MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics

Anthony Chen *UC Irvine*

Proceedings of EMNLP 2020

Gabriel Stanovsky
The Hebrew University*

Sameer Singh
UC Irvine

Matt Gardner
AI2 Irvine

*Work done while at AI2

Motivation

Approach

Collecting MOCHA

LERC: A Learned Metric

Results

Reading Comprehension

Given a passage of text, we want to probe a model’s *understanding* of it via question answering.

What is the Right Format?

Span-Selection

Pros:

- Easy to evaluate (F1)

Cons:

- Requires distractor spans.
- Answer must be spans, which restricts questions.

Multiple-Choice

Pros:

- Easy to evaluate (accuracy)

Cons:

- Distractor choices can introduce unwanted bias.
- Doesn’t allow model to synthesize own answer.

Generation

Pros:

- Allows any question to be asked and model to generate answer.
- No need for distractors.

Cons:

- Evaluation is hard.

What is the Right Format?

Span-Selection

Pros:

- Easy to evaluate (F1)

Cons:

- Requires distractor spans.
- Answer must be spans, which restricts questions.

Multiple-Choice

Pros:

- Easy to evaluate (accuracy)

Cons:

- Distractor choices can introduce unwanted bias.
- Doesn’t allow model to synthesize own answer.

Generation

Pros:

- Allows any question to be asked and model to generate answer.
- No need for distractors.

Cons:

- Evaluation is hard.

What is the Right Format?

Span-Selection

Pros:

- Easy to evaluate (F1)

Cons:

- Requires distractor spans.
- Answer must be spans, which restricts questions.

Multiple-Choice

Pros:

- Easy to evaluate (accuracy)

Cons:

- Distractor choices can introduce unwanted bias.
- Doesn’t allow model to synthesize own answer.

Generation

Pros:

- Allows any question to be asked and model to generate any answer.
- No need for distractors.

Cons:

- Evaluation is hard.

Generation is the “right” format.
Flexible and doesn’t require distractors!

Generation

Pros:

- Allows any question to be asked and model to generate any answer.
- No need for distractors.

Cons:

- Evaluation is hard.

But…Existing Metrics are Insufficient to Handle the Nuances of Reading Comprehension

Example 1: Agnostic to Passage

Passage: With the aid of his daughter, Abigail, Barabas recovers his former assets. Barabas then uses his daughter’s beauty to pit Lodowick and Mathias against each other.

Question: Why did Lodowick and Mathias fight?
Reference: **Over the affection of Abigail**
Candidate: **For Barabas’s daughter love.**

BLEU-1: 0
ROUGE-L: 0
METEOR: 0

Abigail is Baraba’s daughter in passage

Example 2: Reliance on Token Overlap

Passage: The strangest thing that has happened was when they were singing the Chinese National Anthem she was standing in front of the TV swaying and singing.

Question: What is probably true?
Reference: **They are watching the Olympics**
Candidate: **The Olympics are watching them**

BLEU-1: 0.80
ROUGE-L: 0.40
METEOR: 0.41

Semantic role is swapped but same tokens!

Example 3: Oversensitive to length

Passage: … Both doors are heavily soundproofed to prevent the accused from hearing what is behind each one. …

Question: What feature do the doors have?
Reference: soundproofed
Candidate: **They are heavily soundproofed to prevent the accused from hearing what's behind each one.**

BLEU-1: 0.07
ROUGE-L: 0.15
METEOR: 0.17

Candidate adds extra details which are correct.

