Learning to Explain: Datasets and Models for Identifying Valid Reasoning Chains in Multihop Question-Answering

Harsh Jhamtani [1] **Peter Clark** [2]
[1] School of Computer Science, Carnegie Mellon University
[2] Allen Institute for AI
jharsh@cs.cmu.edu, peterc@allenai.org

Abstract

Despite the rapid progress in multihop question-answering (QA), models still have trouble explaining *why* an answer is correct, with limited explanation training data available to learn from. To address this, we introduce three explanation datasets in which explanations formed from corpus facts are annotated. Our first dataset, eQASC, contains over 98K explanation annotations for the multihop question answering dataset QASC, and is the first that annotates *multiple* candidate explanations for each answer. The second dataset eQASC-perturbed is constructed by crowd-sourcing perturbations (while preserving their validity) of a subset of explanations in QASC, to test consistency and generalization of explanation prediction models. The third dataset eOBQA is constructed by adding explanation annotations to the OBQA dataset to test generalization of models trained on eQASC. We show that this data can be used to significantly improve explanation quality (+14% absolute F1 over a strong retrieval baseline) using a BERT-based classifier, but still behind the upper bound, offering a new challenge for future research. We also explore a delexicalized chain representation in which repeated noun phrases are replaced by variables, thus turning them into *generalized reasoning chains* (for example: "X is a Y" AND "Y has Z" IMPLIES "X has Z"). We find that generalized chains maintain performance while also being more robust to certain perturbations.[1]

1 Introduction

While neural systems have become remarkably adept at question answering (QA), e.g., (Clark and Gardner, 2018), their ability to *explain* those answers remains limited. This creates a barrier for deploying QA systems in practical settings, and limits their utility for other tasks such as education and tutoring, where explanation plays a key role. This need has become particularly important with *multihop question-answering*, where multiple facts are needed to derive an answer. In this context, seeing a *chain of reasoning* leading to an answer, can help a user assess an answer's validity. Our research here contributes to this goal.

Q: What can cause a forest fire?
(1) rain (2) static electricity (3) microbes (4) ...
A: static electricity
Q+A (declarative): Static electricity can cause a forest fire.

Explanation (reasoning chain): [positive (valid)]
Static electricity can cause sparks // (from corpus)
AND *Sparks can start a forest fire* // (from corpus)
→ *Static electricity can cause a forest fire* // (Q+A)

Explanation (Generalized reasoning chain, GRC):
X *can cause* **Y AND Y** *can start* **Z** → **X** *can cause* **Z**

Figure 1: Our datasets contain annotated (valid and invalid) *reasoning chains* in support of an answer, allowing explanation classifier models to be trained and applied. We also find that using a variabilized version of the chains improves the models' robustness.

We are interested in questions where the decomposition into subquestions - hence the explanation structure - is *not evident from the question*, but has to be found. For example, "Does a suit of armor conduct electricity?" might be answered (hence explained) by first identifying what material armor is made of, even though the question itself does not mention materials. This contrasts with earlier multihop QA datasets, e.g., HotpotQA (Yang et al., 2018), where the explanation structure is evident in the question itself. For example, "What nationality was James Miller's wife?" implies a chain of reasoning to first finds Miller's wife, then her nationality. Such cases are easier but less representative of natural questions. Multihop datasets of the kind where explanation structure is not evident

[1] Code and datasets can be found at https://allenai.org/data/eqasc

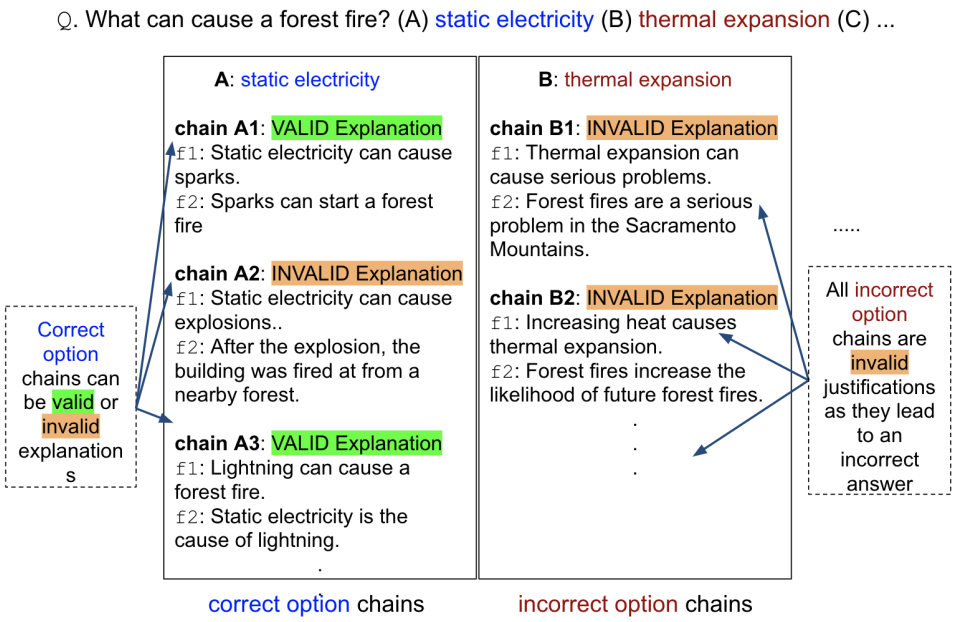

Figure 2: QASC contains multiple-choice questions, plus one gold (valid) reasoning chain for the correct answer. To find valid reasoning chains, we first generate candidates for each answer option using a 2-step retrieval process (Section 3.2). We then collect annotations for the correct answer option chains to train and evaluate models to detect valid reasoning chains. (Above, chains A1 and A3 are valid, while A2, B1, and B2 are invalid).

include OpenBookQA (Mihaylov et al., 2018) and more recently QASC (Khot et al., 2020). However, although providing QA pairs, these datasets provide limited explanation information. OpenBookQA does not come with any explanation data, and QASC only provides a single gold explanation for each answer, while in practice there may be multiple valid explanations.

