A Manual Analysis of SQuAD Questions

We conduct a manual analysis of the 68 most-frequent n-gram question openers from SQuAD, to investigate the prevalence of causal questions which exist. We restrict the question openers to the most frequent n-grams which cover 80% of the dataset, excluding unigrams other than "why" because they were uninformative. Two researchers hand-labeled each opener as one of: cause-and-effect, not cause-and-effect, and unknown, where unknown question openers could be the start of cause-and-effect questions but would require reading the entire question to determine. A third researcher tie-broke disagreements. The average Cohen's Kappa (Cohen, 1960) is 0.72 (substantialagreement). Of the 87,599 questions in SQuAD which could be labeled by the 68 most-frequent n-gram question openers, 1,194 (1.4%) were cause-and-effect, 29,540 (33.7%) were not cause-and-effect, and 56,865 (64.9%) were unknown. The full list of labeled question openers is found in our Github repo.

B Causal Extraction Details

For the causal extraction section of the pipeline, we modified the "as" pattern. The original pattern is formulated as:

> &R@Complete@ (,) (-such/-same/-seem/-regard/-regards/-regarded/-view/-views/-viewed/-denote/-denoted/-denotes) as (-if/-follow/-follows/-&adv) &C@Complete@

Where @Complete@ indicates that the text piece is a clause which must have predicate and subject, "-" indicates tokens followed should not be matched, and "()" indicates tokens that are not required. &R and &C represent the extracted cause and effect. However, the original pattern assumes that the cause is always before "as." In reality, "as" can be included before both the cause and the effect, such as in the following example:

> Some renewable resources are too expensive to be widely used. As the technology improves and more people use renewable energy, the prices will come down. The cost of renewable resources will go down relative to fossil fuels as we use fossil fuels up.

For this example, the causal phrase extracted by original pattern is "Some renewable resources are too expensive to be widely used." The effect phrase extracted by original pattern is "The technology improves and more people use renewable energy, the prices will come down."

We implement a new pattern (pattern-id = 145): ",/;/./– As &C , &R". For each cause-and-effect extracted in the original pattern, if the new pattern is also a match, we replace the cause and effect with the output from the new pattern. For the example sentences above, the causal phrases extracted by our new pattern is "the technology improves and more people use renewable energy." The corresponding effect phrase extracted by new pattern is "The prices will come down."

C Crowdworking Task Interface

Figure 2 contains the cause (bolded and highlighted in orange) and effect (underlined and highlighted in blue) shown to workers when evaluating the quality of an extracted cause and effect. Figure 3 contains a sample stimulus showing the intended answer and generated question.

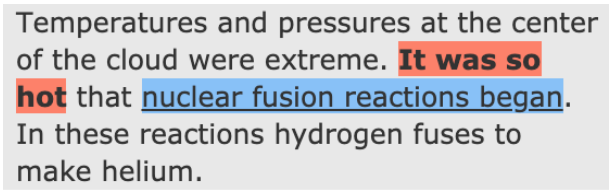

Figure 2: Example crowdworking presentation of passage, cause, and effect, from TQA dataset.

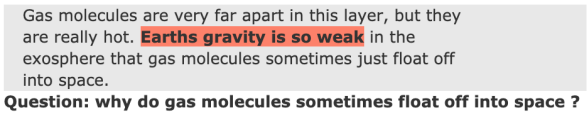

Figure 3: Example crowdworking presentation of passage, intended answer, and generated question, from TQA dataset.

Table 10 contains the crowdworker ratings for the Cao et al. (2016) causal extraction system, stratified by typology. Each main typology category is further stratified by the type and sub-type of link. For example, the Adverbial link category contains two types: (A) Anaphoric and (B) Cataphoric. The Anaphoric category is further segmented into three sub-types: (1) Implicit Cohesion (e.g.,"therefore") (2) Pronominal Cohesion (e.g., "for this reason"),

and (3) Pronominal + Lexical Cohesion (e.g., "because of" *NP*) (Altenberg, 1984). We refer to the main category by name, with the subcategories denoted with codes, e.g., Adv.1.a.

