Figure 2. The baseline model recognizes two slot chunks “公司 (company)” and “甘河子镇 (Ganhezi town)” separated by a special word “是 (is)”. The latter value is corrected by the VER module, whereas the former value is retained because it is also available in the value candidates corresponding to the act-slot pair *inform-dest*, resulting in an incorrect triplet. By introducing the VER module during training, the tagging model learns to produce outputs

Table 5. Ablation study of the training procedure for our proposed method. Lexicon features are utilized. Average F$_1$-score(%)/joint-accuracy(%) are reported.

Pre-training on $\mathcal{D}_{tscp}$	**Exploiting** $\mathcal{D}_{tscp}$ **in RL-training**	**avg.**
✓	✓	**92.28/85.83**
✗	✓	91.89/85.12
✓	✗	91.87/85.01
✗	✗	43.06/34.69



manual transcription		要 去 乌 苏 市 甘 河 子 镇 (want to go to Wusu city Ganhezi town)
ASR hypothesis		去 公 司 是 甘 河 子 镇 (go to company is Ganhezi town)
golden semantic triplets		inform-dest-乌苏市甘河子镇 (Wusu city Ganhezi town)
Focus	*slot tags*	O B-dest I-dest O B-dest I-dest I-dest I-dest
	semantic triplets	inform-dest-公司 (company); inform-dest-甘河子镇 (Ganhezi town)
	corrected	inform-dest-公司 (company); inform-dest-乌苏市甘河子镇 (Wusu city Ganhezi town)
proposed	*slot tags*	O B-dest I-dest I-dest I-dest I-dest I-dest I-dest
	semantic triplets	inform-dest-公司是甘河子镇 (company is Ganhezi town)
	corrected	inform-dest-乌苏市甘河子镇 (Wusu city Ganhezi town)

Fig. 2. An example of slot tagging and value error recovery in *map* domain. Note that during evaluation, only the ASR hypothesis is available as input.

more suitable for the subsequent VER module. Although word like "是 (is)" is unlikely to appear in a destination name, the tagging model considers it as a part of the slot, showing the capability to delimit the range of slots softly.

In the view of data used, both DA methods and our proposed method utilize ASR hypotheses during training, while they treat hypotheses in different ways. DA-Gen uses a pre-trained tagging model to produce pseudo labels for ASR hypotheses. Therefore, noisy data is included for training, which will have a negative impact. In our proposed method, we can train the slot tagging model on ASR hypotheses with unaligned labels (i.e., *act(slot=value)* triplets in this paper) directly.

Different Character Error Rate (CER) To further investigate why our model achieves higher performance, we split the test set into various groups according to the character error rate (CER) of the ASR hypotheses, and compare our proposed method with the Focus and DA-Align baselines. The results in the four domains are presented in Figure 3. With the increase of CER, the F$_1$-scores decline sharply. For utterances with low CERs (e.g., less than 10%), there is no significant difference (under 1%) between the baselines and our model. As the CER gets higher, our model can outperform the Focus and DA-Align by a larger margin. The improvements are particularly dramatic when the CER is higher than 90%. Note that exceptions happen occasionally, but in these cases, the amount of data is too small to draw a reliable conclusion. This finding further proves the robustness of our method against noisy ASR data.

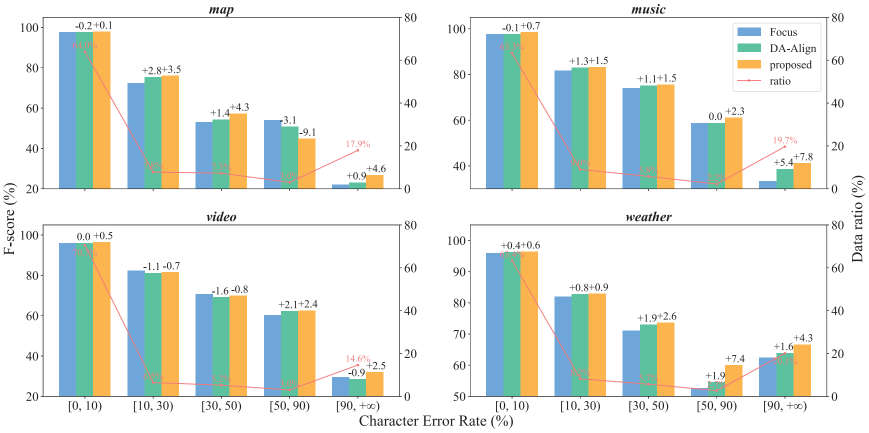


Fig. 3. F_1-scores of our proposed model (with lexicon features) on the test set across four domains with various CER. The data ratio of each group is displayed in the form of line chart. The differences between the Focus baseline and the other two models are annotated on the figure for clarity.