To alleviate this lack of data, we contribute three new datasets: The first (and largest) is `eQASC`, containing annotations on over 98K candidate explanations for the QASC dataset, including on *multiple* (typically 10) possible explanations for each answer, including both valid and invalid explanations. The second, `eQASC-perturbed`, contains semantically invariant perturbations of a subset of QASC explanations, for better measuring the generality of explanation prediction models. Finally `eOBQA` adds adding explanation annotations to the OBQA test set, to further test generality of models trained on `eQASC`. In addition, we use these datasets to build models for detecting valid explanations, to establish baseline scores. Finally, we explore a delexicalized chain representation in which repeated noun phrases are replaced by variables, thus turning them into *generalized reasoning chains*, as illustrated in Figure 1. We find that generalized chains maintain performance while also being more robust to perturbations, suggesting a promising avenue for further research.

2 Related Work

In the context of QA, there are multiple notions of explanation/justification, including showing an authoritative, answer-bearing sentence (Perez et al., 2019), a collection of text snippets supporting an answer (DeYoung et al., 2020), an attention map over a passage (Seo et al., 2016), a synthesized phrase connecting question and answer (Rajani et al., 2019), or the syntactic pattern used to locate the answer (Ye et al., 2020; Hancock et al., 2018). These methods are primarily designed for answers to "lookup" questions, to explain where and how an answer was found in a corpus.

For questions requiring inference, the focus of this paper, an explanation is often taken as the chain of steps (typically sentences) leading to an answer. HotpotQA's support task goes partway towards this by asking for answer-supporting sentences (but not how they combine) (Yang et al., 2018). The R4C dataset takes this further, annotating how (and which) HotpotQA supporting sentences chain together (Inoue et al., 2019). However, in HotpotQA and R4C, the decomposition (hence structure of the explanation) is evident in the question (Mihaylov et al., 2018), simplifying the task. More recently,

multihop datasets where the decomposition is not evident have appeared, e.g., WikiHop (Welbl et al., 2018), OBQA (Mihaylov et al., 2018), and QASC (Khot et al., 2020), posing more a realistic explanation challenge. However, explanation annotations are sparse (only QASC contains a single gold explanation per question), limiting their support for both training and evaluation of explanation systems, hence motivating this work.

Finally there are a few human-authored explanation datasets. e-SNLI (Camburu et al., 2018) adds crowdsourced explanations to SNLI entailment problems (Bowman et al., 2015), and CoS-E (Rajani et al., 2019) adds explanations for CommonsenseQA questions (Talmor et al., 2019). This work differs from ours in two ways. First, the authored explanations are single-hop, directly linking a question to an answer. Second, the datasets were primarily designed for (explanation) language generation, while our goal is to assemble explanations from an authoritative corpus so that they have credible provenance.

Our work is quite different from prior work focusing on textual entailment. Our goal is not to decide if a sentence is entailed, but to identify a valid explanation for why. For example, a SciTail (Khot et al., 2018) model may predict that "metals conduct heat" entails "a spoon transmits energy", but not offer an explanation as to why. Our work fills this gap by providing an explanation (e.g., "spoon is made of metal", "heat is energy" from a larger retrieved context). Similarly, another entailment-based dataset is FEVER (Thorne et al., 2018), testing where a larger context entails a claim. However, the FEVER task requires finding a context sentence that simply paraphrases the claim, rather than a reasoned-based explanation from more general statements - the aim of this work.

3 Explanation Datasets

We now present our new datasets, first describing how we construct candidate chains for each QA pair, and then how they were annotated.

3.1 Task Definition

We consider the task where the input is a question Q, (correct) answer A, and a corpus of sentences T. The (desired) output is a valid reasoning chain, constructed from sentences in T, that supports the answer. We define a reasoning chain as a sequence of sentences $C = [s_1, ..., s_n]$ plus a conclusion sentence H, and a *valid* reasoning chain as one where C *entails* H. Following the textual entailment literature (Dagan et al., 2013), we define entailment using human judgements rather than formally, i.e., C entails H if a person would reasonably conclude H given C. This definition directly aligns with our end-goal, namely to provide users with a credible reason that an answer is correct.

	Train	Dev	Test
Total number of questions	8134	926	920
Total no. of chains tagged	80449	9190	9141
No. of valid chains	21551	2186	2210
No. of invalid chains	58898	7004	6931

Table 1: Summary statistics for eQASC, the annotated chains for the correct answers in QASC. Each chain is tagged by three annotators, and we use majority judgement.

For generating candidate chains C, we construct each C from sentences in the corpus T, as described below. Following the design of the QASC dataset, we consider just 2-sentence chains, as this was the maximum chain length used in its creation, although our approach could be extended to N-sentence chains.

3.2 Candidate Chain Construction

Given Q + A, we use the procedure described in (Khot et al., 2020) to assemble candidate chains from T (below). This procedure aims to find plausible chains by encouraging word overlap:

(1) Using ElasticSearch (a standard retrieval engine), retrieve K (=20 for efficiency) facts F1 from T using Q+A as the search query.
(2) For each fact f1 $\in$ F1, retrieve L (=4 to promote diversity) facts F2, each of which contains at least one word from Q+A $\setminus$ f1 and from f1 $\setminus$ Q+A;
(3) Remove [f1,f2] pairs that do not contain any word from Q or A;
(4) Select the top M (here, =10) [f1,f2] pairs sorted by the sum of their individual IR (ElasticSearch) scores.

