# Better Error Detection with Calibrated Neural Confidence Modeling

Ammar and Trey
(plus our awesome mentor Sina)

# Motivation

User utterance: “Are there museums downtown?”

**ThingTalk Semantic Parser**

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 Semantic Parser**

Confidence: 20%
Likely error!

ThingTalk output:

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

# Motivation

User utterance: "Tell me about some parks."

**ThingTalk Semantic Parser**

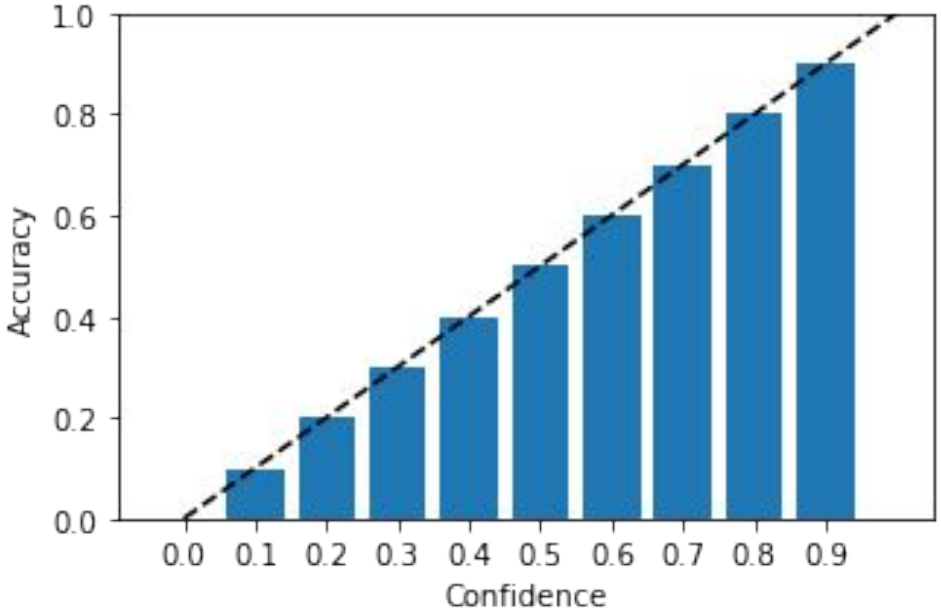
Confidence: 20%

Likely error!

Almond output:

`"I'm sorry; I didn't understand that."`

# Calibration = Confidence vs Accuracy

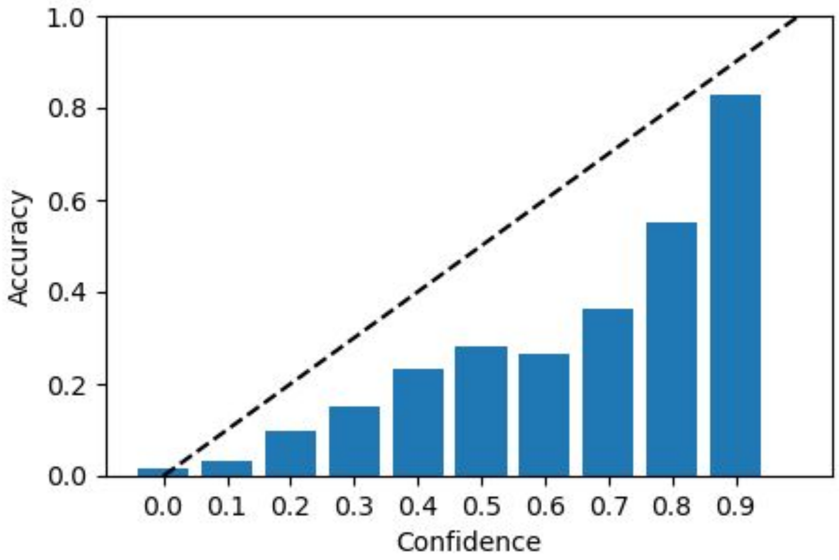


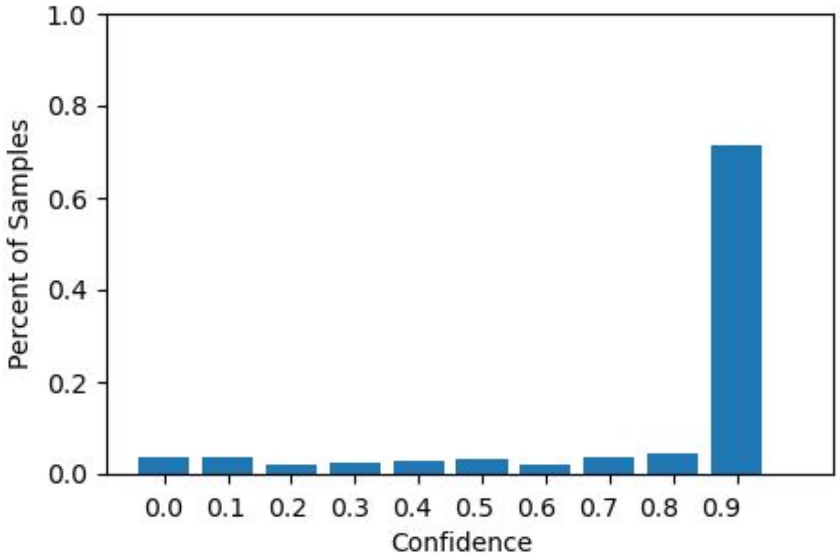
# Baseline

Baseline method: Simply use the semantic parser’s softmax output probability as the confidence

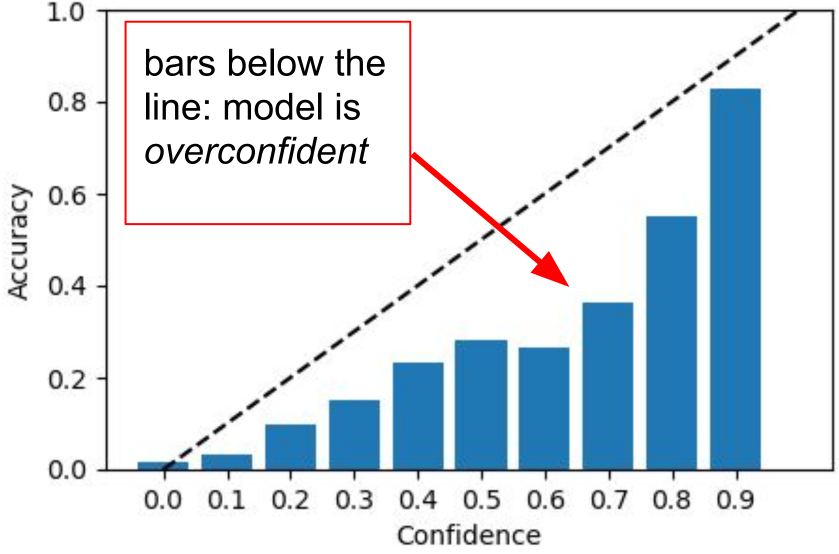
# Baseline: not well calibrated!