Motivation

Approach

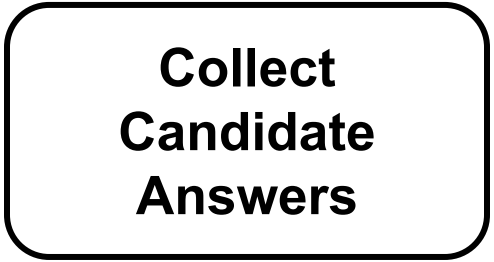
Collecting MOCHA

LERC: A Learned Metric

Results

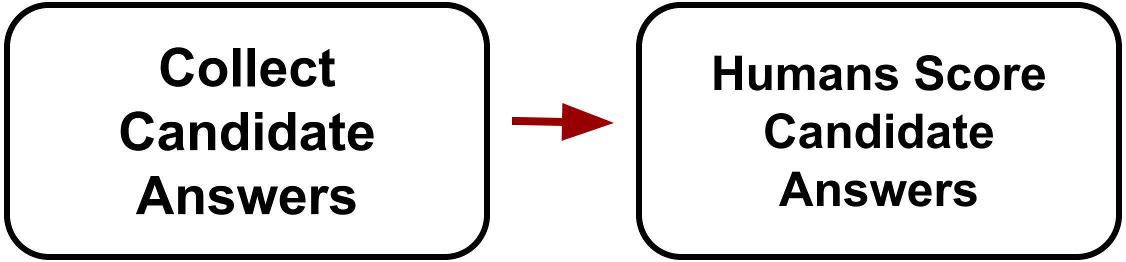
Train a *learned* metric to mimic human judgement scores.

Train a *learned* metric to mimic human judgement scores.



Collect Candidate Answers

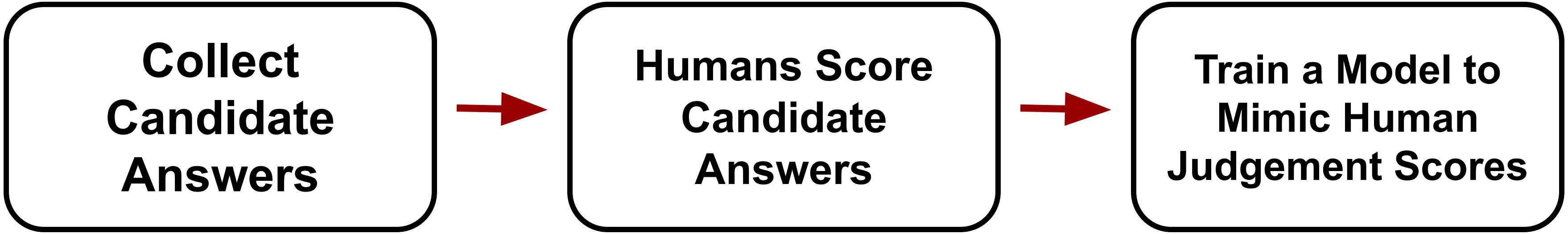
Train a *learned* metric to mimic human judgement scores.



Collect Candidate Answers

Humans Score Candidate Answers

Train a *learned* metric to mimic human judgement scores.



Collect Candidate Answers

Humans Score Candidate Answers

Train a Model to Mimic Human Judgement Scores

Motivation

Approach

Collecting MOCHA

LERC: A Learned Metric

Results

MOCHA is a dataset that pairs QA instances (passage, question, reference) with candidates and associated human judgement scores (1-5).

Collecting Candidate Answers

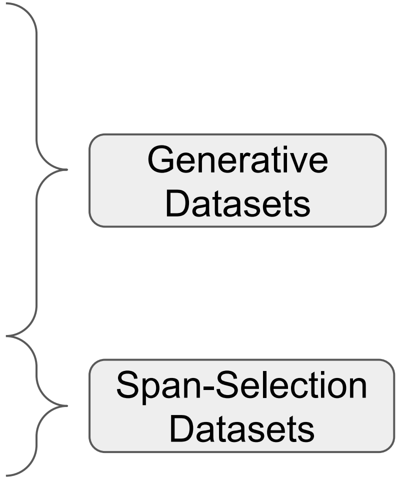
Collect candidates from 6 constituent QA datasets:

- NarrativeQA
- MCScript
- CosmosQA
- SocialIQA
- DROP
- Quoref

Collecting Candidate Answers

Collect candidates from 6 constituent QA datasets:

- NarrativeQA
- MCScript
- CosmosQA
- SocialIQA
- DROP
- Quoref



Generative Datasets

Span-Selection Datasets

MCScript, CosmosQA, & SocialIQA were originally multiple choice datasets

Collecting Candidate Answers

We generate candidates using:

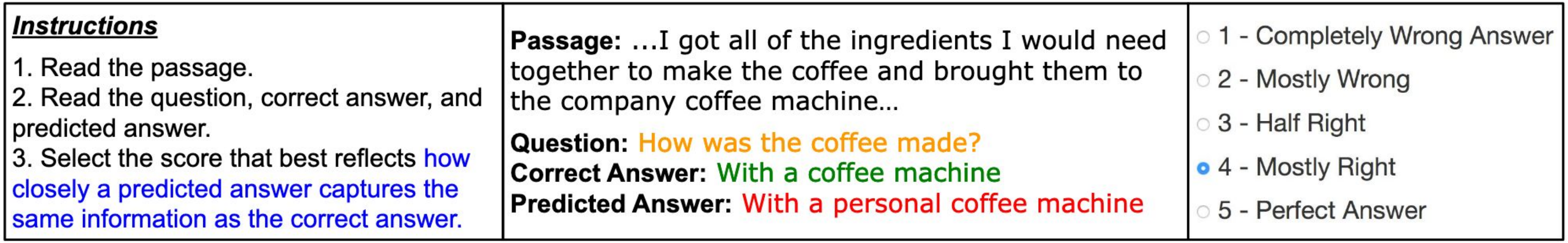
- Model outputs:
  - Multi-hop Pointer Generator Model
  - GPT-2 Small
  - BERT/NABERT Base
- Backtranslation

Collecting Candidate Answers

In total, MOCHA contains 40K candidates from 6 constituent datasets.

The 40K candidates are split into train (75%), validation (10%), and test (15%) sets.

Gathering Human Judgements



Instructions

1. Read the passage.
2. Read the question, correct answer, and predicted answer.
3. Select the score that best reflects how closely a predicted answer captures the same information as the correct answer.

Passage: ...I got all of the ingredients I would need together to make the coffee and brought them to the company coffee machine...

Question: How was the coffee made?
Correct Answer: With a coffee machine
Predicted Answer: With a personal coffee machine

○ 1 - Completely Wrong Answer
○ 2 - Mostly Wrong
○ 3 - Half Right
● 4 - Mostly Right
○ 5 - Perfect Answer

Each training instance gets 1 judgement score.

Each validation/test instance gets 3 and are averaged.