Step (3) ensures that the chain contains at least some mention of part of Q and part of A, a minimal requirement. Step (4) imposes a preference for chains with greater overlap with Q+A, and between f1 and f2. Note that this procedure does not guarantee *valid* chains, rather it only finds candidates that may be plausible because of their overlap. Some example chains are produced by this method are shown in Figure 2. In all our experiments, we use

the QASC corpus[2] as the corpus T, namely the same corpus of 17M cleaned up facts as used in (Khot et al., 2020).

3.3 eQASC - Explanations for QASC

The original QASC dataset includes only a single gold (valid) reasoning chain for each correct answer, and no examples of invalid chains. To develop a richer explanation dataset, suitable for both training and evaluation, we generate eQASC as follows. First, we use the above algorithm to generate (up to) 10 candidate chains for each Q + correct answer option A pair. This resulted in a total of **98780 chains** for QASC's 9980 questions.

We then use (Amazon Turk) crowdworkers to annotate each chain. Workers were shown the question, correct answer, and reasoning chain, e.g.:

> Question: **What is formed by rivers flowing over rocks?**
> Answer: **soil**
> Because:
> fact 1: **Rivers erode the rocks they flow over**, *and*
> fact 2: **soil is formed by rocks eroding**

They were then asked if fact 1 and fact 2 *together* were a reasonable chain of reasoning for the answer, and to promote thought were offered several categories of "no" answer: fact 1 alone, or fact 2 alone, or either alone, justified the answer; or the answer was not justified; or the question/answer did not make sense. (Two "unsure" categories were also offered but rarely selected). The full instructions to the workers are provided in the Appendix. Each chain was annotated by 3 workers. To ensure quality, only AMT Masters level workers were used, and several checks were performed: First, for cases where at least two workers agreed, if a worker's annotations disagreed with the majority unreasonably often (from inspection, judged as more than 25% of the time), then the worker was (paid but then) blocked, and his/her annotations redone. Second, if a worker's distribution of labels among the six categories substantially deviated from other workers (e.g., almost always selecting the same category), or if his/her task completion time was unrealistically low, then his/her work was sampled and checked. If it was of low quality then he/she again was (paid and) blocked, and his/her annotations redone. Pairwise agreement was 74% (2 class) or 45% (for all six subclasses), with a Fleiss κ (inter-annotator agreement) of 0.37 ("fair agreement" (Landis and Koch, 1977)). There was a majority agreement (using all six subclasses) of 84%, again suggesting fair annotation quality. For the final dataset, we adopt a conservative approach and treat the no majority agreement cases as invalid chains. Summary statistics are in Table 1.

[2]https://allenai.org/data/qasc

3.4 eQASC-perturbed - Testing Robustness

For a test of robustness of model for reasoning chain explanation detection, we also created `eQASC-perturbed`, a dataset of valid eQASC reasoning chains, perturbed in a way so as to preserve their validity. To do this, we first randomly selected a subset of the valid reasoning chains from the test split of `eQASC-perturbed`. We then asked crowdworkers to modify the chains by replacing a word or phrase shared between at least two sentences with a different word or phrase, and to make sure that the resulting new chain remained valid. (e.g., *"amphibians"* became *"frogs"*, or *"eats other animals"* became *"consumes its prey"*). We collected **855 perturbed**, (still) valid reasoning chains in this way.

3.5 eOBQA - Testing Generalization

Finally, to further measure the generality of our model (without re-fine-tuning), we created a smaller set of annotations for a different dataset, namely OBQA (4-way multiple choice) (Mihaylov et al., 2018). The original dataset has no explanations and no associated corpus. Thus to generate explanations, we use sentences from the QASC corpus, and annotate the top two (for all test questions) formed by the retrieval step (Section 3.2). Note that for some questions, there may be no valid justification which can be formed from the corpus. We followed the same annotation protocol as for `eQASC` to have crowd workers annotate the chains (Section 3.3). The resulting dataset containing **998 annotated chains**, of which 9.5% were marked as valid reasoning explanations.

4 Learning to Score Chains

Our full approach to explaining an answer has two steps, namely candidate chain retrieval followed by chain scoring, to find the highest-ranked chain supporting an answer. For chain retrieval, we assume the same procedure described earlier to identify candidate chains. For chain scoring, we train a BERT-based model to distinguish valid chains from invalid ones, using the training data collected in the `eQASC` dataset, as we now describe.

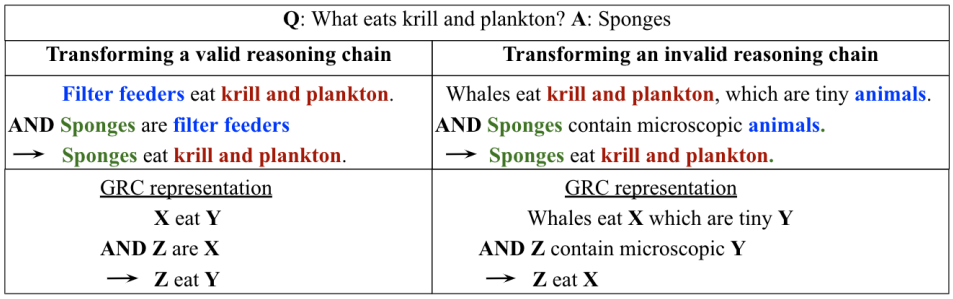

Figure 3: Generalized reasoning chains (GRCs) are formed by replacing repeated noun phrases with variables.

