Adversarial Knowledge Distillation For Robust Spoken Language Understanding

Ye Wang1 Baishun Ling1 Yanmeng Wang1 Junhao Xue2 Shaojun Wang1 Jing Xiao1

1Ping An Technology, Beijing, China
2School of Computer Science and Engineering, Beihang University, Beijing, China
{wangye430,lingbaishun271,wangyanmeng219,wangshaojun851,xiaojing661}@pingan.com.cn
{justin_xue}@buaa.edu.cn

Abstract

In spoken dialog systems, Spoken Language Understanding (SLU) usually consists of two parts, Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU). In practice, such decoupled ASR/NLU design is beneficial for fast model iteration on both components. However, it also leads to the problem that NLU model suffers from the errors introduced by ASR, which degrades the overall performance. Improving the NLU model through Knowledge Distillation (KD) from large Pre-trained Language Models (PLMs) is proved to be effective and has drawn a lot of attention recently. In this work, we propose a novel Robust Adversarial Knowledge Distillation (RAKD) framework by introducing adversarial training into knowledge distillation to improve the robustness of NLU model to ASR-error. We conduct experiments on our own built classification dataset from a real-world spoken dialog system as well as existing datasets, where our proposed framework is proved to yield significant improvement over competitive baselines.

Index Terms: adversarial training, knowledge distillation, spoken language understanding, ASR-error robustness

1. Introduction

Nowadays the spoken dialog systems are widely applied in customer services of bank, insurance and e-commerce industries. The intelligent customer bot trained from historical recordings and manual scripts parses the user inputs and gives appropriate responses automatically. Compared with traditional human customer service, it can handle a large number of concurrent inquiries, with much lower cost and higher efficiency. The Spoken Language Understanding (SLU) plays an essential role in spoken dialog systems.

Typical SLU includes two sub-systems, ASR and NLU. ASR transforms voice signals to ASR hypotheses in text form and NLU provides intent classification and slot filling from the text. In practice, the ASR and NLU are cascaded for fast model iteration. They are both trained in a supervised manner with datasets containing speech recordings, manual transcriptions and annotations based on the transcriptions. The speech recordings and manual transcriptions are used to train ASR model, and manual transcriptions and annotations are exploited for improving the NLU performance.

However, in this pipeline structure, the upstream ASR error is propagated and even magnified into downstream NLU. NLU models trained and evaluated in offline environments have never seen ASR errors, while in actual deployment, the inputs to NLU such as ASR hypotheses or ASR outputs will inevitably contain ASR errors. Therefore, the actual performance of the NLU model will not reach the expected level.

Many efforts have been devoted to improve the robustness of NLU system by building ASR-robust representations. For example, Simonet et al. [1] proposed to expand the train data by simulating ASR errors. The lattices or Word Confusion Networks (WCN) of ASR module are also applied to improve the robustness of SLU models [2] [3] [4]. Huang et al. [5] proposed a pre-training method to generate ASR-error robust contextualized embeddings, which takes the word confusion networks generated by ASR model as extra features. However, when the ASR model is regularly updated, these ASR-robust representations also need to be updated.

Another direction is to design an end-to-end SLU system that interprets human voice directly into pre-defined intents without manual transcriptions [6] [7] [8]. However, this approach would not be practical. First, beyond intent classification, real-world SLU system normally consists of many other functions such as slot filling, domain predicting, sentiment predicting, etc. The end-to-end models require separated annotations of speech for all the above tasks, which is inefficient. Since the setting of semantic labels may be altered for different product releases or scenarios, the annotations from audio to the final task need to be conducted completely. Second, it is difficult for end-to-end system to utilize the large pre-trained language model (e.g. BERT [9] or ELECTRA [10]) and text-based data augmentation to promote NLU model’s performance.

Therefore, it still deserves studying how to improve the robustness of NLU model to upstream ASR errors in decoupled SLU. Recently large PLMs such as BERT [9], RoBerta [11] and Electra [10] have significantly promoted the performance in NLP domain. Since they have large parameters to store representations and are sufficiently trained by a huge dataset that contains many perturbations or noises, PLMs are also robust to the perturbed inputs [12]. Knowledge Distillation (KD) is widely applied to compress the large model [13] [14] [15]. It then brings an interesting question: is it feasible to promote the ASR-error robustness of NLU by applying KD from large PLMs.