Motivation

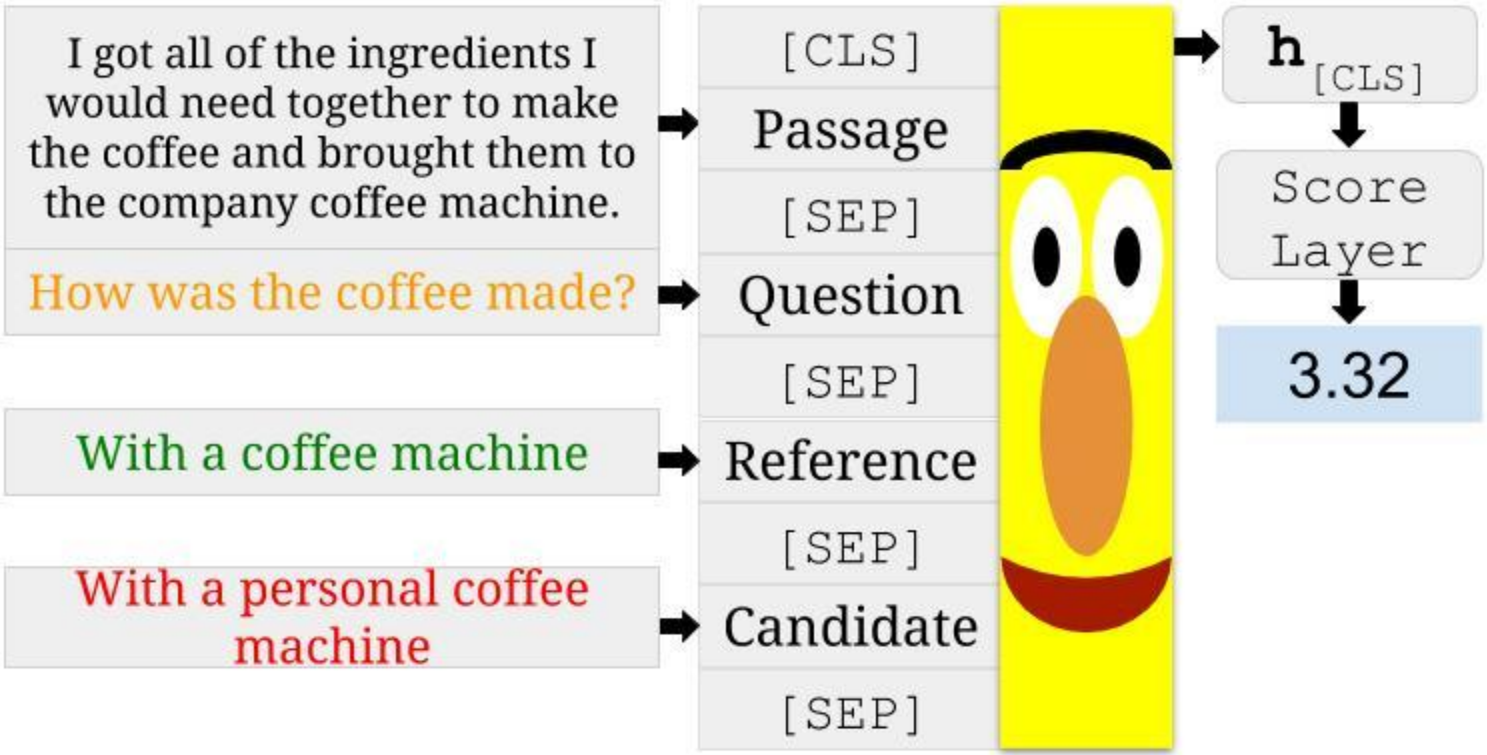
Approach

Collecting MOCHA

LERC: A Learned Metric

Results

LERC: A Learned Metric for Reading Comprehension



I got all of the ingredients I would need together to make the coffee and brought them to the company coffee machine.

How was the coffee made?

With a coffee machine

With a personal coffee machine

[CLS]

Passage

[SEP]

Question

[SEP]

Reference

[SEP]

Candidate

[SEP]