We evaluate using all three collected datasets. We also evaluate two different ways of presenting the chain to the model to score (both train and test): (a) in its original form (with Q+A flipped to a declarative sentence), (b) in a generalized form, where repeated noun phrases are variabilized. Our interest is how well a model can perform, both to assess practical use and as a baseline for further improvement; and how the two different chain representations impact performance.

4.1 Chain Representation

Declarative form For a chain to support the answer to a question, we construct H as a declarative form of the question + answer using standard QA2D tools, e.g., (Demszky et al., 2018). For example, for the question + answer "What can cause a forest fire? Static electricity", the hypothesis H to be entailed by C is "Static electricity can cause a forest fire.". An alternate representation for H is to simply append answer to the end of the question. We did not observe any significant change in the best dev split performances on switching to the alternate representation described above.

Generalized Reasoning Chains (GRC) : We observe that specific reasoning chains are often instantiations of more general patterns. For example, in Figure 1, the specific explanation can be seen as an instantiation of the more general pattern "X can cause Y" AND "Y can start Z" IMPLIES "X can cause Z". We refer to such patterns as *Generalized Reasoning Chains* (GRCs). To encourage our model to recognize valid and invalid chains at the pattern level, we explore the following strategy: First, we transform candidate chains into generalized chains (GRCs) by replacing repeated noun phrases with variables (special tokens), a process known as delexicalization (Suntwal et al., 2019). We then train and test the model using the GRC representation. We hypothesize that distinguishing a valid justification chain from an invalid one should not need typing information in most cases.

To identify the phrases to variabilize, (1) we first perform part-of-speech tagging on the sentences, and (2) extract candidate entities by identifying repeating nouns i.e. those which occur in at least two of the sentences in the chain (We stem the words before matching, and include any matching preceding determiners and adjectives into detected entities). e.g. 'the blue whale is a mammal' and 'the blue whale breathes..' leads to detection of 'the blue whale'). (3) Then, we assign a special token to each of the candidates, using a predefined set of unused tokens, which can be viewed as a set of variables. Some examples of GRCs are shown in Figure 3 and later in Figure 4, using `X, Y, Z` as the special token set (As our models are BERT-based, we use `unused`$i \in \{1, 2, ...\}$ to denote these tokens).

4.2 Model Training

To distinguish valid from invalid chains, we fine-tune a pre-trained BERT model (Devlin et al., 2019) for scoring the possible explanation chains. We encode a chain $f1$ AND $f2 \rightarrow H$ as:

[CLS] $f1$ [SEP] $f2$ [SEP] H

where [SEP] is a sentence boundary marker. Thereafter, we pass the chain through the BERT model (BERT-base-uncased). We employ a two layer feed-forward neural network with ReLU non-linearity, as a binary classifier on the pooled [CLS] representation to predict valid vs invalid reasoning chains. Model parameters are trained to minimize the binary cross entropy loss.

5 Experiments

For training, we use the annotated chains in the train split of `eQASC` alongwith the 'gold' chains provided in the QASC dataset (QSC gold chains

Model	Delexicalized Representation	Classification		Ranking	
		F1 (dev) test	AUC-ROC (dev) test	P@1 (dev) test	NDCG (dev) test
RETRIEVAL	n/a	(0.52) 0.50	(0.75) 0.74	(0.47) 0.47	(0.59) 0.60
BERT-QA	n/a	(0.44) 0.43	(0.52) 0.51	(0.47) 0.47	(0.48) 0.49
BERT-CHAIN	No	(0.68) **0.64**	(0.88) **0.87**	(0.57) **0.55**	(0.65) **0.65**
BERT-GRC	Yes	(0.63) **0.62**	(0.85) **0.85**	(0.55) **0.54**	(0.64) **0.64**
Performance upper-bound:		(1.00) 1.00	(1.00) 1.00	(0.76) 0.76	(0.76) 0.76

Table 2: The ability of models to identify valid explanations (classification) or rank the set of explanations for each answer (ranking), with best test results highlighted. BERT-GRC and BERT-CHAIN perform better than RETRIEVAL and BERT-QA methods, though fall short of the upper bound. Using the generalized chains (BERT-GRC) performs similarly to BERT-CHAIN, even though it is using less information (masking out overlapping noun phrases).

are always considered valid reasoning chains). We try two different ways of presenting chains to the model, namely the original and generalized chain representations (GRCs), thus produce two models that we refer to as **BERT-CHAIN** and **BERT-GRC** respectively. In earlier experiments, we did not find using chains for negative answer options (which are all invalid chains) to be useful (see Section 6.3), so we use chains for correct answer options only. We use AllenNLP (Gardner et al., 2018) toolkit to code our models.

We test on all the three proposed datasets. Since we are interested in finding explanations for the correct answer, we ignore the incorrect answer chains for the purpose of testing (they still accompany the dataset and can be used as additional training data since they are invalid reasoning chains by definition: Section 6.3). For `eQASC` and `eOBQA`, we evaluate in two ways: First, treating the task as classification, we measure F1 and AUC-ROC (below). Second, treating the task as ranking the *set* of explanations for each answer, we measure P@1 and Normalized Discounted Cumulative Gain (NDMC) (also below). We use the trained model's probability of a chain being valid to rank the retrieved candidate chains for a given question and answer.

