What-if I ask you to explain:
Explaining the effects of perturbations in procedural text

Dheeraj Rajagopal[1], Niket Tandon[2], Bhavana Dalvi[2], Peter Clark[2], Eduard Hovy[1]

[1]Carnegie Mellon University   [2]Allen Institute for Artificial Intelligence
{dheeraj,hovy}@cs.cmu.edu {niket,bhavanad,peterc}@allenai.org

Abstract

Our goal is to explain the effects of perturbations in procedural text, e.g., given a passage describing a rabbit's life cycle, explain why illness (the perturbation) may reduce the rabbit population (the effect). Although modern systems are able to solve the original *prediction* task well (e.g., illness results in less rabbits), the *explanation* task - identifying the causal chain of events from perturbation to effect - remains largely unaddressed, and is the goal of this research. We present QUARTET, a system that constructs such explanations from paragraphs, by modeling the explanation task as a multitask learning problem. QUARTET constructs explanations from the sentences in the procedural text, achieving $\sim$ 18 points better on explanation accuracy compared to several strong baselines on a recent process comprehension benchmark. On an end task on this benchmark, we show a surprising finding that good explanations do not have to come at the expense of end task performance, in fact leading to a 7% F1 improvement over SOTA.

Figure 1: Given a procedural text, the task is to explain the effect of the perturbation using the input sentences.

1 Introduction

Procedural text is common in natural language (in recipes, how-to guides, etc.) and finds many applications such as automatic execution of biology experiments (Mysore et al., 2019), cooking recipes (Bollini et al., 2012) and everyday activities (Yang and Nyberg, 2015). However, the goal of procedural text understanding in these settings remains a major challenge and requires two key abilities, (i) understanding the dynamics of the world *inside* a procedure by tracking entities and what events happen as the narrative unfolds. (ii) understanding the dynamics of the world *outside* the procedure that can influence the procedure.

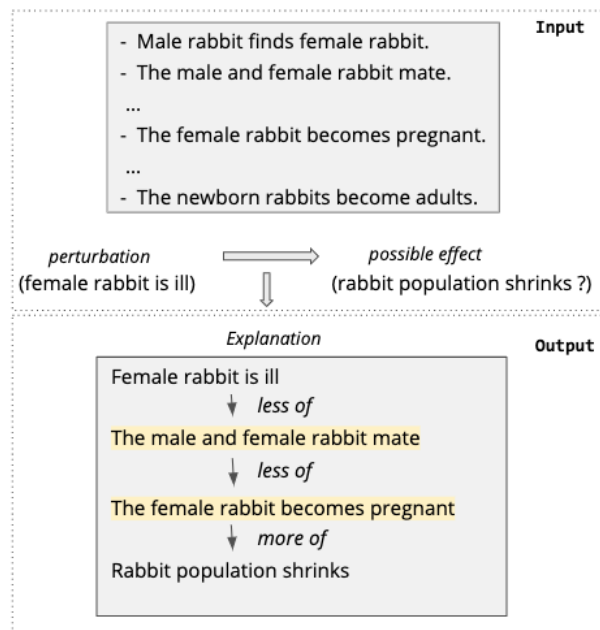While recent systems for procedural text comprehension have focused on understanding the dynamics of the world *inside* the process, such as tracking entities and answering questions about what events happen, e.g., (Tandon et al., 2018; Bosselut et al., 2018; Henaff et al., 2017), the extent to which they understand the influences of *outside* events remains unclear. In particular, if a system fully understands a process, it should be able to predict what would happen if it was perturbed in some way due to an event from the *outside* world. Such counterfactual reasoning is particularly challenging because, rather than asking what happened (described in text), it asks about what **would** happen in an alternative world where the change occurred.

Recently, Tandon et al. (2019) introduced the WIQA dataset that contains such problems, requiring prediction of the effect of perturbations in a procedural text. They also presented several strong models on this task. However, it is unclear whether those high scores indicate that the mod-

els fully understand the described procedures, i.e., that the models have knowledge of the causal chain from perturbation to effect. To test this, Tandon et al. (2019) also proposed an explanation task. While the general problem of synthesizing explanations is hard, they proposed a simplified version in which explanations were instead assembled from sentences in the input paragraph and qualitative indicators (more/less/unchanged). Although they introduced this explanation task and dataset, they did not present a model to address it. We fill this gap by proposing the first solution to this task.