$\mathbf{h}_{\text{[CLS]}}$

Score Layer

3.32

Motivation

Approach

Collecting MOCHA

LERC: A Learned Metric

Results

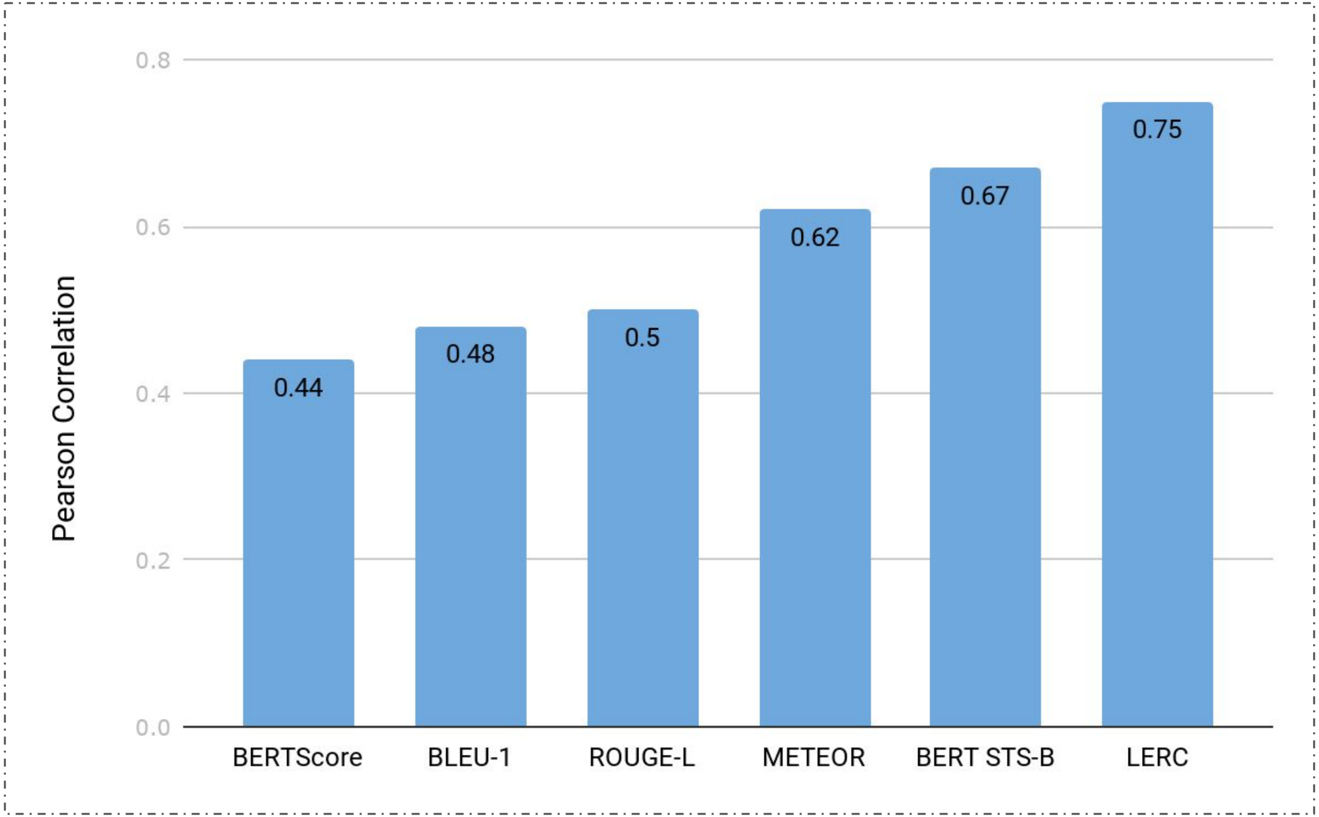
Experimental Setup

Baselines:

- BLEU-1, METEOR, ROUGE-L, and BERTScore.
- BERT paraphrase detection model trained on STS-B.

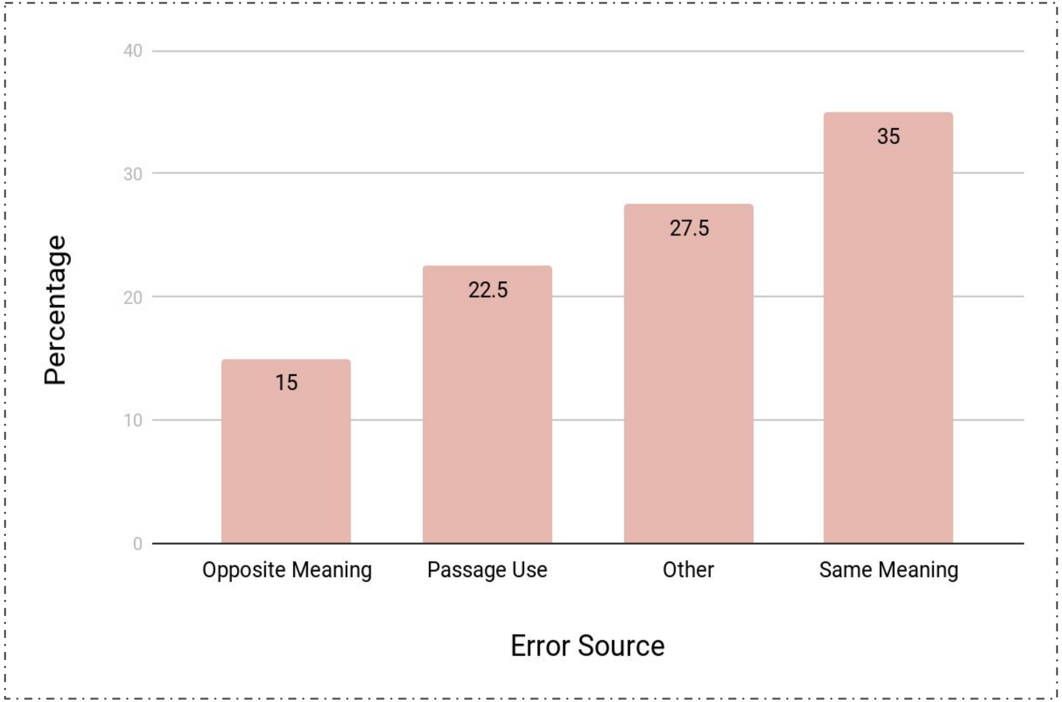
We train LERC in a out-of-dataset fashion: for each constituent dataset we evaluate on we hold out that dataset from our training set.

Pearson Correlation (Test Set)



Error Analysis (Validation Set)

We take the 40 instances with the largest gap between LERC and human scores and then categorize by error source.



Evaluating Robustness with Minimal Pairs

Given a *(passage, question, reference)* tuple, create two candidates that have high overlap, but one of which is much more correct.

200 minimal pairs total.