5.1 Metrics

F1 and **AUC-ROC**: Viewing the task as classifying individual explanations, we report the area under the ROC (Receiver Operating Characteristics) curve, treating the valid explanation chains as the positive class. ROC curves are plots of true positive rate on the Y-axis against false positive rate on the X-axis. A larger area under the curve is better, with 1.0 being the best. Additionally, we report F1 for the positive class.

P@1 and **NDCG**: Viewing the task as ranking the *set* of explanations for each answer, P@1 measures the fraction of cases where the topmost ranked chain is a valid chain. This reflects the model's ability to find a valid explanation for an answer, given the retrieval module. Note that the upper bound for this measure is less than 1.0 for `eQASC`, as there are questions for which *none* of the candidate chains are valid (discussed shortly in Section 6.4). NDCG (Normalized Discounted Cumulative Gain) measures how well ranked the candidates are when ordered by score, and is a measure widely used in the learning-to-rank literature. Consider an ordered (as per decreasing score) list of N(=10) chains and corresponding labels $y_i \in \{0, 1\}; i \in 1, 2, .., N$, where $y_i = 1$ represents a valid chain. NDCG is defined per question (then averaged) as:

$$\text{NDCG} = \frac{1}{Z} \sum_{i=1}^{N} \frac{y_i}{\log_2(i+1)}$$

where Z is a normalization factor so that perfect ranking score (when all the valid chains are ranked above all the invalid chains) is 1. We define NDCG as 0 if there are no valid chains.

5.2 Baselines

We compare our model with two baselines, **RETRIEVAL** and **BERT-QA**. Recall that our method first collects the top M candidate chains, ordered by retrieval score (Section 3.2). Thus a simple baseline is to use that retrieval score itself as a measure of chain validity. This is the **RETRIEVAL** baseline.

We also consider a baseline, **BERT-QA**, by adapting the approach of Perez et al. (2019) to our task. In the original work, given a passage of text and a multiple choice question, the system identifies the sentence(s) S that are the most convincing evidence for a given answer option a_i. To do this, it iteratively finds the sentence that most increases the probability of a_i when added to an (initially empty) pool of evidence, using a QA system originally trained on the entire passage. In other words, the probability increase is used as a measure of how

Original chain	Edited chain	BERT -CHAIN	BERT -GRC
tadpole changes into a frog **AND** the frog is a totem of *metamorphosis* → tadpoles undergo *metamorphosis*	tadpole changes into a frog **AND** the frog is a totem of *transformation* → tadpoles undergo *transformation*	0.21	0.00
insects can spread disease and destroy crops **AND** food crops are produced for local consumption → insects can destroy food	insects can spread disease and decimate crops **AND** food crops are produced for local consumption → insects can decimate food	0.11	0.00

Table 3: Prediction Consistency: Examples from `eQASC-perturbed` with changes in probability score (of being a valid reasoning chain) for different methods. Here, BERT-GRC has (desirably) not changed its score due to an immaterial perturbation, while BERT-CHAIN has, indicating greater stability for the GRC representation. This trend holds generally (Table 4).

Model	% cases with 0.0 change
BERT-CHAIN	0.23%
BERT-GRC	**40.80%**

Table 4: Given an immaterial perturbation to a reasoning chain, a model's predicted probability of validity should not change if it is making consistent predictions. We evaluate the absolute difference in probability scores of original and edited reasoning chains in `eQASC-perturbed`. We observe that for 40.8% and 0.23% of the cases did not show any change in score for BERT-GRC and BERT-GRC respectively. The results suggest that GRCs improve prediction consistency.

Model	P@1	AUC-ROC
RETRIEVAL	0.70	0.58
BERT-CHAIN	0.85	0.89
BERT-GRC	**0.89**	0.86

Table 5: Application of our (`eQASC`-trained) model to a new dataset `eOBQA`. The high AUC-ROC figure suggests the model remains good at distinguishing valid from invalid chains. We report P@1 only for the questions which have at least one valid chain, i.e., where ranking is meaningful.

convincing the evidence sentence is. We adapt this by instead finding the *chain* that most increases the probability of a_i (compared with an empty pool of evidence), using a QA system originally trained with all the candidate chains for a_i. For the QA system, we use the straightforward BERT-based model described in (Khot et al., 2020). We then use that increase in probability of the correct answer option, measured for each chain, as a measure of chain validity. We call this baseline **BERT-QA**.

5.3 Results: Performance on eQASC

The test results on the `eQASC` are shown in Table 2. There are several important findings:

1. The best performing versions of BERT-CHAIN and BERT-GRC significantly outperforms the baselines. In particular, the AUC-ROC is 11% higher (absolute), NDCG rises from 0.60 to 0.64, and P@1 rises from 0.47 to 0.54 for BERT-GRC, indicating substantial improvement.

2. The generalized chain representation does not lead to a significant reduction (nor gain) in performance, despite abstracting away some of the lexical details through variabilization. This suggests the abstracted representation is as good as the original, and may have some additional benefits (Section 5.4).

3. The BERT-QA baseline scores surprisingly low. A possible explanation is that, in the original setting, Perez et al. (2019)'s model learned to spot a (usually) single relevant sentence among a passage of irrelevant sentences. In our setting, though, all the chains are partially relevant, making it harder for the model to distinguish just one as central.

5.4 Results:Consistency in eQASC-perturbed

We posit that the generalized (GRC) chain representation may improve robustness to small changes in the chains, as the GRC abstracts away some of the lexical details. To evaluate this, we use the crowdworker-perturbed, (still) valid chains in `eQASC-perturbed`. As the perturbed chain often follows the same/similar reasoning as the original one, this test can be considered one of consistency: the model's prediction should stay same. To measure this, we record the model's predicted probability of a chain being valid, then compare these probabilities for each pair of original and perturbed chains. Ideally, if the model is consistent and the perturbations are immaterial, then these probabilities should not change.