4 Related Work

SLU is often regarded as a sequence labelling problem modelled with Recurrent Neural Network (RNN) [7, 16, 5, 19] and recent transformers [1, 8]. However, most of them assume that there are no ASR errors. To improve the robustness of SLU to ASR errors, previous works may apply input adaptations to reduce the gap between training and testing [18, 9], or correct predicted values by searching the most similar candidates in pronunciation [11, 13, 4]. However, these two methods are not optimized jointly. There are other works to directly train the SLU model on ASR hypotheses [12, 15, 6]. However, these methods require qualified aligned data annotation on ASR hypotheses, which costs a lot.

Except for the sequence labelling problem, SLU can also be directly considered as an unaligned task where outputs are semantic forms. In this view, unaligned annotations (semantic forms) can be transferred from manual transcriptions to ASR hypotheses straightforwardly. With unaligned data, SLU can be considered as a classification task [14] or a generative task [17]. These methods do not require word-aligned labels but may lose generalization capability to unseen samples, which is confirmed by the HD baseline in our experiments.

5 Conclusion

In this paper, we propose a robust SLU framework with a slot tagging model and value error recovery module. The value error recovery is utilized to guide the adaptive training of the slot tagging model on ASR hypotheses with reinforcement learning. Extensive experiments confirm that our model is more robust to ASR errors than the baselines.

Acknowledgements. We thank the anonymous reviewers for their thoughtful comments. This work has been supported by the National Key Research and Development Program of China (Grant No. 2017YFB1002102) and Shanghai Jiao Tong University Scientific and Technological Innovation Funds (YG2020YQ01).

References

1. Chen, Q., Zhuo, Z., Wang, W.: BERT for joint intent classification and slot filling. arXiv preprint arXiv:1902.10909 (2019)
2. Goo, C.W., Gao, G., Hsu, Y.K., Huo, C.L., Chen, T.C., Hsu, K.W., Chen, Y.N.: Slot-gated modeling for joint slot filling and intent prediction. In: NAACL (2018)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Li, H., Liu, C., Zhu, S., Yu, K.: Robust spoken language understanding with acoustic and domain knowledge. In: ICMI. pp. 531–535 (2019)
5. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: INTERSPEECH. pp. 685–689 (2016)
6. Liu, C., Zhu, S., Zhao, Z., Cao, R., Chen, L., Yu, K.: Jointly encoding word confusion network and dialogue context with bert for spoken language understanding. arXiv preprint arXiv:2005.11640 (2020)
7. Mesnil, G., He, X., Deng, L., Bengio, Y.: Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: INTERSPEECH. pp. 3771–3775 (2013)
8. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proc. EMNLP-IJCNLP. pp. 2078–2087 (2019)
9. Schumann, R., Angkititrakul, P.: Incorporating ASR errors with attention-based, jointly trained RNN for intent detection and slot filling. In: ICASSP (2018)
10. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: NeurIPS (2000)
11. Tan, C., Ling, Z.: Multi-classification model for spoken language understanding. In: ICMI. pp. 526–530 (2019)
12. Tür, G., Deoras, A., Hakkani-Tür, D.: Semantic parsing using word confusion networks with conditional random fields. In: INTERSPEECH. pp. 2579–2583 (2013)
13. Wang, X., Tang, C., Zhao, X., Li, X., Jin, Z., Zheng, D., Zhao, T.: Transfer learning methods for spoken language understanding. In: ICMI. pp. 510–515 (2019)
14. Williams, J.D.: Web-style ranking and SLU combination for dialog state tracking. In: SIGDIAL. pp. 282–291 (2014)
15. Yang, X., Liu, J.: Using word confusion networks for slot filling in spoken language understanding. In: INTERSPEECH. pp. 1353–1357 (2015)
16. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: SLT (2014)
17. Zhao, Z., Zhu, S., Yu, K.: A hierarchical decoding model for spoken language understanding from unaligned data. In: ICASSP. pp. 7305–7309 (2019)
18. Zhu, S., Lan, O., Yu, K.: Robust spoken language understanding with unsupervised ASR-error adaptation. In: ICASSP. pp. 6179–6183 (2018)
19. Zhu, S., Yu, K.: Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In: ICASSP. pp. 5675–5679 (2017)
20. Zhu, S., Zhao, Z., Zhao, T., Zong, C., Yu, K.: CATSLU: The 1st chinese audio-textual spoken language understanding challenge. In: ICMI. pp. 521–525 (2019)