As shown in our experiments (Section 4.4), we found that the conventional KD has limited performance in enhancing NLU model’s robustness to ASR-error. We introduce Adversarial Training (AT) [16] to KD to bridge this gap.

The main contributions of this paper are in three folds: 1) To improve ASR-error robustness of NLU model, we introduce a Robust Adversarial Knowledge Distillation (RAKD) framework, where AT is introduced into KD effectively. 2) To achieve such enhancement in RAKD, we propose a novel Adversarial knowledge distillation which calculates an additional Kullback-Leibler (KL) divergence between the perturbed inputs of teacher model and student model. 3) The proposed method is validated.
on one real-scene dataset and two public datasets, and outperforms baselines significantly.

2. Related works

KD is first proposed by Hinton et al. [16], which forces the small student model to mimic the output of the large teacher model by minimizing the KL divergence between the probability distributions on the classification logits of teacher model and the probability of student model.

AT is a common approach to enhance the robustness of neural networks, first proposed by Goodfellow et al. [16]. It augments the inputs with small perturbations to make the current model predictions away from the correct labels, then trains the model to be more robust to the perturbations. The perturbations are usually generated by random vectors. Takeru Miyato et al. [17] proposed Virtual Adversarial Training (VAT) by minimizing the KL divergence between the output of standard inputs and the output of perturbed inputs, while the vanilla AT utilizes the cross entropy between the output of perturbed inputs and the labels. We assume the cross entropy loss of perturbed inputs is still helpful in training, so we propose Joint Adversarial Training (JAT), which combines VAT with the cross entropy loss of perturbed inputs.

Recently, Ruan et al. [18] employed VAT and use the ASR hypothesis as the adversarial samples. The loss function includes a KL divergence term that penalizes the difference between predictive distribution from manual transcriptions and the distribution from ASR hypothesis. While in our work, we only utilize the manual transcriptions in training.

For the first time in the SLU systems, we propose a method to combine AT and KD to improve the robustness of NLU model. The framework proposed in this paper helps to improve the performance of SLU systems in practical applications.

3. Methodology

Intent classification and slot filling are two typical tasks of SLU, which predicts the speaker’s intents and extracts the corresponding entities from input utterances. We focus on intent classification task in this paper. The classification model consists of three parts: 1) Embedding layer which projects the input utterances to word embeddings, 2) Encoder module which generates the sentence embeddings, 3) Classify layer which is a linear layer to transform sentence embeddings to the logits for intents classification. The probabilities of intents are calculated by a softmax function after the logits.

The proposed framework is illustrated in Figure 1. We use \( x \) and \( x_t \) to represent the word embeddings of input utterances in student model and teacher model, respectively. \( y_t \) indicates the corresponding intent labels, \( r \) denotes the perturbations generated by AT, and \( x' \) is the perturbed embeddings based on \( x \). \( z_s(x) \) and \( z_t(x_t) \) represent the outputs of student model and teacher model, respectively, which are the network logits of different intents before the softmax layer.

Our goal is to predict the correct intents based on input utterances, which are manual transcriptions in training, and are ASR hypotheses in testing. In Figure 1, we use an LSTM model with ELMo [19] pre-trained embeddings as the student model followed previous work [5], and ELECTRA[10] with a classify layer as the teacher model, which has been proved to be a superior pre-trained LM than BERT[9] and RoBerta[11]. For the datasets in which every sample only has one intent, the classify layer is a softmax layer. For multi-intent dataset, we apply multi binary classify layers which apply the sigmoid function.