We present a model, QUARTET (QUAlitative Reasoning wiTh ExplanaTions) that takes as input a passage and a perturbation, and its qualitative effect. The output contains the qualitative effect and an explanation structure over the passage. See Figure 1 for an example. The explanation structure includes up to two supporting sentences from the procedural text, together with the qualitative effect of the perturbation on the supporting sentences (*more of* or *less of* in Figure 1). QUARTET models this qualitative reasoning task as a multitask learning problem to explain the effect of a perturbation.

Our main contributions are:

- We present the first model that explains the effects of perturbations in procedural text. On a recent process comprehension benchmark, QUARTET generates better explanations compared to strong baselines.

- On an end task on this benchmark, we show a finding that good explanations do not have to come at the expense of end task performance, in fact leading to a 7% F1 improvement over SOTA. (refer §6). Prior work has found that optimizing for explanation can hurt end-task performance. Ours is a useful datapoint showing that good explanations do not have to come at the expense of end-task performance[1].

2 Related work

Procedural text understanding: Machine reading has seen tremendous progress. With machines reaching human performance in standard QA benchmarks (Devlin et al., 2018; Rajpurkar et al., 2016), more challenging datasets have been proposed (Dua et al., 2019) that require background knowledge, commonsense reasoning (Talmor et al., 2019) and visual reasoning (Antol et al., 2015;

Zellers et al., 2018). In the context of procedural text understanding which has gained considerable amount of attention recently, (Bosselut et al., 2018; Henaff et al., 2017; Dalvi et al., 2018) address the task of tracking entity states throughout the text. Recently, (Tandon et al., 2019) introduced the WIQA task to predict the effect of *perturbations*.

Understanding the effects of perturbations, specifically, qualitative change, has been studied using formal frameworks in the qualitative reasoning community (Forbus, 1984; Weld and De Kleer, 2013) and counterfactual reasoning in the logic community (Lewis, 2013). The WIQA dataset situates this task in terms of natural language rather than formal reasoning, by treating the task as a mixture of reading comprehension and commonsense reasoning. However, existing models do not explain the effects of perturbations.

Explanations: Despite large-scale QA benchmarks, high scores do not necessarily reflect understanding (Min et al., 2019). Current models may not be robust or exploit annotation artifacts (Gururangan et al., 2018). This makes explanations desirable for interpretation (Selvaraju et al., 2017).

Attention based explanation has been successfully used in vision tasks such as object detection (Petsiuk et al., 2018) because pixel information is explainable to humans. These and other token level attention models used in NLP tasks (Wiegreffe and Pinter, 2019) do not provide full-sentence explanations of a model's decisions.

Recently, several datasets with natural language explanations have been introduced, e.g., in natural language inference (Camburu et al., 2018), visual question answering (Park et al., 2018), and multi-hop reading comprehension (HotpotQA dataset) (Yang et al., 2018). In contrast to these datasets, we explain the effects of perturbations in procedural text. HotpotQA contains explanations based on two sentences from a Wikipedia paragraph. Models on the HotpotQA would not be directly applicable to our task and require substantial modification for the following reasons: (i) HotpotQA models are not trained to predict the qualitative structure (more or less of chosen explanation sentences in Figure 1). (ii) HotpotQA involves reasoning over named entities, whereas the current task focuses on common nouns and actions (models that work well on named entities need to be adapted to common nouns and actions (Sedghi and Sabharwal, 2018)). (iii) explanation paragraphs in HotpotQA are not

---

ears less protected → (MORE/+) sound enters the ear → (MORE/+) sound hits ear drum → (MORE/+) more sound detected
blood clotting disorder → (LESS/−) blood clots → (LESS/−) scab forms → (MORE/+) less scab formation
breathing exercise → (MORE/+) air enters lungs → (MORE/+) air enters windpipe → (MORE/+) oxygen enters bloodstream
squirrels store food → (MORE/+) squirrels eat more → (MORE/+) squirrels gain weight → (MORE/+) hard survival in winter
less trucks run → (LESS/−) trucks go to refineries → (LESS/−) trucks carry oil → (MORE/+) less fuel in gas stations
coal is expensive → (LESS/−) coal burns → (LESS/−) heat produced from coal → (LESS/−) electricity produced
legible address → (MORE/+) mailman reads address → (MORE/+) mail reaches destination → (MORE/+) on-time delivery
more water to roots → (MORE/+) root attract water → MORE/+) roots suck up water → (LESS/−) plants malnourished
in a quiet place → (LESS/−) sound enters the ear → (LESS/−) sound hits ear drum → (LESS/−) more sound detected
eagle hungry → (MORE/+) eagle swoops down → (MORE/+) eagle catches mouse → (MORE/+) eagle gets more food

Table 1: Examples of our model's predictions on the dev. set in the format: "$q_p \rightarrow d_i \ x_i \rightarrow d_j \ x_j \rightarrow d_e \ q_e$". Supporting sentences x_i, x_j are compressed e.g., "the person has his ears less protected" → "ears less protected"

procedural while the current input is procedural in nature with a specific chronological structure.

