Extracting cause and effect from sentences

Italo Luis da Silva

King's College London

2023-03-28

Problem statement

- Fine-grained causal reasoning[1][2]
- Extract cause and effect from context
  - Substrings of the context, not trigger words
- Classify the relation between cause and effect
  - **Cause**: sufficient and necessary condition
    - ★ Cause is enough to make the effect happen
    - ★ Must happen for the effect to happen
  - **Enable**: sufficient but not necessary condition
    - ★ Cause is enough to make the effect happen
    - ★ There are other conditions that can also lead to the effect
  - **Prevent**: sufficient condition to stop the effect from happening
    - ★ If the cause happens, the effect cannot happen

[1] Towards Fine-grained Causal Reasoning and QA

[2] github.com/YangLinyi/Fine-grained-Causal-Reasoning

Example (pt. 1)

There can be more than one pair of cause/effect in the context:

> *The firm's gross margin is set to stabilize* as *Harley refocuses its efforts on more profitable markets*, *and our base case assumes that it stabilizes around 32% in 2029, helped by a more measured approach to entering new markets.*

- Cause$_1$: Harley refocuses its efforts on more profitable markets
- Effect$_1$: The firm's gross margin is set to stabilize
- **Relation$_1$**: cause

Example (pt. 2)

There can be more than one pair of cause/effect in the context:

> *The firm's gross margin is set to stabilize as Harley refocuses its efforts on more profitable markets, and our base case assumes that it stabilizes around 32% in 2029, helped by a more measured approach to entering new markets.*

- Cause$_2$: a more measured approach to entering new markets
- Effect$_2$: it stabilizes around 32% in 2029
- **Relation**$_2$: enable

Dataset statistics

Extraction

Split	# Examples	# Relations	# Causes	# Effects
Dev	2482	3224	3224	3238
Train	19892	25938	26174	26121
Test	2433	3045	3065	3062

Classification

Split	# Relations	% Cause	% Prevent	% Enable
Dev	3224	63.78%	5.40%	30.82%
Train	25938	63.05%	5.90%	31.05%
Test	3045	64.00%	5.38%	30.62%

Preliminary results

Model name	Token F1	EM	Class Acc.	Class F1
GenQA (extraction)	81.09%	48.14%	-	-
GenQA (joint)	79.47%	52.16%	71.19%	54.08%
Sequence Labelling	73.23%	22.95%	-	-
BERT (extraction)[3]	84.37%	51.48%	-	-
BERT (classification)[3]	-	-	70.43%	71.74%
BERT (joint)[3]	-	21.21%	-	-

[3] Baseline from the original paper

Problem with EM evaluation

- Exact Match is a flawed metric because it punishes the model for correct answers that don't exactly match the ground truth
- Different annotators will annotate the same sentence differently
- The model output won't learn the "style" of all annotators simultaneously

Exact Match example 1

Annotation

BB&T and SunTrust have completed their merger, forming Truist, which we believe will drive the next step up in profitability for the franchises.

Model prediction

BB&T and SunTrust have completed their merger, forming Truist, which we believe will drive the next step up in profitability for the franchises.

Exact Match example 2

Annotation

Given Tulip's lack of profitability (management has stated the business was not profitable at the time of the October 2019 acquisition), we do not believe the business maintains a cost advantage.

Model prediction

Given Tulip's lack of profitability (management has stated the business was not profitable at the time of the October 2019 acquisition), we do not believe the business maintains a cost advantage.

Proposed solution

- Use an RL framework to train the model to produce correct answers instead of trying to match the ground truth exactly
- The RL loop enables detection of such correct answers by reconstructing the original text from the structured output
- An entailment model is used to detect whether the reconstructed text follows from the original text