3.1. Joint adversarial training

The loss of VAT [17] can be formulated as:

\[
L_{VAT}(x) = D(p(y|x), p(y|x + r))
\]

\[
r = \arg \max_{|r| \leq \epsilon} D(p(y|x), p(y|x + l))
\]

where \( p(y|x) \) represents the output probability calculated by the softmax layer, \( r \) is the perturbations added to \( x \), \( \epsilon \) denotes the range of perturbations. \( l \) is initialized by a randomly sampled vector \( \delta \), \( \xi \) defines the step size of \( l \). \( D \) is a non-negative function that represents the distance between the original distributions and the perturbed distributions, which can be KL divergence or cross entropy. Generally, the \( r \) can be approximately calculated by:

\[
r \approx \epsilon \frac{g}{||g||}
\]

\[
g = \nabla_l D(p(y|x), p(y|x + l))
\]

where \( g \) is the gradient of distance calculation. In our approach, \( D \) is set to KL divergence, and we set \( x' = x + r \).

However, VAT ignores the cross entropy between the output of perturbed inputs and the labels, which is also helpful to guide the model in classification task. So we propose the Joint Adversarial Training (JAT) by adding the cross-entropy loss to VAT which can be formulated as:

\[
L_{JAT}(x, x', y_t) = D_{CE}(h(y_t), p(y|x'))
+ D_{KL}(p(y|x), p(y|x'))
\]

where \( h(y_t) \) represents the one-hot representations of train data labels. \( D_{CE} \) and \( D_{KL} \) represent the cross-entropy loss and the KL divergence respectively.

3.2. Adversarial knowledge distillation

KD is a common approach to improve the performance of the student model. The loss of KD can be summarized as:

\[
L_{KD}(x, x_t) = \mu^2 D_{KL}(\sigma(z_s(x)/\nu), \sigma(z_t(x)/\nu))
\]

\[
L = \lambda L_{CE} + (1 - \lambda)L_{KD}
\]
where $z_t$ and $z_s$ are the logits vectors of teacher model and student model. $\sigma$ indicates the softmax function. $\nu$ and $\mu$ are the hyperparameters for softening the predictions. We use different values of $\nu$ and $\mu$ to adjust the value range of $L_{CE}$, which are set to be equal to the values in the original paper[20]. $L_{CE}$ denotes the cross-entropy loss of the classification task, and $\lambda$ controls the contributions of cross-entropy and KD loss.

Transferring the robust capability from teacher model to student model is essential in our framework. As shown in section 4.4, original KD is not effective in improving the ASR-error robustness of student model. To address this problem, we add a regularization term to the original KD which denotes the distance between the probability distributions of perturbed inputs from teacher model and the perturbed probabilities of student model.

First, we proposed the adversarial KD which can be formulated as follows:

$$L_{adv-KD}(x, x', x_t, x_t') = \mu^2 \left[ D_{JS}(\sigma(z_t(x)), \sigma(z_t(x'))) + D_{JS}(\sigma(z_s(x)), \sigma(z_s(x'))) \right]$$

where $x'$ is the perturbed inputs of student model in adversarial training, and $x_t'$ is the perturbed inputs of teacher model. The first part of $L_{adv-KD}$ is the KD loss with Jenson’s Shannon(JS) divergence, which actually is symmetric KL divergence, and the second part learns the robustness of teacher model by applying adversarial learning.

Note that the corresponding perturbed inputs of teacher model $x_t'$ are hard to be calculated. Because when the word embedding layers of teacher model and student model are different, for example, the teacher model uses ELECTRA and the student model uses ELMo, the inputs of teacher model $x_t$ and the perturbations $r$ generated by AT from student model are always mismatched. We adopt a teacher model trained by JAT to solve this problem, assuming that the output probabilities of perturbed inputs should be close to the output probabilities of standard inputs when the model is trained with JAT, which can be denoted as $z_t^\text{adv}(x_t') \approx z_t^\text{adv}(x_t)$.

Then we combine the adversarial KD with JAT for further improvement. The total loss $L_{RAKD}$ of our proposed framework can be formulated as:

$$L_{RAKD} = \alpha L_{CE} + \beta L_{JAT} + \gamma L_{adv-KD}$$

$\alpha$, $\beta$, $\gamma$ are the hyperparameters for different terms. $L_{CE}$ is the cross entropy loss of student model for the classification task.

4. Experiment Study

4.1. Datasets

In order to verify the effectiveness of our proposed framework, we evaluate it on three SLU datasets:

- **Snips** [21] is a classification dataset for benchmarking NLU systems. We use the expanded version which is provided by Huang et al. [5] by translating the manual transcript in Snips to audio then to ASR hypotheses in the order with TTS\(^1\) and ASR\(^2\). The WER of the ASR hypotheses is very high since the low accuracy of TTS and ASR models, which increases the difficulty in intents prediction.