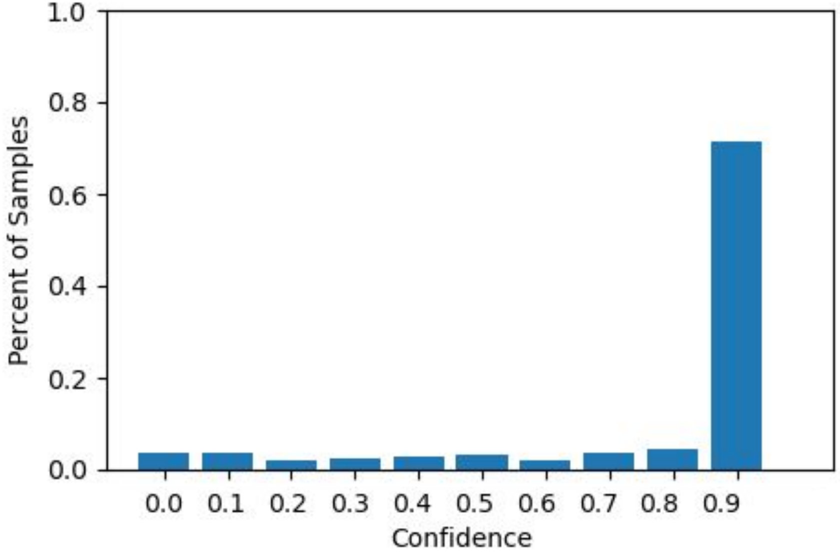
Baseline method: Simply use the semantic parser’s softmax output probability as the confidence



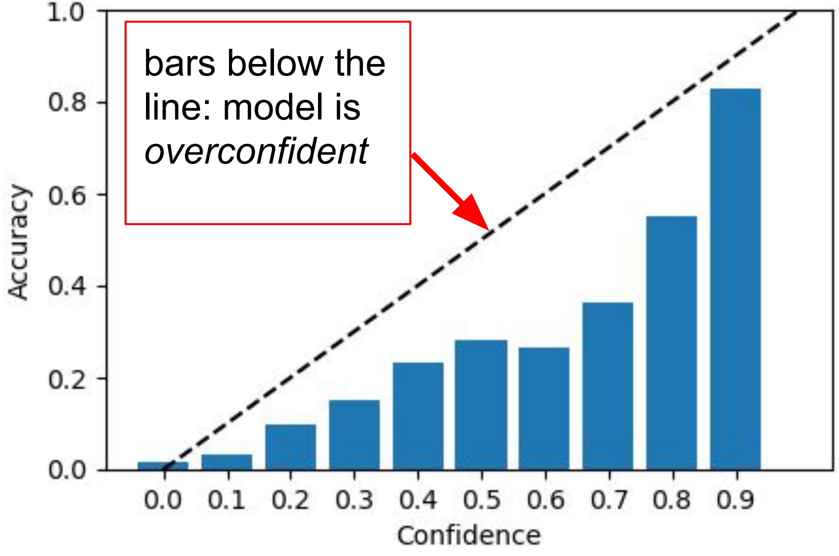
# Baseline: not well calibrated!