RL framework: forward pass

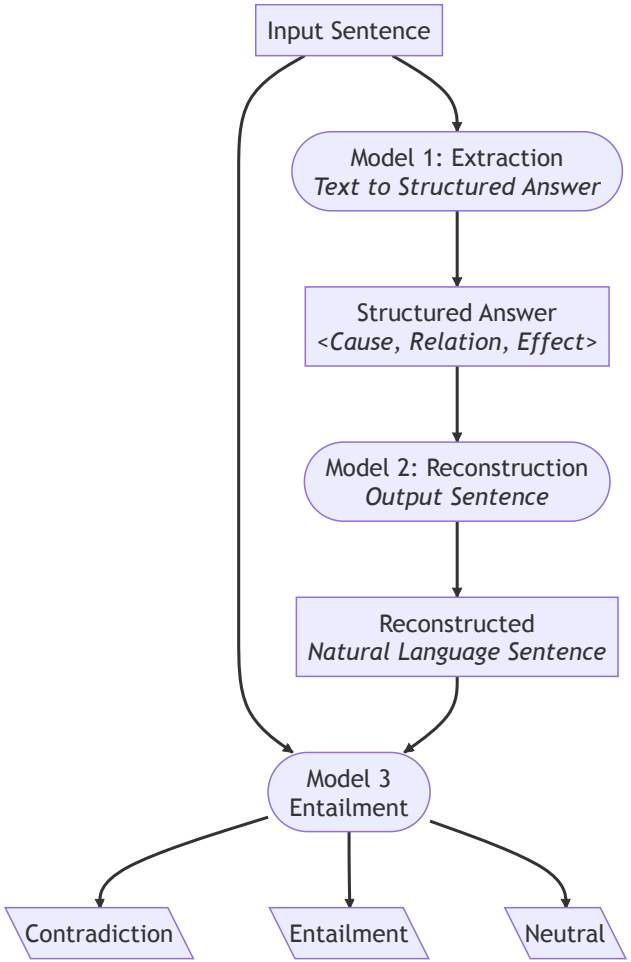

Figure 1: RL forward pass

RL framework: backward pass

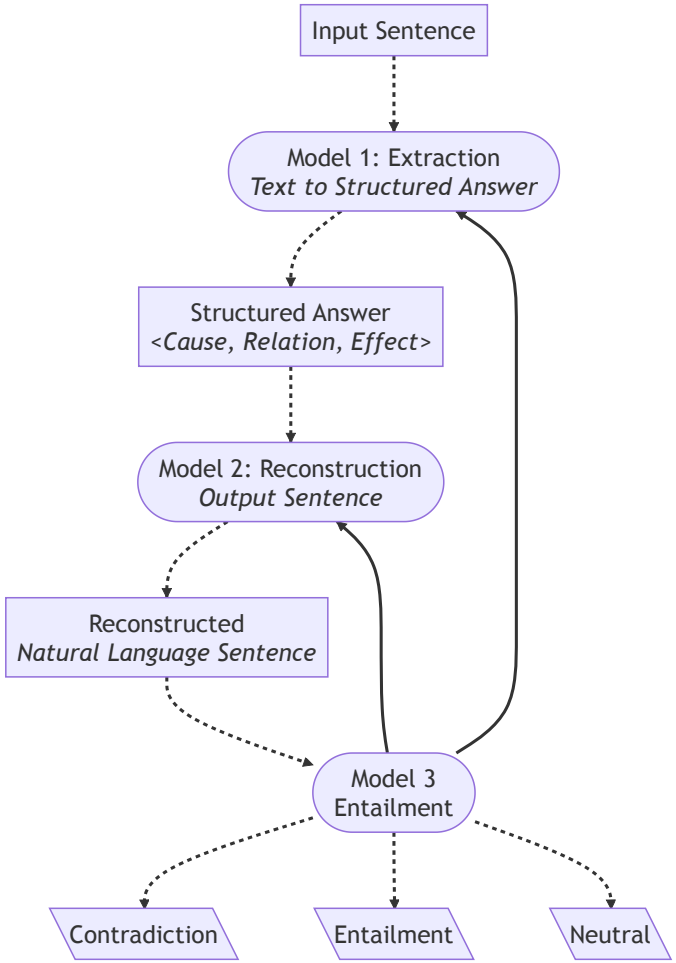

Figure 2: RL backward pass

RL framework: models

- Model 1: Extraction
  - Same as the GenQA (joint) above
  - T5-base generative QA
- Model 2: Reconstruction
  - Now: T5-base generative QA
  - Maybe: specialised structured to text model
- Model 3: Entailment
  - DeBERTa-base-MNLI
  - Easy problem: any transformer works here
- Models are finetuned for a few epochs before RL

RL framework: data

- Model 1: original extraction dataset
- Model 2:
  - Input: structured answers
  - Output: reconstructed spans from the original context
- Model 3:
  - Input: sentence 1 and sentence 2
  - Sentence 1 is always the context
  - Sentence 2:
    - ★ Entailment: sentence from the same context
    - ★ Neutral: sentence from another context
    - ★ Contradiction: sentence from the same context with cause and effect flipped

Next steps

- Implementation of the RL framework
  - Which library to use
  - How to connect the models
- Experiments to determine the best algorithm, setup and rewards

Current issues

- Model size in memory
  - 3 transformers means high VRAM usage
  - I can use small versions for development, but I need large versions for the final results
- How to best train this?
  - v1: alternate between freezing model 1 and training model 2, and vice-versa
  - v2: train all models at the same time?

Thanks!

github.com/oyarsa/event_extraction/self_critique

Slides: t.ly/1X2S