Another line of work provides more structure and organization to explanations, e.g., using scene graphs in computer vision (Ghosh et al., 2019). For elementary science questions, Jansen et al. (2018) uses a science knowledge graph. These approaches rely on a knowledge structure or graph but knowledge graphs are incomplete and costly to construct for every domain (Weikum and Theobald, 2010). There are trade-offs between unstructured and structured explanations. Unstructured explanations are available abundantly while structured explanations need to be constructed and hence are less scalable (Camburu et al., 2018). Generating free-form (unstructured) explanations is difficult to evaluate (Cui et al., 2018; Zhang et al., 2019), and adding qualitative structure over them is non-trivial. Taking a middle ground between free-form and knowledge graphs based explanations, we infer a qualitative structure over the sentences in the paragraph. This retains the rich interpretability and simpler evaluation of structured explanations as well as leverages the large-scale availability of sentences required for these explanation.

It is an open research problem whether requiring explanation helps or hurts the original task being explained. On the natural language inference task (e-SNLI), Camburu et al. (2018) observed that models generate correct explanations *at the expense of* good performance. On the Cos-E task, recently Rajani et al. (2019) showed that explanations help the end-task. Our work extends along this line in a new task setting that involves perturbations and enriches natural language explanations with qualitative structure.

3 Problem definition

We adopt the problem definition described in Tandon et al. (2019), and summarize it here.

Input: 1. Procedural text with steps $x_1 \ldots x_K$. Here, x_k denotes step k (i.e., a sentence) in a procedural text comprising K steps.
2. A perturbation q_p to the procedural text and its likely candidate effect q_e.

Output: An explanation structure that explains the effect of the perturbation q_p:

$$q_p \rightarrow d_i x_i \rightarrow d_j x_j \rightarrow d_e q_e$$

- i: step id for the first supporting sentence.
- j: step id for the second supporting sentence.
- $d_i \in \{+ - \cdot\}$: how step id i is affected.
- $d_j \in \{+ - \cdot\}$: how step id j is affected.
- $d_e \in \{+ - \cdot\}$: how q_e is affected.

See Figure 1 for an example of the task, and Table 1 for examples of explanations.

An explanation consists of up to two (i.e., zero, one or two) supporting sentences i, j along with their qualitative directions d_i, d_j. If there is only one supporting sentence, then $j = i$. If $d_e = \cdot$, then $i = \varnothing$, $j = \varnothing$ (there is no valid explanation for no-effect).

While there can be potentially many correct explanation paths in a passage, the WIQA dataset consists of only one gold explanation considered best by human annotators. Our task is to predict that particular gold explanation.

Assumptions: In a procedural text, steps $x_1 \ldots x_K$ are chronologically ordered and have a forward flowing effect i.e., if $j > i$ then more/increase of x_i will result in more/increase of x_j. Prior work on procedural text makes a similar assumption (Dalvi et al., 2018). Note that

this assumption does not hold for cyclic processes, and cyclic processes have already been flattened in WIQA dataset. We make the following observations based on this *forward-flow* assumption.

a1: $i <= j$ (*forward-flow* order)

a2: $d_j = d_i$ (*forward-flow* assumption)[2]

a3: For the WIQA task, d_e is the answer label because it is the end node in the explanation structure.

a4: If $d_i = \cdot$ then answer label $= \cdot$ (since q_p does not affect q_e, there is no valid explanation.)

a5: $1 \le i \le K$; if $d_i = \cdot$, then i $= \emptyset$ (see a4)

a6: $i \le j \le K$; if $d_e = \cdot$, then j $= \emptyset$ (see a4)

This assumption reduces the number of predictions, removing d_j and answer label (see a2, a3). Given $x_1 \ldots x_K, q_p, q_e$ the model must predict four labels: i, j, d_i, d_e.