The results are shown in Table 4. In a large fraction of the instances, generalized chain representation exhibits no change. This is perhaps expected given the design of the GRC representations. Thus,

X can cause Y AND Y can start Z → X can cause Z
X is used for Y AND Z are X → Z are used for Y
X are formed by Y AND Y are made of Z
→ X are formed by Z
X are Y AND Y are Z → X are Z
X produce Y AND Y is a Z → X produce Z
X increases Y AND X occurs as Z → Z increases Y
X changes Y AND Y is Z → X changes Z
X is Y AND X carries Z → Y carries Z
X changes an Y AND Z are examples of X → Z change an Y
X are formed by Y AND X are formed through Z
→ Y can cause Z
X changes a Y AND Z start most X → Z can change Y

Figure 4: Examples of some of the highest scoring generalized reasoning chains (GRCs) found in eQASC.

using GRC not only achieves similar performance (Table 2), but produces more consistent predictions for certain types of perturbations. Table 3 shows some examples.

5.5 Results: Generalization to eOBQA

We are also interested in the generality of the model, i.e., how well it can transfer to a new dataset with no explanation training data (i.e., the situation with most datasets). To measure this, we ran our (eQASC-trained) models on eOBQA, namely the annotated top-2 candidate chains for OBQA test questions, to see if the models can still detect valid chains in this new data.

The results are shown in Table 5, and again illustrate that the BERT trained models continue to significantly outperform the retrieval baseline. High $P@1$ scores suggest that model is able to score a valid reasoning as the highest among the candidate whenever there is at least one such valid chain. The high AUC-ROC suggests that the model is able to effectively distinguish valid from invalid chains.

6 Analysis and Discussions

6.1 GRC as Explicit Reasoning Rationale

A potentially useful by-product of GRCs is that the underling reasoning *patterns* are made explicit. For example, Figure 4 show some of the top-scoring GRCs. This may be useful for helping a user understand the rationale behind a chain, and a repository of high-scoring patterns may be useful as a knowledge resource in its own right. This direction is loosely related to certain prior works on inducing general semantic reasoning rules (such as Tsuchida et al. (2011) who propose a method that induces rules for semantic relations based on a set of seed relation instances.)

6.2 Error Analysis

However, the BERT-GRC model was not always able to correctly distinguish valid from invalid GRC explanations. To better understand why, we analyzed 100 scoring failures on eQASC (dev), looking at the top 50 chains (i.e., ranked as most valid by our model) that were in fact annotated as invalid (false positives, FP), and the bottom 50 chains (ranked most invalid) that were in fact marked valid (false negatives, FN). We observed four main sources of error:

1. Over-generalization: ($\approx 45\%$ of the FP cases, $\approx 40\%$ of FN cases). Some generalized reasoning chains are merely plausible rather than a deductive proof, meaning that their instantiations may be annotated as valid or invalid depending on the context. For example, for the GRC

X contains Y AND *Z are in X* → *Z are in Y*

its instantation may have been marked as valid in

Cells contain nuclei AND *Proteins are in cells* → *Proteins are in nuclei*

but not for

Smog contains ozone AND *Particulates are in smog* → *Particulates are in ozone*

(Ozone itself does not contain particulates). Here the context is important to the perception of validity, but has been lost in the generalized form.

2. Incorrect Declarative Form: (FP $\approx 20\%$, FN $\approx 30\%$). Sometimes the conversion from question + answer to a declarative form H goes wrong, eg

What do octopuses use ink to hide from? sharks

was converted to the nonsensical sentence

Octopuses do use sharks ink to hide from.

In these cases, the annotations on chains supporting the original answer do not meaningfully transfer to the declarative formulation. (Here, FP/FN are due to label rather than prediction errors).

3. Shared Entity Detection: (FP $\approx 10\%$, FN $\approx 10\%$) To detect and variabilize shared entities during GRC construction, we search for repeated noun phrases in the sentences. This operational definition of "shared entities" can sometimes make mistakes, for example sometimes shared entities may be missed, e.g., *frog* and *bullfrog*, or incorrectly equated due to stemming or wrong part of speech tagging, e.g., *organic* and *organism*. The resulting GRC may be thus wrong or not fully generalized, causing some errors.

4. Model Failures: (FP $\approx 25\%$, FN $\approx 10\%$) The remaining failures appear to be simply due

the model itself, representing incorrect generalization from the training data. Additional training data may help alleviate such problems.

Despite these, GRCs often abstract away irrelevant details, and may be worthy of further study in explanation research.

6.3 Chains for Negative Answer Options

We also investigated whether we could skip using the eQASC annotations completely, and instead simply use the single QASC gold chains as positives, and chains for wrong answers as negatives (a form of distant supervision). However, we observed that but the results were significantly worse. We also tried adding chains for wrong answers as additional negative examples to the full `eQASC` dataset. However, we observed that this did not significantly improve (or hurt) scores. One possible reason for this is that `eQASC` may already contain enough training signal. Another possible reason is that (invalid) chains for wrong answers may qualitatively differ in some way from invalid reasoning chains for right answers, thus this additional data does not provide reliable new signal.

