Robust Spoken Language Understanding with Unsupervised ASR-error Adaptation

Su Zhu, Ouyu Lan, Kai Yu
SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{paul2204,blue-0-0-,kai.yu}@sjtu.edu.cn

Summary

- **Motivation:** Speech Recognition Errors & Robustness
  - to improve robustness of SLU (Spoken Language Understanding).
- **Our approach:** Parameter Partial-Sharing BLSTMs & Unsupervised ASR-error Adaptation
  - Only speech recognized text is used for adaptation. There is no SLU annotation on the speech recognized text.
- **Result:**
  - Our method improves the robustness of SLU significantly.
  - No need of SLU annotation on the speech recognized text.

1. Introduction

- **Slot tagging task of SLU**

<table>
<thead>
<tr>
<th>Input: words</th>
<th>show</th>
<th>flights</th>
<th>from</th>
<th>Boston</th>
<th>to</th>
<th>New</th>
<th>York</th>
<th>today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: slots</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>B-ToCity</td>
<td>O</td>
<td>B-ToCity</td>
<td>I-ToCity</td>
<td>B-DepartDate</td>
</tr>
</tbody>
</table>

SDS: Spoken Dialogue System; ASR: Automatic Speech Recognition

- **Robustness of SLU to ASR-error**
  - Inputs of SLU (e.g. slot tagging):
    1. Manual transcription (Oracle)
    2. ASR output (May contain errors)
  - Target of SLU (e.g. slot tagging):
    - Human annotation based on the inputs of SLU. It aims to investigate SLU module independently.
- **Traditional Methods: Prepare Training Data**
  - Human annotation on the manual transcription. ✗ (Our method)
  - Human annotation on the ASR output. ✗
  - What if the ASR system changes? (i.e. If ASR output is changed, we need to renew the semantic annotation.) ✗ labor-intensive & time-consuming
  - Unlabeled ASR output for adaptation. ✓ (Our method)

2. Unsupervised ASR-error Adaptation

- **Types of Data samples:**
  1. tag: manual transcription & semantic annotation of slot-tags;
  2. tseq: manual transcription
  3. hyp: ASR output (1-best, unlabeled)

- **Model Architecture**

```
shared
BLSTM
BLSTM
BLSTM
BLSTM
BLSTM

Labelled transcript
Reconstruction

Transcript
Reconstruction

ASR hyp.
```

- **Input reconstruction (L_rec):**
  - Word to word (W2W): \( p(x|x) = \sum p(x_i|x) \) — a naive try
  - Sequence to sequence (S2S): \( p(x|x) = \sum p(x_i|x_{0:i-1}) \)
  - Bidirectional language model (BLM): \( p(x|x) = \sum p(x_{i+1}|x_i) + \sum p(x_{i-1}|x_{i+1}) \)
  - Adversarial task classification loss \( L_{adv} \)
  - We use random prediction training (Kim, 2017) to force the shared encoder more task-invariant, i.e. the label of task classifier is randomly set to task 1/2/3 with equal probability.

3. Experiments

- **Dataset:** collected from a Chinese commercial SLD.
  - Domain: car navigation (13 different slots).

<table>
<thead>
<tr>
<th>System</th>
<th>Reconstruction</th>
<th>F1-score on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle1</td>
<td>---</td>
<td>84.85</td>
</tr>
<tr>
<td>Oracle2</td>
<td>---</td>
<td>85.64</td>
</tr>
<tr>
<td>Baseline1</td>
<td>---</td>
<td>81.90</td>
</tr>
<tr>
<td>Baseline2</td>
<td>---</td>
<td>78.71</td>
</tr>
</tbody>
</table>

**Character Error Rate**

- **Systems:**
  - Oracle1: It is trained on the data of ASR output with SLU annotation.
  - Oracle2: It is trained on the data of both manual transcription and ASR output with SLU annotation.
  - Baseline1: It is trained on manual transcription with SLU annotation.
  - Baseline2: It is trained on ASR output with SLU annotation. (word alignment between transcript and ASR 1-best)
  - Domain adaptation (Kim, 2017): \( BLSTM_{ggene} = BLSTM_{gaug} \)
  - \( L_{tag} + L_{rec} : \) Training driven by slot tagging and reconstruction.
  - \( L_{tag} + L_{rec} + L_{adv} \): Additional adversarial task classification.

**Results of slot tagging on ASR output and manual transcriptions.**

- **Character Error Rate**

**Reconstruction networks are separated (not shared).**

- **Models:**
  - Bidirectional language model is a good choice for input reconstruction.
  - Adv. performs better (but not significantly).
  - Our method becomes very close to the upper bound.
  - We need more data to verify our method.

---