4 QUARTET model

We can solve the problem as a classification task, predicting four labels: i, j, d_i, d_e. If these predictions are performed independently, it requires several independent classifications and this can cause error propagation: prediction errors that are made in the initial stages cannot be fixed and can propagate into larger errors later on (Goldberg, 2017).

To avoid this, QUARTET predicts and explains the effect of q_p as a multitask learning problem, where the representation layer is shared across different tasks. We apply the widely used parameter sharing approach, where a single representation layer is followed by task specific output layers (Baxter, 1997). This reduces the risk of overfitting to a single task and allows decisions on i, j, d_i, d_e to influence each other in the hidden layers of the network. We first describe our encoder and then the other layers on top, see Figure 2 for the model architecture.

Encoder: To encode $x_1 \ldots x_K$ and question q we use the BERT architecture (Devlin et al., 2018) that has achieved state-of-the-art performance across several NLP tasks (Clark et al., 2019),

where the question $q = q_p \oplus q_e$ ($\oplus$ stands for concatenation). We start with a byte-pair tokenization (Sennrich et al., 2015) of the concatenated passage and question ($x_1 \ldots x_K \oplus q$) . Let $[x_k]$ denote the byte-pair tokens of sentence x_k. The text is encoded as `[CLS]` $[x_1]$ $[unused1]$ `[SEP]` $[x_2]$ $[unused2]$ `[SEP]` `..` $[q]$ `[SEP]`. Here, `[CLS]` indicates a special classification token. `[SEP]` and $[unused1..K]$ are special next sentence prediction tokens.

These byte-pair tokens are passed through a 12-layered Transformer network, resulting in a contextualized representation for every byte-pair token. In this contextualized representation, the vector $\mathbf{u} = [\mathbf{u_1}, ... \mathbf{u_K}, \mathbf{u_q}]$ where $\mathbf{u_k}$ denotes the encoding for $[x_k]$, and $\mathbf{u_q}$ denotes question encoding. Let E^l be the embedding size resulting from l^{th} transformer layer. In that l^{th} layer, $[\mathbf{u_1}, ... \mathbf{u_K}] \in \mathbb{R}^{K*E^l}$. The hidden representation of all transformer layers are initialized with weights from a self-supervised pre-training phase, in line with contemporary research that uses pre-trained language models (Devlin et al., 2018).

To compute the final logits, we add a linear layer over the different transformer layers in BERT that are individual winners for individual tasks in our multitask problem. For instance, out of the total 12 transformer layers, lower layers (layer 2) are the best predictors for $[i, j]$ while upper layers (layer 10 and 11) are the best performing predictors for $[d_i, d_e]$. Zhang et al. (2019) found that the last layer is not necessarily the best performing layer. Different layers seem to learn complementary information because their fusion helps. Combining different layers by weighted averaging of the layers has been attempted with mixed success (Zhang et al., 2019; Clark et al., 2019). We observed the same trend for simple weighted transformation. However, we found that learning a linear layer over concatenated features from winning layers improves performance. This is probably because there is very different information encoded in a particular dimension across different layers, and the concatenation preserves it better than simple weighted averaging.

Classification tasks: To predict the first supporting sentence x_i, we obtain a softmax distribution $s_i \in \mathbb{R}^K$ over $[\mathbf{u_1}, ... \mathbf{u_K}]$. From the *forward-flow* assumption made in the problem definition section earlier, we know that $i \le j$, making it possible to model this as a span prediction $x_{i:j}$. Inline with standard span based prediction models (Seo et al.,