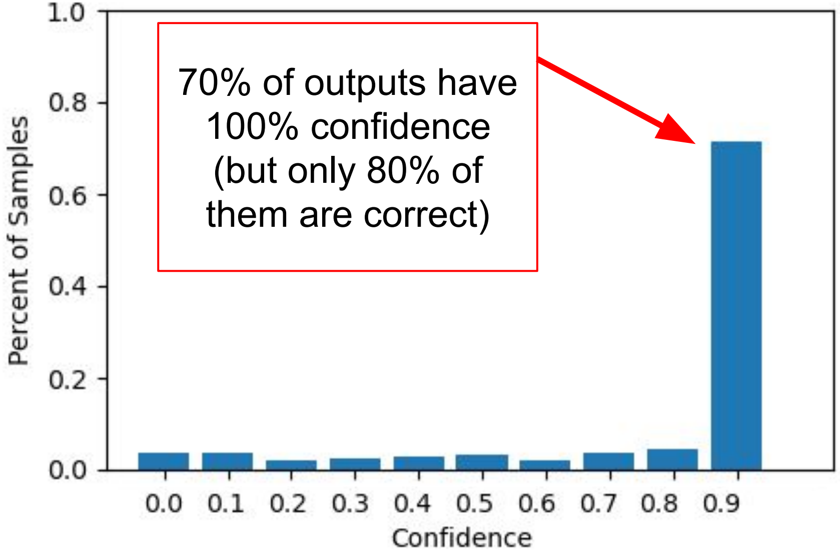



Expected Calibration Error (ECE): **0.19**

# Baseline: not well calibrated!



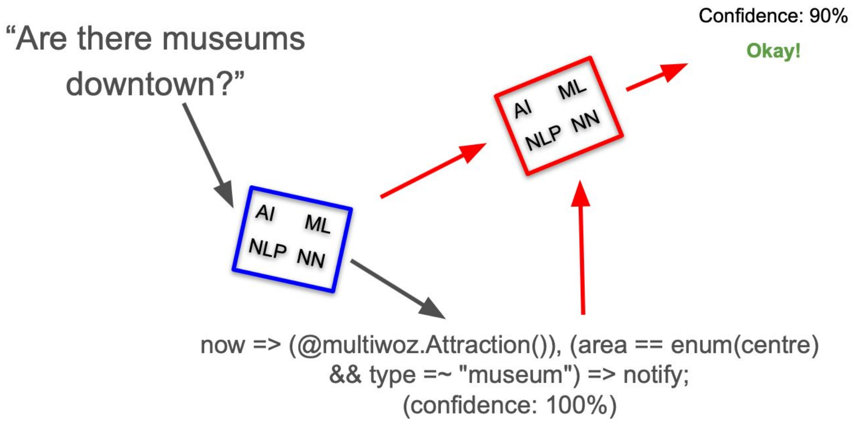

Expected Calibration Error (ECE): **0.19**

# How to Calibrate a Model?

ThingTalk Semantic Parser

**Input**: user utterance

**Output**: ThingTalk code



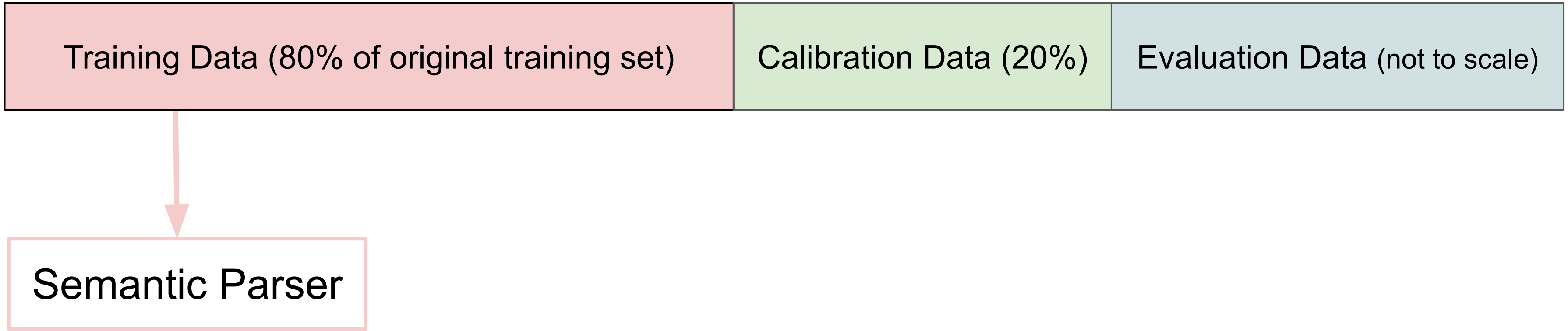

Calibrator Model (Random Forest)