6.4 Limitations of Retrieval

Our focus in this paper has been on recognizing valid chains of reasoning, assuming a retrieval step that retrieves a reasonable pool of candidates to start with (Section 3.2). However, the retrieval step itself is not perfect: For QASC, designed so that at least one valid chain always exists, the retrieved pool of 10 contains no valid chains for 24% of the questions (upper bound in Table 2), capping the overall system's performance. To gauge the performance of our model when coupled with an improved retrieval system, we ran an experiment where, at test time, we explicitly add the gold chain to the candidate pool if it does not get retrieved (and even if there is some other valid chain already in the pool). We find the P@1 score rises from 0.54 (Table 2) to 0.82 (upper bound is now 1.0). This indicates the model scoring algorithm is performing well, and that improving the retrieval system, e.g., by considering may more chains per question or modifying the search algorithm itself, is likely to have the biggest impact on improving the overall system. Note also that the corpus itself is an important component: finding valid chains requires the corpus to contain a broad diversity of general facts to build chains from, hence expanding/filtering the corpus itself is another avenue for improvement.

6.5 Future Directions

The main purpose of this dataset is to generate explanations as an end-goal in itself, rather than improve QA scores (we do not make any claims in terms of QA accuracy or ability to improve QA scores). Although much of NLP has focused on QA scores, more recent work has targeted explanation as an end-goal in itself, with ultimate benefits for tutoring, validation, and trust. Nonetheless, a useful future direction is exploring answer prediction and explanation prediction as joint goals, and perhaps they can benefit each other.

Additionally, in the current work we have explored only a sequence of two sentences as an explanation for the third. Extending the proposed approaches for longer chains is an important future direction. We have proposed a technique for reducing reasoning chains to abstract chains. This technique makes assumptions about being able to match overlapping words. A future extension could explore more robust techniques for identifying abstract chains which do not make such assumptions.

7 Summary and Conclusion

Explaining answers to multihop questions is important for understanding *why* an answer may be correct, but there is currently a dearth of suitable, annotated data. To address this, and promote progress in explanation, we contribute three new explanation datasets, including one with over 98k annotated reasoning chains - by far the largest repository of annotated, corpus-derived explanations to date. We also have shown this data can significantly improve explanation quality on both in-domain (QASC) and out-of-domain (OBQA) tasks. Finally, we have proposed and explored using a lightweight method to achieve a delexicalized representation of reasoning chains. While preserving explanation quality (despite removing details), this representation appears to be more robust to certain perturbations.

Acknowledgements We thank Ashish Sabharwal, Tushar Khot, Dirk Groeneveld, Taylor Berg-Kirkpatrick, and anonymous reviewers for useful comments and feedback. We thank Michal Guerquin for helping with the QA2D tool. This work was partly carried out when HJ was interning at AI2. HJ is funded in part by a Adobe Research Fellowship.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *ArXiv*, abs/1508.05326.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 9560–9572.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018*, pages 845–855. Association for Computational Linguistics.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, E. Lehman, Caiming Xiong, R. Socher, and Byron C. Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *ACL*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2018:1884–1895.

Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2019. RC-QED: evaluating natural language derivations in multi-hop reading comprehension. *CoRR*, abs/1910.04601.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *AAAI Conference on Artificial Intelligence 2020*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 5189–5197. AAAI Press.

J. R. Landis and G. G. Koch. 1977. The measurement of oberver agreement for categorical data. *Biometrics*, 33:159–174.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. 2019. Finding generalizable evidence by learning to convince q\&a models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2402–2411.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 4932–4942. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *ICLR*.

Sandeep Suntwal, Mithun Paul, Rebecca Sharp, and Mihai Surdeanu. 2019. On the importance of delexicalization for fact verification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3404–3409.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *NAACL-HLT*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

Masaaki Tsuchida, Kentaro Torisawa, Stijn De Saeger, Jong-Hoon Oh, Jun'ichi Kazama, Chikara Hashimoto, and Hayato Ohwada. 2011. Toward finding semantic relations not written in a single sentence: An inference method using auto-discovered

rules. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 902–910. The Association for Computer Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Qinyuan Ye, Xiaozhen Huang, and Xiang Ren. 2020. Teaching machine comprehension with compositional explanations. *ArXiv*, abs/2005.00806.

APPENDIX

A. Additional Implementation Details

- Optimizer: We use Adam optimizer with initial learning rate of 2e-5
- Number of params: $\sim$ 110M parameters (Bert-base uncased and classification layer)
- Hyper-parameters: We search over following options for hyperparameter (1) one layer vs two layer classifier (2) negative class weight ($(0.1, 0.2, ..., 0.9)$) (3) using negative option chains or not; for BERT-GRC as well as BERT-CHAIN. We perform model selection based on best dev split performance as per $P@1$.
- Best model configuration for BERT-Chain: negative class weight = 0.2; without using negative option chains; using a two layer classifier. Best model configuration for BERT-GRC: negative class weight = 0.3; without using negative option chains; using a two layer classifier.
- We have uploaded code at `https://github.com/harsh19/Reasoning-Chains-MultihopQA`

B. Instructions to Crowdworkers

Below are the instructions provided to the (Amazon Mechanical Turk) crowdworkers for chain annotation.

Instructions (click here to collapse/expand instructions)

As part of an artificial intelligence (AI) project, we are trying to teach the computer to *reason*. We are planning to eventually make the system available as a free, open source resource on the internet.

To measure how well the computer is reasoning, we are wanting to assess whether the computer can *explain* a (correct) answer to a question. The HIT here is to look at five computer-generated explanations for five answers, and assess whether the explanations seem reasonable or not.

Each explanation consists of two facts. A good explanation is one where the two facts combine or "chain" together to explain an answer in a *sensible* way. Or, in some cases, just *one* of the facts is sufficient to justify the answer. For example:

Question: **What is formed by rivers flowing over rocks?**
Answer: **soil**
Because:
fact 1: **Rivers erode the rocks they flow over**, *and*
fact 2: **soil is formed by rocks eroding**

Now **select a choice below:** Do fact 1 and fact 2 seem like a good explanation for the Answer to the Question?

