Extracting cause and effect from sentences

Italo Luis da Silva

2023-03-28

King's College London

Problem statement

- Fine-grained causal reasoning (Yang et al., 2022)[1][2]
- Extract cause and effect from the context
  - Spans 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 effect to happen
  - **Enable**: sufficient but not necessary condition
    - Cause is enough to make the effect happen
    - Other conditions 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 (part 1)

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 (part 2)

There can be more than one relation 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%	-	-
Sequence Labelling (extraction)	73.23%	22.95%	-	-
GenQA (joint)	79.47%	52.16%	71.19%	54.08%
BERT (extraction)[3]	84.37%	51.48%	-	-
BERT (classification)[3]	-	-	70.43%	71.74%
BERT (joint)[3]	-	21.21%	-	-

[3] Baselines from the original paper

Problem with Exact Match evaluation

- Exact Match is an incomplete metric because it requires the output to be 100% identical to the annotation
- Different annotators can annotate the same sentence differently
- The model won't learn the "style" of all annotators simultaneously
  - It can't be exactly right all the time

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.

Cause Effect

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.

Cause Effect

Proposed solution

- A human can see that the prediction is correct even if it's not an exact match
- **Apply Reinforcement Learning to train the model to produce correct answers instead**
  - Keep EM as a guardrail metric
- RL enables the detection of correct answers by reconstructing the original text from the extracted
- An entailment model detects whether the reconstructed text follows from the original
  - Entailment = correct answer

RL framework: forward pass

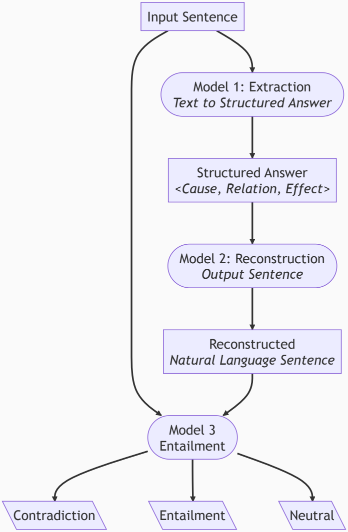

Figure 1: RL forward pass

RL framework: backward pass

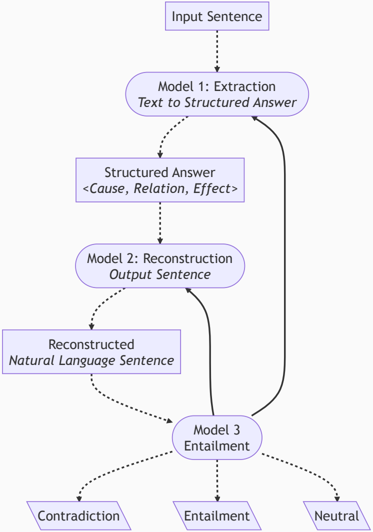

Figure 2: RL backward pass

RL framework: models

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

Next steps

- Implementation of the RL framework
- Experiments to determine the best algorithm, setup, rewards, etc.

Current issues

- Model size in memory
  - 3 transformers means high VRAM usage
  - Small versions for development (t5-small, deberta-v3-xsmall)
  - Larger versions for final results (t5-base, deberta-base-mnli)
    - Batch size
    - Multiple GPUs
- 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