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

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

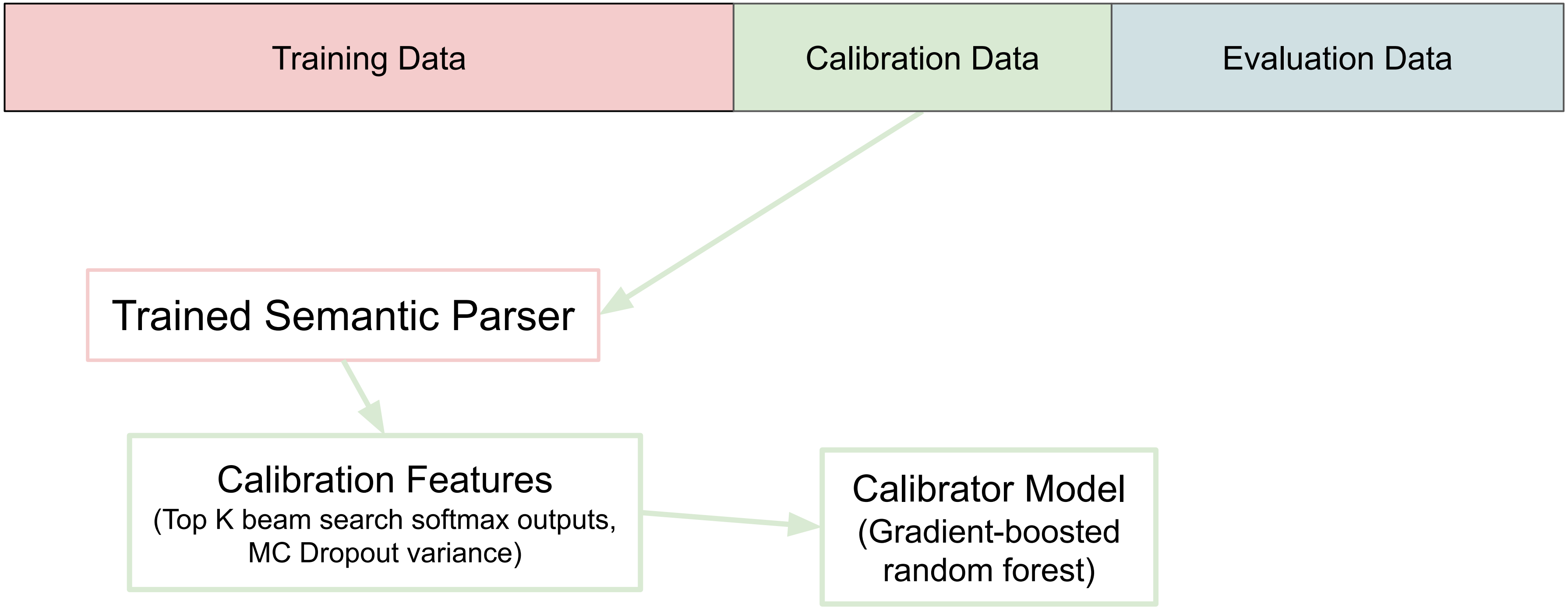
# Calibrator methodology: Training

**Dataset: Annotated MultiWOZ**



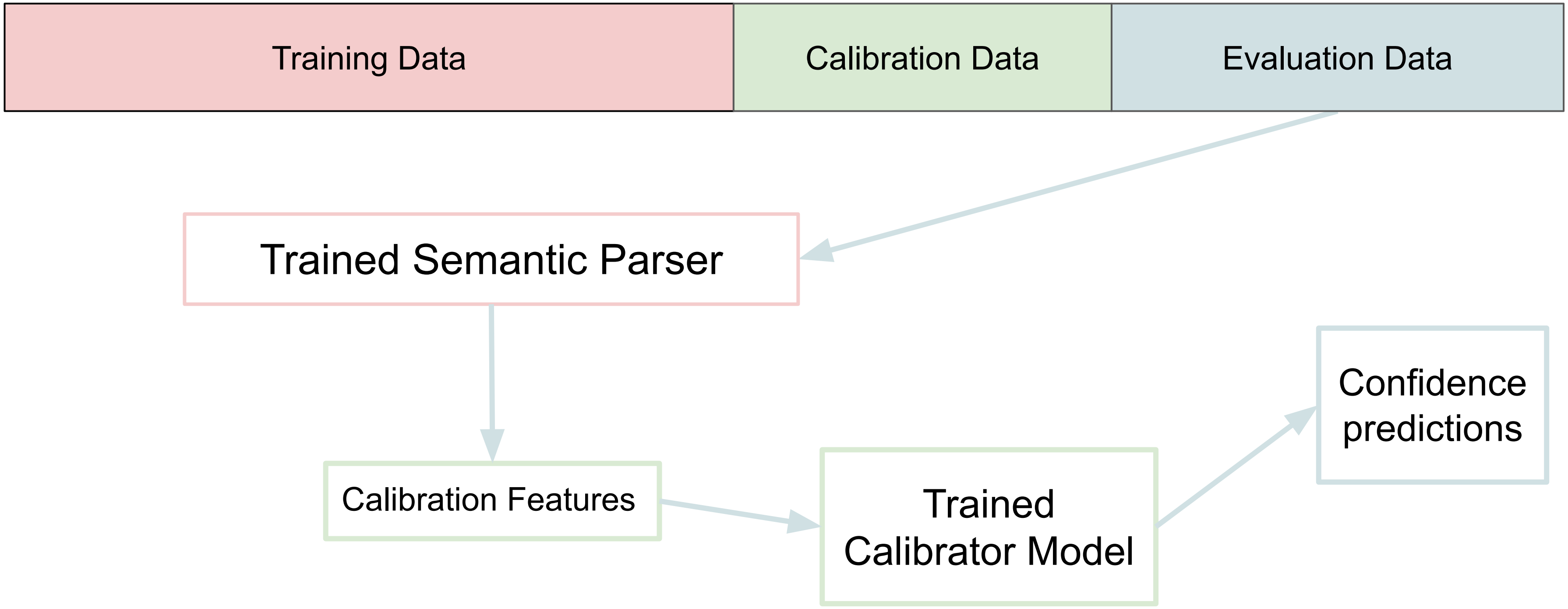
# Calibrator methodology: Calibration

**Dataset: Annotated