Category	TQA	#T	SQuAD	#S
Adv.1.a	33	34	16	21
Adv.1.b	2	3	2	2
Adv.1.c	0	4	1	3
Adv.2.a	2	2	1	2
Prep.1.a	8	11	16	21
Sub.1.a	12	27	18	29
Sub.2.a	2	3	0	4
Sub.3.a	2	3	3	3
C-I.1.a	3	4	3	5
C-I.1.b	3	3	2	3
C-I.1.c	2	3	2	3
C-I.1.e	*	0	1	1
C-I.1.f	1	1	*	0
C-I.1.h	*	0	1	1
C-I.2.a	0	2	2	2

Table 10: Number of extracted relations that were labeled as causal by crowdworkers for original Cao et al. (2016) system, organized by linguistic category. #T is total number in TQA and #S is total number in SQuAD. '*' indicates no relation found.

Type	**Recall**	**#T**	**Recall**	**#S**
Adv.1.a	0.76	1309	0.60	229
Adv.1.b	0.55	28	0.32	8
Adv.1.c	0.37	94	0.39	15
Adv.2.a	0.45	3	0.67	3
Total Adv.	0.73	1434	0.58	255
Prep.1.a	0.72	470	0.56	275
Total Prep.	0.72	470	0.56	275
Sub.1.a	0.69	1146	0.52	385
Sub.2.a	0.73	57	0.54	35
Sub.3.a	0.78	70	0.59	15
Total Sub.	0.70	1273	0.53	435
C-I.1.a	0.71	99	0.57	48
C-I.1.b	0.61	33	0.56	11
C-I.1.c	0.79	29	0.90	13
C-I.1.e	*	0	0.12	1
C-I.1.f	1	1	*	0
C-I.1.h	*	0	0.14	1
C-I.2.a	0.61	20	0.32	6
Total C-I	0.69	182	0.59	80

Table 11: Average cause/effect presence recall in the TQA and SQuAD datasets, categorized by typology. Questions are generated by ProphetNet fine-tuned on Syn-QG. #T refers to number of questions in TQA; #S the same for SQuAD. '*' indicates no relation found.

D Model specifics

Causal Extraction: The approximate runtime for this algorithm on an Intel(R) Core(TM) i7-6850K CPU machine is 72 hours for the TQA dataset and 24 hours for SQuAD.

Question Generation: ProphetNet has 391,324,672 parameters; our version is unchanged from Qi et al. (2020). We finetune the provided question generation model checkpoint, which is a 16 GB model fine-tuned on SQuAD. The approximate runtime to fine tune this model on an auxiliary dataset on a p3.2xlarge AWS ec2[6] machine is 0.5 hours. For our fine-tuning process, we train for 3 epochs with a learning rate of 1e-6 with a batch size of 1. The rest of parameters are kept the same as what is found in the examples provided by the ProphetNet GitHub repository README. Approximate inference time is 10 minutes for TQA and 5 minutes for SQuAD. We utilize the Fairseq library (Ott et al., 2019) to facilitate the training and inference processes. Comparing the fine-tuned model's generated questions to the Syn-QG questions, the fine-tuned QG model achieves 57.01 training BLEU and 53.89 test BLEU (Papineni et al., 2002).

Question Answering: The QA model we utilize has 334,094,338 parameters. The approximate runtime to fine tune this model on an auxiliary dataset on a p3.2xlarge AWS ec2 machine is 0.5 hours. For our fine-tuning process, we train for 10 epochs with an initial learning rate of 1e-5, a batch size of 4, 500 warm-up steps, and a weight decay of 0.01. The rest of the parameters are the defaults set by the HuggingFace TrainingArguments class. We also truncate each example to a max of 512 tokens. Approximate inference time is 1 minutes for TQA and 1 minute for SQuAD. On the Syn-QG dataset, the fine-tuned QA model achieves 0.97 training F1 and 0.95 test F1.

E Cause/Effect Present Results

Table 11 shows the results for the automatic cause/effect present metric segmented by typology categories. For SQuAD, the lowest-performing category is Subordination, which corresponds to the category with the lowest proportion of extracted relationships labeled as causal by crowdworkers (Section 4).

[6] https://aws.amazon.com/ec2/