Better Error Detection with Calibrated Neural Confidence Modeling

Ammar and Trey
(plus our awesome mentor Sina)
Motivation

User utterance: “Are there museums downtown?”

ThingTalk output:
now => (@multiwoz.Attraction()), (area == enum(centre) && type =~ "museum") => notify;
Motivation

User utterance: “Tell me about some parks.”

ThingTalk Semantic Parser

ThingTalk output:
now => (@multiwoz.Attraction()), (type =~ "museum") => notify;
Motivation

User utterance: “Tell me about some parks.”

ThingTalk output:
now => (@multiwoz.Attraction()), (type =~ "museum") => notify;

ThingTalk Semantic Parser

Confidence: 20%
Likely error!
Motivation

User utterance: “Tell me about some parks.”

Almond output: “I’m sorry; I didn’t understand that.”

Confidence: 20%
Likely error!
Calibration = Confidence vs Accuracy
Baseline

Baseline method: Simply use the semantic parser’s softmax output probability as the confidence
Baseline: not well calibrated!

Baseline method: Simply use the semantic parser’s softmax output probability as the confidence
Baseline: not well calibrated!

Expected Calibration Error (ECE): 0.19
Baseline: not well calibrated!

Expected Calibration Error (ECE): 0.19

bars below the line: model is **overconfident**

70% of outputs have 100% confidence (but only 80% of them are correct)
How to Calibrate a Model?

**ThingTalk Semantic Parser**

**Input:** user utterance

**Output:** ThingTalk code

**Calibrator Model (Random Forest)**

**Input:** original model state evaluated on an input

**Output:** confidence score (probability that the model’s output is correct)
Calibrator methodology: Training

Dataset: Annotated MultiWOZ

- Training Data (80% of original training set)
- Calibration Data (20%)
- Evaluation Data (not to scale)

Semantic Parser
Calibrator methodology: Calibration

Dataset: Annotated MultiWOZ

- Training Data
- Calibration Data
- Evaluation Data

Trained Semantic Parser

Calibration Features
(Top K beam search softmax outputs, MC Dropout variance)

Calibrator Model
(Gradient-boosted random forest)
Calibrator methodology: Evaluation

Dataset: Annotated MultiWOZ

- Training Data
- Calibration Data
- Evaluation Data

Trained Semantic Parser

Calibration Features

Confidence predictions

Trained Calibrator Model
Results
## Experiments

<table>
<thead>
<tr>
<th>Calibrator features</th>
<th>ECE</th>
<th>Best F1</th>
<th>Coverage @ best F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.19</td>
<td>0.87</td>
<td>77%</td>
</tr>
<tr>
<td>1 beam</td>
<td>0.04</td>
<td>0.86</td>
<td>71%</td>
</tr>
<tr>
<td>2 beams</td>
<td>0.04</td>
<td>0.85</td>
<td>85%</td>
</tr>
<tr>
<td>1 beam + MC Dropout</td>
<td>0.04</td>
<td>0.86</td>
<td>76%</td>
</tr>
<tr>
<td>2 beams + MC Dropout</td>
<td><strong>0.03</strong></td>
<td>0.86</td>
<td>82%</td>
</tr>
<tr>
<td>4 beams + MC Dropout</td>
<td>0.06</td>
<td>0.85</td>
<td>78%</td>
</tr>
</tbody>
</table>
Summary: better calibration

Baseline

ECE: 0.19

Top Calibrator Model (2 beams + dropout)

ECE: 0.04
Summary: better dispersion

Baseline

Top Calibrator Model
(2 beams + dropout)
Summary: better performance

The top calibrator model (2 beams + dropout) gets the same best F1 score (0.87) with better coverage (82% of inputs vs. 77%)
Further work

- Train calibrator on more data
  - can calibrator precision improve at high confidence thresholds with more data?
- Dropout in all layers
  - reproduce the results using same theoretical guarantees
- Error analysis
  - better calibration allows us to perform more nuanced error analysis: which high-confidence outputs are incorrect? what kinds of inputs lead to low-confidence outputs?
- Uncertainty interpretation: reproduce further results from Dong et. al of retrieving token-level uncertainty through dropout backpropagation