- **Snips** contains 59 intents, which is manually scripts written by human for training and ASR hypotheses.

- **ATIS** [22] is a dataset consisting of audio recordings and corresponding manual transcriptions about humans asking for flight information on automated airline travel inquiry systems. In order to make a noisy copy of ATIS, we exploit the method of MS-SNSD [23] to perform data augmentation by introducing noise with a signal-to-noise ratio of 20dB, then translate the noisy audio data into ASR hypotheses by online ASR service\(^3\).

We also build an in-house Chinese spoken dialog dataset which is collected from a real-world online spoken dialog system for the telephone custom services in banking, named Chinese Spoken dialog of Bank Services (CSDBS). It consists of manual scripts written by human for training and ASR hypotheses collected from the online ASR system for testing. CSDBS is a multi-intent classification dataset where each sample is labeled with one or more intents. It contains 59 intents, which is more than other existing open spoken dialog datasets.

The dataset statistics are shown in Table 1. We use test to denote the test dataset of clean manual transcriptions, ASR-test means the test dataset of ASR hypotheses.

4.2. Baselines

We compare our framework with four baselines:

- **LSTM\(_{ELMO}$$** the NLU model applied in Huang et al. [5], which is an LSTM model with the ELMO\([19]\) embeddings.

- **AT and VAT**: We compare our method with the traditional robust training methods, AT [16] and VAT [17].

- **KD**: original KD method [20] by using LSTM\(_{ELMO}$$ as the student model, large PLMs as the teacher model.

4.3. Implementation Details

In our approach, we first train the teacher model with manual transcript, then train the student model by KD with manual transcriptions as well. The hyperparameters are chosen according to the model performance on the validation set.

For Snips and ATIS datasets, we use a 2-layers LSTM model with ELMO [19] pre-trained embeddings as the student model. The hidden size of LSTM model is 300. We adopt the Electra-base model as our teacher model, which has 12-layers and the hidden size is 768. The LSTM model is trained with Adam[24] optimizer with an initial learning rate $3 \times 10^{-4}$. The batch size and maximum epoch number are set to 64 and 10 respectively. In addition, we set the parameters that control the size of the perturbations $\epsilon = 5.0$ and $\xi = 10^{-4}$. The $\nu$ and $\mu$ are set to 5 and 10. The hyperparameters $\alpha$, $\beta$, and $\gamma$ of RAKD are all set to 1.0.

For CSDBS dataset, we use the official released Chinese BERT-base model\(^4\) as the teacher model, which has 12-layers.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Test</th>
<th>ASR-test</th>
<th>Intents</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDBS</td>
<td>28794</td>
<td>1655</td>
<td>1488</td>
<td>59</td>
<td>0.2611</td>
</tr>
<tr>
<td>Snips</td>
<td>13084</td>
<td>700</td>
<td>700</td>
<td>7</td>
<td>0.4199</td>
</tr>
<tr>
<td>ATIS</td>
<td>4978</td>
<td>893</td>
<td>893</td>
<td>26</td>
<td>0.1687</td>
</tr>
</tbody>
</table>

\(^1\)https://cloud.google.com/text-to-speech
\(^2\)https://kaldi-asr.org/models/m1
\(^3\)https://ai.baidu.com/tech/speech/asr
\(^4\)https://github.com/google-research/bert
Table 2: Accuracy on ATIS and Snips test set (%). Manual and ASR indicates evaluating on manual and ASR scripts respectively. LSTM+RAKD outperforms other models significantly (t-test of 10 runs, p-values < 0.01).