Passage: Norman is the supposed son of Frenchman deVac...As de Vac dies, he reveals Norman is Richard, the king's son and Edward's brother, who he kidnapped.
Q: Who is the Frenchman de Vac?
Ref: a fencing master who kidnapped Norman

Cand1: a fencing master who kidnapped Richard
Cand2: a fencing master who kidnapped Edward

Score1: 5
Score2: 2

Evaluating Robustness with Minimal Pairs

Given a minimal pair:

(pass, ques, ref, cand1)

(pass, ques, ref, cand2)

do metrics assign a higher score to the better candidate?

Passage: Norman is the supposed son of Frenchman deVac...As de Vac dies, he reveals Norman is Richard, the king's son and Edward's brother, who he kidnapped.
Q:Who is the Frenchman de Vac?
Ref: a fencing master who kidnapped Norman

Cand1: **a fencing master who kidnapped Richard**
Cand2: **a fencing master who kidnapped Edward**

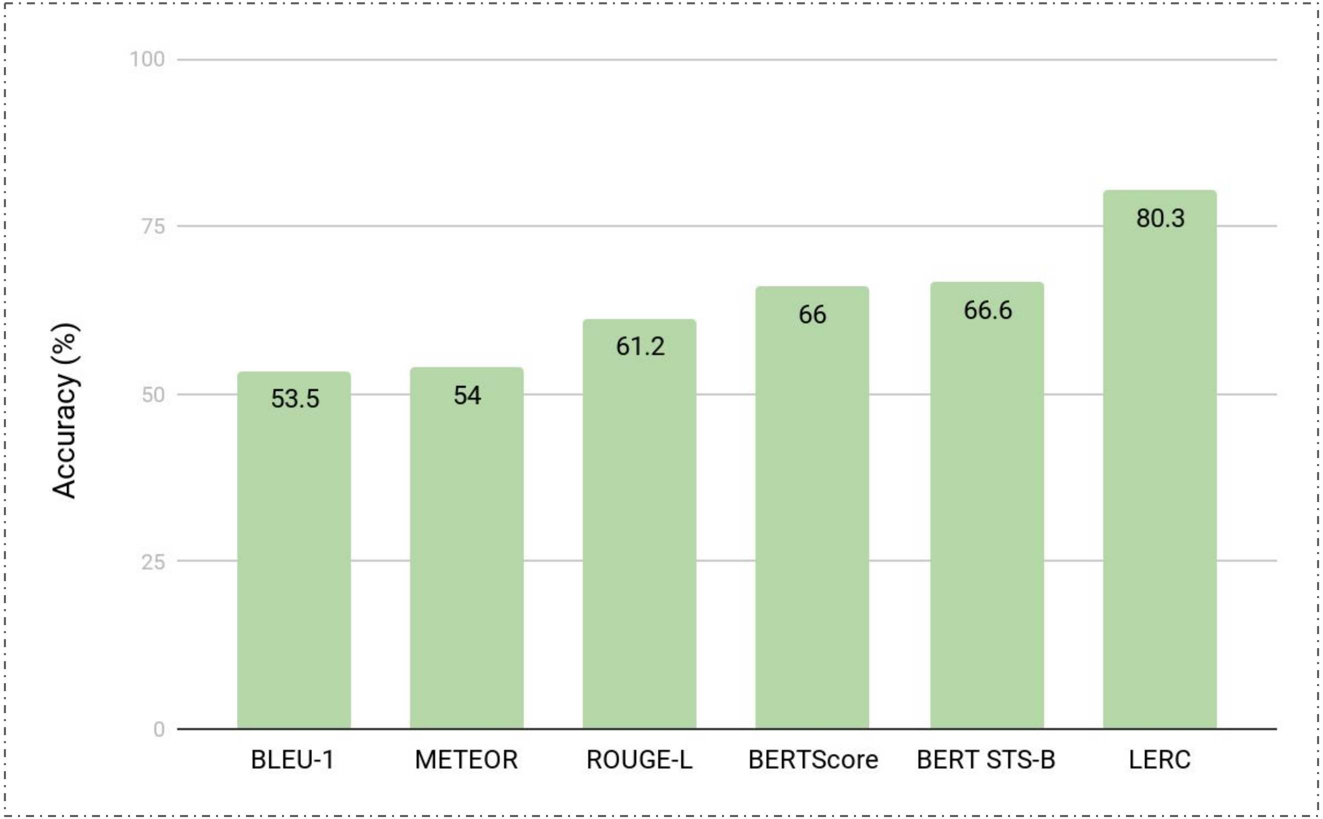
Score1: 5
Score2: 2

Evaluating Robustness with Minimal Pairs

Minimal pairs created to test understanding of variety of phenomena:

- Coreference
- Hyponymy
- Negation
- Semantic Role
- Syntax
- Word Sense

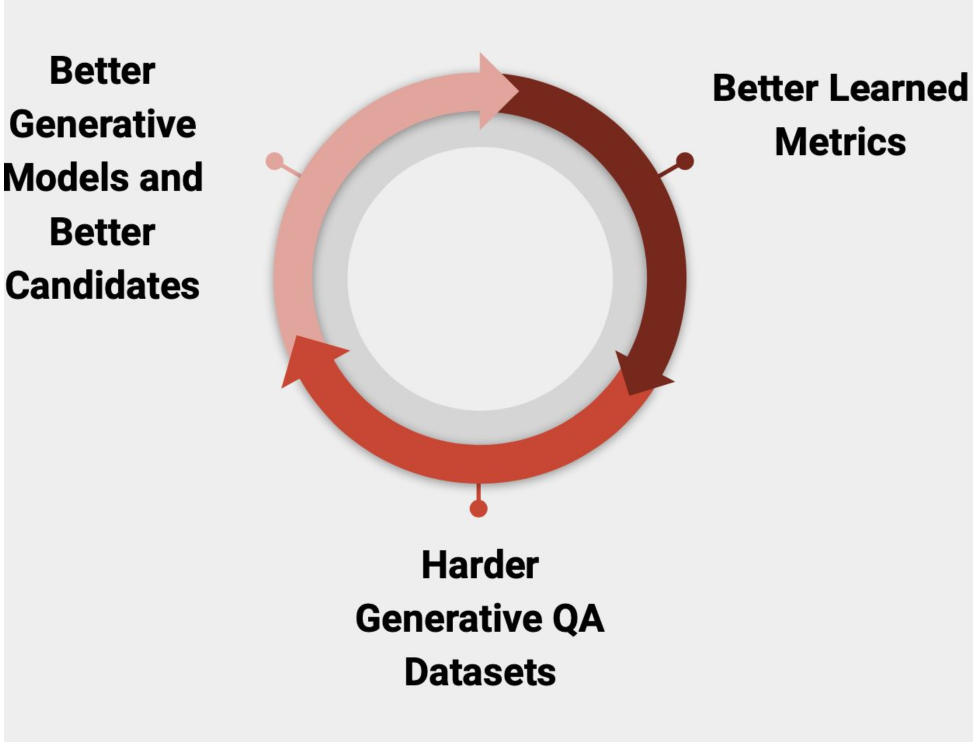
Results on Minimal Pairs



Takeaways

Learned Metric > Engineered Metric (with training data)

LERC is weak on some phenomena (need more targeted training data)