MultiWOZ**



# Calibrator methodology: Evaluation

**Dataset: Annotated MultiWOZ**



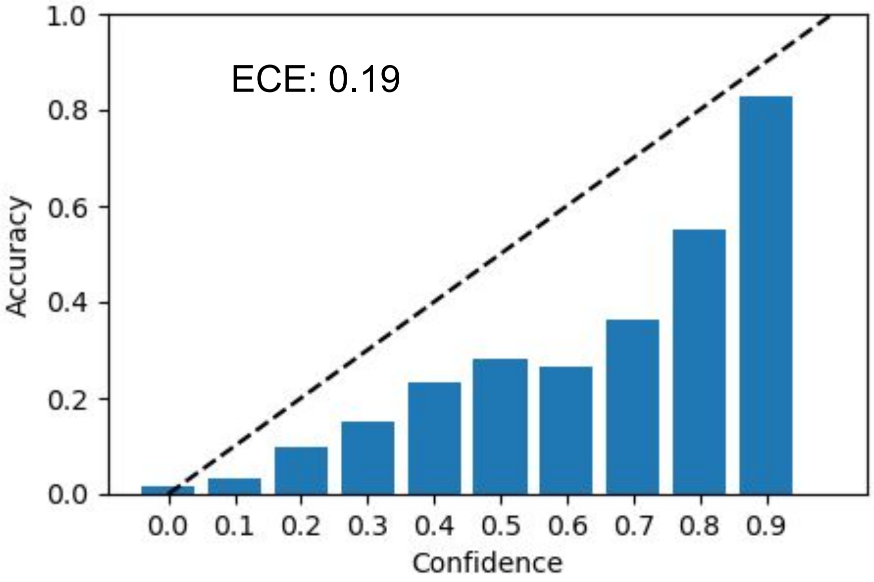
# Results

# Experiments

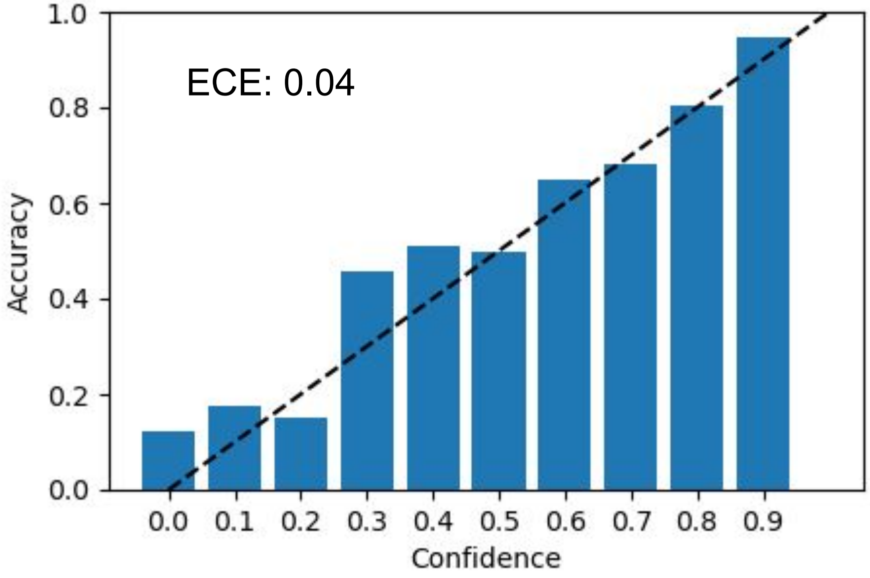
| Calibrator features | ECE | Best F1 | Coverage @ best F1 |
|---|---|---|---|
| Baseline | 0.19 | **0.87** | 77% |