---

[2]Note that this does not assume all sentences have the same directionality of influence. For example, a paragraph could include both positive and negative influences: "Predators arrive. Thus the rabbit population falls...". Rather, the $d_j = d_i$ assumption is one of narrative coherence: the *more* predators arrive, the *more* the rabbit population falls. That is, within a paragraph, we assume enhancing one step will have enhanced effects (both positive or negative effects) on future steps - a property of a coherently authored paragraph.
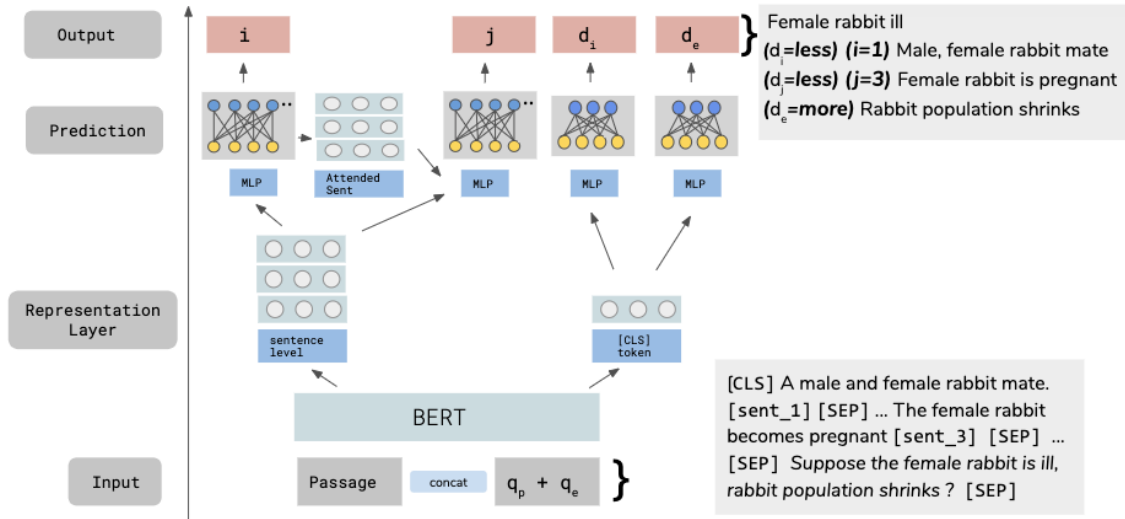
Figure 2: QUARTET model. *Input*: Concatenated passage and question using standard BERT word-piece tokenization. *Representation Layer*: The input is encoded using BERT transformer. We obtain `[CLS]` and sentence level representations. *Prediction*: From the sentence level representation, we use an MLP to model the distributions for i and j (using attended sentence representation). From `[CLS]` representation, we use MLP for d_i (and d_j, since $d_i = d_j$) and d_e distributions. *Output*: Softmax to predict $\{i, j, d_i, d_j, d_e\}$

2017), we use an attended sentence representation $(s_i \odot [\mathbf{u_1}, ...\mathbf{u_K}]) \oplus ([\mathbf{u_1}, ...\mathbf{u_K}]) \in \mathbb{R}^{K*2E^l}$ to predict a softmax distribution $s_j \in \mathbb{R}^K$ to obtain x_j. Here, $\odot$ denotes element-wise multiplication and $\oplus$ denotes concatenation.

For classification of d_i (and d_j, since $d_i = d_j$), we use the representation of the first token (i.e., `CLS` token $\in \mathbb{R}^{E^l}$) and a linear layer followed by softmax to predict $d_i \in \{ + - \cdot \}$. Classification of d_e is performed in exactly the same manner.

The network is trained end-to-end to minimize the sum of cross-entropy losses for the individual classification tasks i, j, d_i, d_e. At prediction time, we leverage assumptions (a4, a5, a6) to generate consistent predictions.

5 Experiments

Dataset: We train and evaluate QUARTET on the recently published WIQA dataset [3] comprising of 30,099 questions from 2107 paragraphs with explanations (23K train, 5K dev, 2.5K test). The perturbations q_p are either linguistic variation (17% examples) of a passage sentence (these are called in-para questions) or require commonsense reasoning to connect to a passage sentence (41% examples) (called, out-of-para questions). Explanations are supported by up to two sentences from the pas-

sage: 52.7% length 2, 5.5% length 1, 41.8% length 0. Length zero explanations indicate that $d_e = \cdot$ (called, no-effect questions), and ensure that random guessing on explanations gets low score on the end task.

Metrics: We evaluate on both explainability and the downstream end task (QA). For explainability, we define explanation accuracy as the average accuracy of the four components of the explanation: $acc_{expl} = \frac{1}{4} * \sum_{i \in \{i, j, d_i, d_e\}} acc(i)$ and $acc_{qa} = acc(d_e)$ (by assumption a3). The QA task is measured in terms of accuracy.

Hyperparameters: QUARTET fine-tunes BERT, allowing us to re-use the same hyperparameters as BERT with small adjustments in the recommended range (Devlin et al., 2018). We use the BERT-base-uncased version with a hidden size of 768. We use the standard adam optimizer with a learning rate 1e-05, weight decay 0.01, and dropout 0.2 across all the layers[4]. All the models are trained on an NVIDIA V-100 GPU.

Models: We measure the performance of the following baselines (two non-neural and three neural).
• RANDOM: Randomly predicts one of the three labels $\{+ - \cdot\}$ to guess $[d_i, d_e]$. Supporting sentences i and j are picked randomly from $|avg_{sent}|$

---

[3]WIQA dataset link: http://data.allenai.org/wiqa/

[4]Hyperparameter search details in appendix §9.1

sentences.