<table>
<thead>
<tr>
<th>Model</th>
<th>ATIS</th>
<th>Snips</th>
<th>Pre-train on ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELECTRA base</td>
<td>89.24</td>
<td>97.42</td>
<td>81.71</td>
</tr>
<tr>
<td>ELECTRA base + JAT (teacher)</td>
<td>90.25</td>
<td>97.53</td>
<td>83.71</td>
</tr>
<tr>
<td>LSTM+ELMO (student) [5]</td>
<td>84.24</td>
<td>95.51</td>
<td>78.59</td>
</tr>
<tr>
<td>LSTM+ELMO + AT</td>
<td>89.90</td>
<td>95.31</td>
<td>80.25</td>
</tr>
<tr>
<td>LSTM+ELMO + VAT</td>
<td>88.88</td>
<td>95.20</td>
<td>80.84</td>
</tr>
<tr>
<td>LSTM+ELMO + KD</td>
<td>84.19</td>
<td>95.48</td>
<td>79.70</td>
</tr>
<tr>
<td>LSTM+ELMO + KD + VAT</td>
<td>88.93</td>
<td>97.13</td>
<td>81.46</td>
</tr>
<tr>
<td>LSTM+ELMO + RAKD w/o LadvKD</td>
<td>89.80</td>
<td>96.93</td>
<td>82.85</td>
</tr>
<tr>
<td>LSTM+ELMO + RAKD w/o LJAT</td>
<td>89.07</td>
<td>95.76</td>
<td>80.94</td>
</tr>
<tr>
<td></td>
<td>89.24</td>
<td>95.74</td>
<td>82.26</td>
</tr>
</tbody>
</table>

with 768 hidden size. We pre-trained a transformer encoder model named as BERTtiny as the student model, which has 2-layers with 128 hidden size, with the same dataset for Chinese BERT model. The learning rate for fine-tuning BERTtiny is set to 5e-5, with batch size 128 and maximum epoch 20. The perturbations ε and ξ are set to 1.0 and 10^-5, respectively. Other hyperparameters are the same as the training on Snips and ATIS.

4.4. Results and analysis

Following the experiment setting in [5], we conduct our experiments on two cases for Snips dataset: 1) Train on Manual, which fine-tunes the pre-trained models on classification task with manual transcriptions, 2) Pre-train on ASR, which means that first fine-tunes the ELMo or Electra on ASR hypotheses by the PLM task, then trains the model on classification task with manual transcriptions. For ATIS and CSDBS, we only conduct experiments on case 1).

Table 2 provides the accuracy scores obtained from our proposed framework and the baseline models on ATIS and Snips dataset. We report the average result over 10 runs on the ASR-test set. In addition, a significance test (independent two-sample t-test) was performed to compare our model with other comparison models. Our proposed model LSTM+ELMO+RAKD significantly outperforms other models (t-test of 10 runs, p-values < 0.01). RAKD improves the accuracy by 5.42%, 6.60% compared to the original LSTM+ELMO on the case of Train on Manual on Snips and ATIS dataset, respectively. In case of Pre-train on ASR, our method improves the accuracy by 2.77% on Snips. Moreover, our RAKD outperforms the best baseline models by 1.04%, 2.48%, 1.32% in all cases, respectively.

Notice that KD method obtains limited improvement on NLU model both on ATIS and Snips, which illustrates the shortage of KD on transfer learning to ASR-error robustness. While the joint learning of KD and VAT gains little improvement on ASR-test set compared to the best baseline model, our proposed framework achieves an improvement of 1.90% EM ratio and a reduction of 2.93% hamming-loss over the base student model, even surpassing the performance of teacher model.

Moreover, we perform the ablation study by removing modules proposed in our framework: 1) Removing the regularization term LadvKD 2) Removing the adversarial training term LJAT. As shown in the last three rows of Table 2, the results of ASR-test are dropped in all ablation cases, which proved the effectiveness of our proposed framework. Besides, the results of removing LadvKD which actually is training with JAT proved the proposed JAT is superior than VAT and AT methods.

5. Conclusion

In this paper, we addressed the model robustness issue in a cascade SLU system, where the robustness of downstream NLU to ASR errors affects the overall performance of SLU. Therefore, we propose a novel RAKD framework for improving the ASR-error robustness of the NLU model by only training on manual transcriptions. Our framework achieves significant improvements on all datasets compared to baseline models.

6. Acknowledgements

This work was partially supported by the Ministry of Science and Technology of the People’s Republic of China under Grant 2020AAA0104200.
7. References