| 1 beam | 0.04 | 0.86 | 71% |
| 2 beams | 0.04 | 0.85 | **85%** |
| 1 beam + MC Dropout | 0.04 | 0.86 | 76% |
| 2 beams + MC Dropout | **0.03** | 0.86 | 82% |
| 4 beams + MC Dropout | 0.06 | 0.85 | 78% |

# Summary: better calibration

Baseline

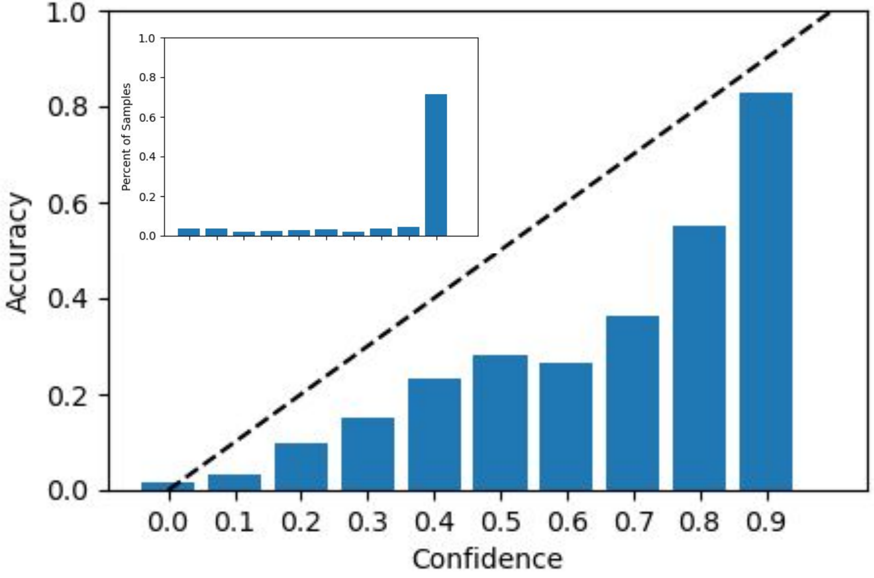



Top Calibrator Model
(2 beams + dropout)