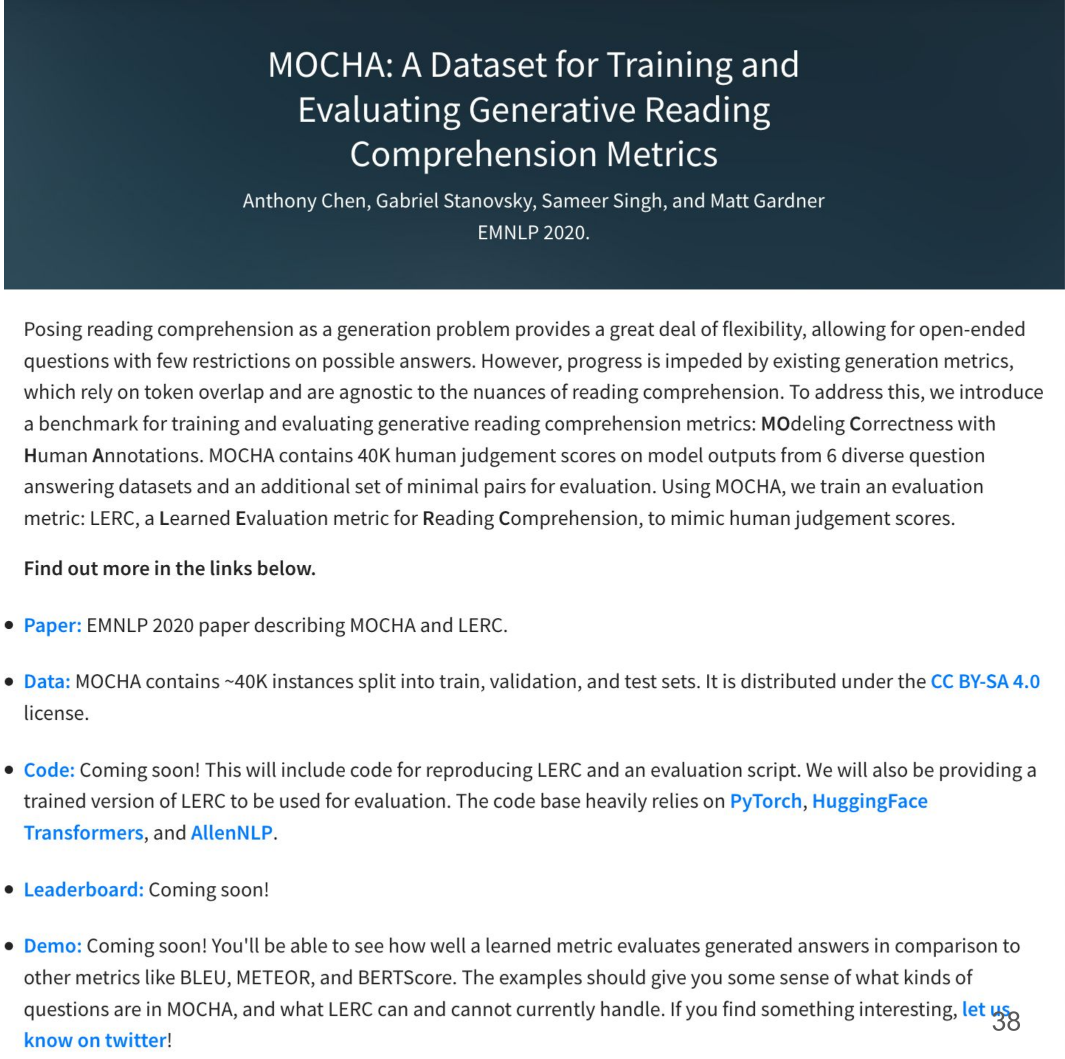
Better Generative Models and Better Candidates

Better Learned Metrics

Harder Generative QA Datasets

Landing Page: allennlp.org/mocha

(Check out the leaderboard and demo!)



MOCHA: A Dataset for Training and Evaluating Generative Reading Comprehension Metrics

Anthony Chen, Gabriel Stanovsky, Sameer Singh, and Matt Gardner
EMNLP 2020.

Posing reading comprehension as a generation problem provides a great deal of flexibility, allowing for open-ended questions with few restrictions on possible answers. However, progress is impeded by existing generation metrics, which rely on token overlap and are agnostic to the nuances of reading comprehension. To address this, we introduce a benchmark for training and evaluating generative reading comprehension metrics: **MO**deling **C**orrectness with **H**uman **A**nnotations. MOCHA contains 40K human judgement scores on model outputs from 6 diverse question answering datasets and an additional set of minimal pairs for evaluation. Using MOCHA, we train an evaluation metric: LERC, a **L**earned **E**valuation metric for **R**eading **C**omprehension, to mimic human judgement scores.

Find out more in the links below.

- **Paper:** EMNLP 2020 paper describing MOCHA and LERC.
- **Data:** MOCHA contains ~40K instances split into train, validation, and test sets. It is distributed under the **CC BY-SA 4.0** license.
- **Code:** Coming soon! This will include code for reproducing LERC and an evaluation script. We will also be providing a trained version of LERC to be used for evaluation. The code base heavily relies on **PyTorch**, **HuggingFace Transformers**, and **AllenNLP**.
- **Leaderboard:** Coming soon!
- **Demo:** Coming soon! You'll be able to see how well a learned metric evaluates generated answers in comparison to other metrics like BLEU, METEOR, and BERTScore. The examples should give you some sense of what kinds of questions are in MOCHA, and what LERC can and cannot currently handle. If you find something interesting, **let us know on twitter**!

Thanks!

If you want to chat over a mocha 👇

Website: anthonywchen.github.io
Email: anthony.chen@uci.edu
Tweeter: @_anthonychen