Determining Question-Answer Plausibility in Crowdsourced Datasets Using Multi-Task Learning

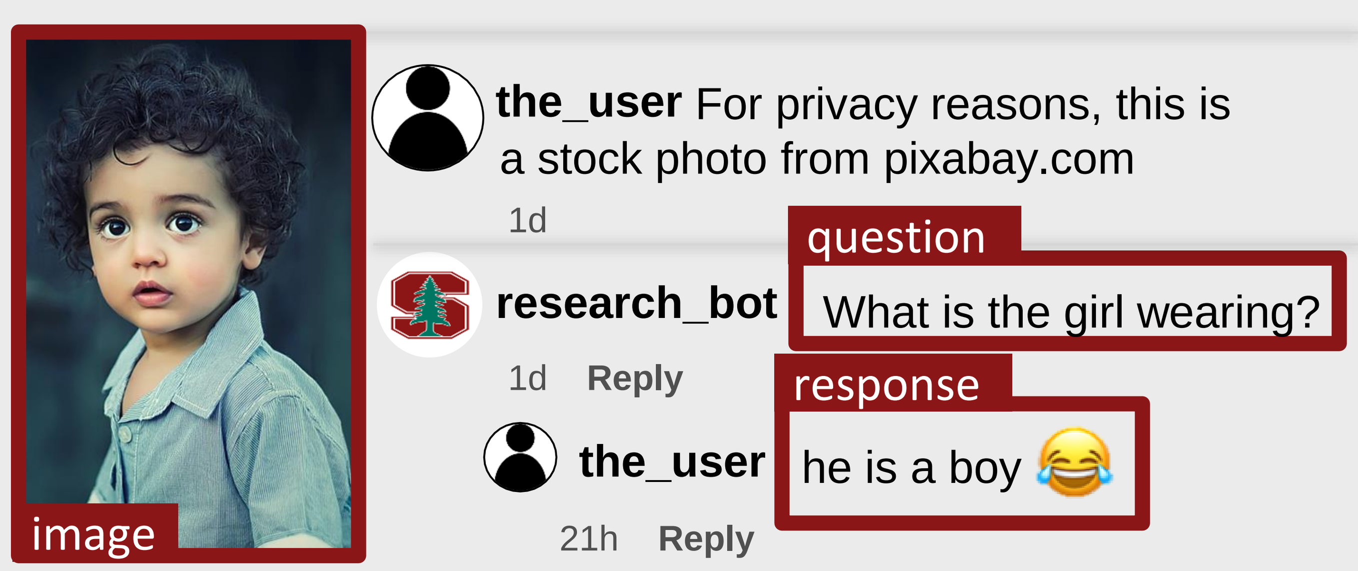
Rachel Gardner, Maya Varma, Clare Zhu, Ranjay Krishna
Department of Computer Science, Stanford University

Motivation

- Large datasets are often labeled by paid crowdworkers, who…
  - are expensive at scale
  - lack context
  - can be inaccurate
  - may take weeks to finish the task
- **Goal:** create a Q+A dataset from noisy text such as social media using primarily automatic methods

VQA Case Study

- We generate a Visual Question Answering (VQA) dataset as an example of our proposed task
- All data preprocessing and model code is available (github.com/rachel-1/qa_plausibility)
- 50k image-question-response trios, obtained from users on social media
  - Questions asked by bot that analyzed image
  - 7.2k examples labeled by Amazon Mechanical Turk workers (though for privacy reasons, the dataset itself cannot be released at this time)



- Manual analysis of ~5% of the labeled data showed that only a handful of examples required the image to determine question/answer plausibility (and all of which were "where" questions which were excluded from the dataset)

Proposed Task

Question-Answer Plausibility

Given a (possibly invalid) question and a user response, a model must:

1. Determine if the question is plausible (i.e. relevant and answerable)
2. Determine if the response is plausible (i.e. a reasonable answer)
3. Extract the specific answer from the free-form response

Why plausibility and not accuracy?