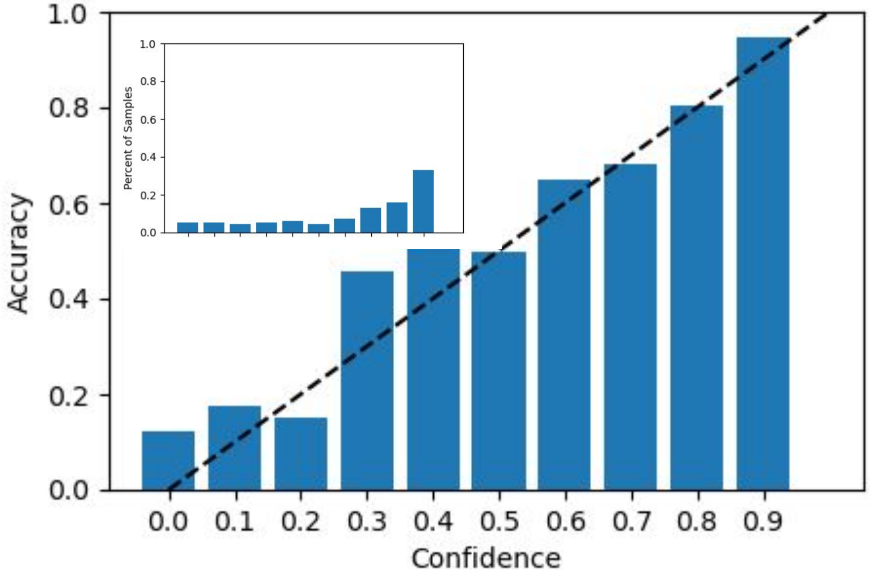


# Summary: better dispersion

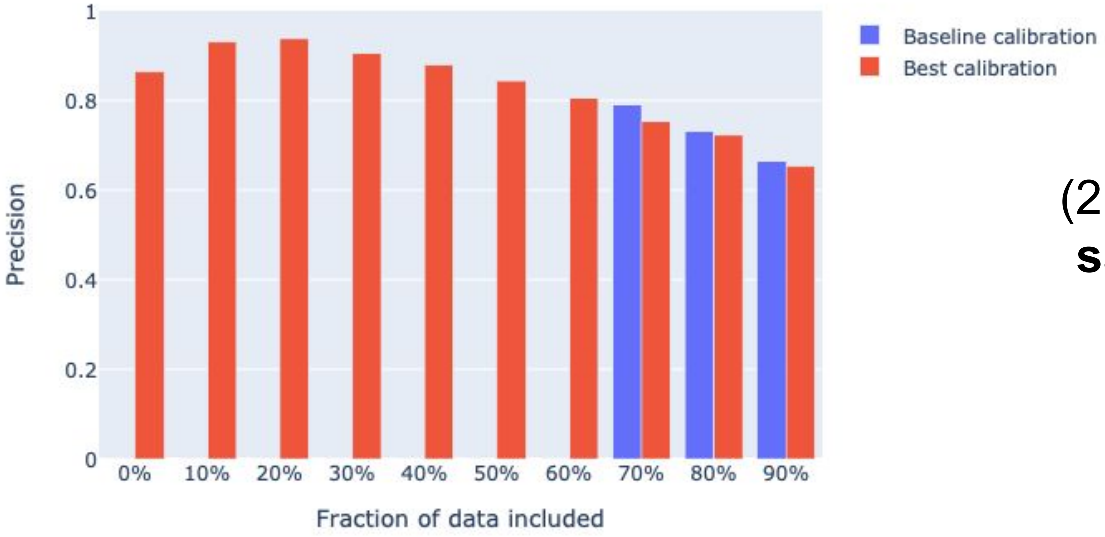
Baseline




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

# References


Rodriguez, P., Feng, S., Iyyer, M., He, H., & Boyd-Graber, J. 2019. Quizbowl: The case for incremental question answering. *arXiv preprint arXiv:1904.04792*.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. 2017. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*

Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *arXiv:1506.02142*

Kamath, A.; Jia, R.; and Liang, P. 2020. Question Answering under Domain Shift. *arXiv preprint arXiv:2006.09462*.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. arXiv:1805.04604

Kendall, Alex and Gal, Yarin. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*.