- (●) Yes - fact 1 and fact 2 **together** seems like a reasonable chain of reasoning for the answer
- ( ) Yes - fact 1 alone is enough to justify the answer.
- ( ) Yes - fact 2 alone is enough to justify the answer.
- ( ) Yes - fact 1 alone, AND fact 2 alone, are both separately enough to justify the answer.
- ( ) No - the facts don't combine together to support the answer
- ( ) No - the question/answer itself is incorrect/doesn't make sense.
- ( ) Not quite - the facts combine, but an additional fact is needed:
  Additional fact: [ ]
- ( ) Unsure - this seems like a borderline case

NOTE: In this case, as *rivers* erode *rocks* (fact 1), and *eroding rocks* forms *soil* (fact 2), it follows that the answer to *What is formed by rivers flowing over rocks?* is *soil*. Thus fact 1 and fact 2 seem like a reasonable explanation for the given Answer (soil).

Some important notes:

- A good explanation is one with a reasonable chain of reasoning. Examples are below. Think of what would be a good explanation if you were explaining an Answer to a friend, or writing an explanation as part of an exam.
- Additionally, in some cases a single fact is enough to justify the Answer (i.e., the other fact is not needed). We'd like you to identify these caess also.
- A bad explanation is one where the facts don't combine into a sensible line of reasoning (e.g., the facts are irrelevant to the question, or don't arrive at the Answer)
- Ignore minor grammatical errors, e.g., typo's, extra words - so long as it's clear what the facts are saying.
- If the question/answer itself seems wrong or weird, seleect the "No - the question/answer itself is incorrect/doesn't make sense" option
- Note: a few questions are "complete the sentence" form, e.g., "Question: **Dogs are...** Answer: **mammals**"
- Feel free to use the Web for information, if that helps assess the explanations

- Thank you for your help!!

Now please read the examples below carefully!

The following are examples of **good** explanations ("Yes - fact 1 and fact 2 together seems like a reasonable explanation for the answer")

Question: **What are some invertebrates?** Answer: **insects**
Because:
fact 1: **Some examples of invertebrates are arthropods.**, *and*
fact 2: **Most arthropods are insects.**

Question: **Black objects...** Answer: **absorb sunlight**
fact 1: **Black objects are the best heat absorbers.**, *and*
fact 2: **absorbing sunlight causes objects to heat**

Question: **what is very deadly?** Answer: **the ground shaking**
Because:
fact 1: **Earthquakes Earthquakes are very deadly.**, *and*
fact 2: **an earthquake causes the ground to shake**

Question: **What can our ears detect?** Answer: **matter vibrating**
Because:
fact 1: **When the waves pass our ears, a sound is detected.**, *and*
fact 2: **matter vibrating can cause sound**

In all these cases, you can see a chain of reasoning from the question to the answer using the facts.

The following are examples of a **single fact** explanations ("Yes - fact 1/2 alone is enough to justify the answer"):

Question: **what do flowers attract?** Answer: **bees**
Because:
fact 1: **How flowers attract honey bees and why they do it.**, *and*
fact 2: **bees convert nectar into honey**
Here, fact 1 alone is enough to justify the answer that "flowers attract bees"

Question: **What has permeable skin?** Answer: **frogs**
Because:
fact 1: **skin is used for breathing air by frogs**, *and*
fact 2: **Frogs have permeable skin that both breathes and takes in water.**
Here, fact 2 alone is enough to justify the answer that "frogs have permeable skin"

The following is an example of a **bad question/answer** ("No - the question/answer itself is incorrect/doesn't make sense.")
Question: **What powers rockets?** Answer: **Mechanical energy**
Because:
fact 1: **Power is what they seek, and power is what they get.**, *and*
fact 2: **Most ecosystems get energy from sunlight.**
The answer seems wrong here - rockets are powered by chemical energy, not mechanical energy.

The following are examples of **bad** explanations ("No - the facts don't combine together to support the answer"):

Question: **What do bats do with seeds?** Answer: **spread**
Because:
fact 1: **Insects and bats do it.**, *and*
fact 2: **Insects spread disease and destroy crops.**
Here, neither fact seems relevant to the question.

Question: **What do amphibians easily absorb?** Answer: **chemicals**
Because:
fact 1: **Light, easily absorbed.**, *and*
fact 2: **a flashlight converts chemical energy into light energy**

Question: **What results from plucking a string?** Answer: **sound waves**
Because:
fact 1: **Plucked strings are another matter.**, *and*
fact 2: **matter vibrating can cause sound**

Question: **How is limestone formed?** Answer: **Deposition.**
Because:
fact 1: **Limestone is the rock formed by calcite.**, *and*
fact 2: **sedimentary rocks are formed by deposition**
The link between limestone and sedimentary rocks is unstated.

Question: **What powers sweat?** Answer: **The body's fuel**
Because:
fact 1: **Sweat glands produce sweat.**, *and*
fact 2: **when the body is hot , sweat is produced to cool the body**

The following is an example of an **additional fact being needed** ("Not quite - the facts combine, but an additional fact is needed"):
Question: **Snow leopards coats can be used for what?** Answer: **protection from the cold**
Because:
fact 1: **Snow leopards coats have thick, dense fur.**, *and*
fact 2: **thick fur protects animals in winter**
Here an additional fact that it is cold in winter is needed:
Additional fact: It is cold in winter.

Thank you for your help! You rock!