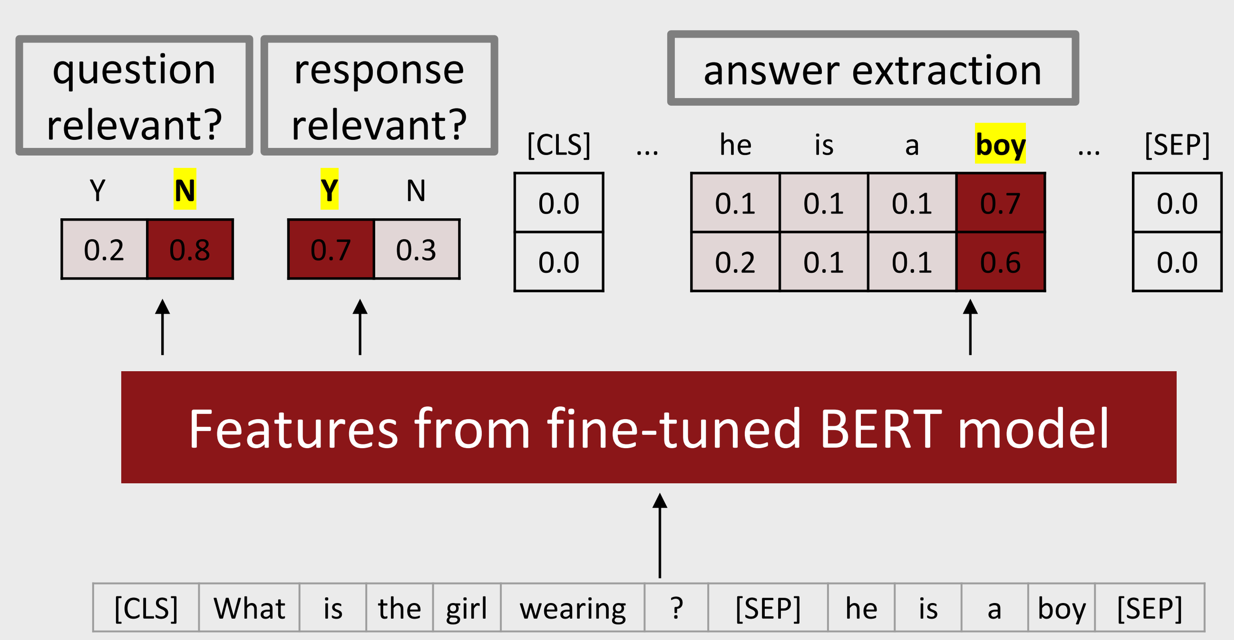
- determining accuracy directly requires the very domain knowledge we are trying to learn, but plausibility can be determined more easily
- in practice users very rarely provide plausible but incorrect answers

Illustrative Examples (VQA Case Study)

#	[Plausible?] Question	[Plausible?] Response	%
1.	[✓] What is on the table?	[✓] beet and carrot juice ☺ ☺	51
2.	[✓] What is the person doing?	[✗] not much lol	22
3.	[✗] What is the hamster doing?	[✗] that is not a hamster	15
4.	[✗] What is on top of the cake?	[✓] that is not cake that's chicken	11

Model Architecture

We designed a multi-task BERT model to jointly perform the three tasks:



- The trunk of a pre-trained BERT model was finetuned on our dataset
- Each of the three outputs is computed simultaneously, but if the response is not relevant, the answer extracted is ignored

For question and response relevance, the model predicts a single score, while for extracting an answer, the model scores each token with its probability of being the start or end token.

Results and Analysis

We evaluated the model architecture when trained on different groupings of the tasks and found the top performing system to be a combination of a question plausibility model and a model which both determines response plausibility and extracts the answer.

QP = Question Plausibility
RP = Response Plausibility
AE = Answer Extraction

Combined Task	QP AUROC	RP AUROC	AE F1
QP only	**0.7488**	-	-
RP only	-	0.7674	-
AE only	-	-	0.568
RP + AE	-	**0.7870**	**0.665**
QP + RP + AE	0.6803	0.6881	0.6160

Since the BERT architecture was so quick for training and evaluation (on the order of an hour), we found it preferable to use the two separate models. However, with a more complicated architecture (or a cleaner dataset) it may be possible to accommodate all three tasks at once.

Conclusion

- This new QA-plausibility task can allow practitioners to leverage noisy text from social media by applying automated data filtering
- A BERT baseline achieves impressive results on the task, but there is room to add better commonsense reasoning and structured linguistic features