- MAJORITY: Predicts the most frequent label (*no effect i.e.* $d_e = \cdot$ in the case of WIQA dataset.)
- q_eONLY : Inspired by existing works (Gururangan et al., 2018), this baseline exploits annotation artifacts (if any) in the explanation dataset by retraining QUARTET using only q_e while hiding the permutation q_p in the question.
- HUMAN upper bound (Krippendorff's alpha inter-annotator values on $[i, j, d_i]$) on explainability reported in (Tandon et al., 2019)[5].
- TAGGING: We can reduce our task to a structured prediction task. An explanation i, j, d_i, d_e requires span prediction $x_{i:j}$ and labels on that span. So, for example, the explanation $i = 1, j = 2, d_i = +, d_j = -$ for input $x_1 \cdot x_5$ can be expressed as a tag sequence: `B-CORRECT E-OPPOSITE O O O`. Explanation $i = 2, j = 4, d_i = +, d_j = -$ would be expressed as: `O B-CORRECT I-CORRECT E-OPPOSITE O`. When $d_e = \cdot$, then the tag sequence will `O O O O O`. This BIEO tagging scheme has seven labels $T = \{$`B-CORRECT, I-CORRECT, B-OPPOSITE, I-OPPOSITE, E-CORRECT, E-OPPOSITE, O`$\}$. Formulating as a sequence tagging task allows us to use any standard sequence tagging model such as CRF as baseline. The decoder invalidates sequences that violate assumptions (a3 - a6). To make the encoder strong and yet comparable to our model, we use exactly the same BERT encoder as QUARTET. For each sentence representation u_k, we predict a tag $\in T$. A CRF over these local predictions additionally provides global consistency. The model is trained end-to-end by minimizing the negative log likelihood from the CRF layer.
- BERT-NO-EXPL: State-of-the-art BERT model (Tandon et al., 2019) that only predicts the final answer d_e, but cannot predict the explanation.
- BERT-W/-EXPL: A standard BERT based approach to the explanation task that predicts the explanation structure. This model minimizes only the cross-entropy loss of the final answer d_e, predicting an explanation that provides the best answer accuracy.
- DATAAUG: This baseline is adapted from Asai and Hajishirzi (2020), where a RoBERTa model is augmented with symbolic knowledge and uses an additional consistency-based regularizer. Com-

pared to our model, this approach uses a more robustly pre-trained BERT (RoBERTa) with data-augmentation optimized for QA Accuracy.
- QUARTET: our model described in §4 that optimizes for the best explanation structure.

5.1 Explanation accuracy

QUARTET is also the best model on explanation accuracy. Table 2 shows the performance on $[i, j, d_i, d_e]$. QUARTET also outperforms baselines on every component of the explanation. QUARTET performs better at predicting i than j. This trend correlates with human performance- picking on the second supporting sentence is harder because in a procedural text neighboring steps can have similar effects.

We found that the explanation dataset does not contain substantial annotation artifacts for the q_eONLY model to leverage (q_eONLY < MAJORITY)

Table 1 presents canonical examples of QUARTET dev predictions.

	acc_i	acc_j	acc_{d_i}	acc_{d_e}	acc_{expl}
RANDOM	12.50	12.50	33.33	33.33	22.91
q_eONLY	32.77	32.77	33.50	44.82	36.00
MAJORITY	41.80	41.80	41.80	41.80	41.80
TAGGING	42.26	37.03	56.74	58.34	48.59
BERT-W/-EXPL	38.66	38.66	69.20	75.06	55.40
QUARTET	**69.24**	**65.97**	**75.92**	**82.07**	**73.30**
HUMAN	75.90	66.10	88.20	96.30	81.63

Table 2: Accuracy of the explanation structure (i, j, d_i, d_e). Overall explanation accuracy is acc_{expl}. (Note that BERT-NO-EXPL and DATAAUG do not produce explanations).

We also tried a simple bag of words and embedding vector based alignment between q_p and x_i in order to pick the most similar x_i. These baselines perform worse than random, showing that aligning q_p and x_i involves commonsense reasoning that the these models cannot address.

6 Downstream Task

In this section, we investigate whether a good explanation structure leads to better end-task performance. QUARTET advocates explanations as a first class citizen from which an answer can be derived.

6.1 Accuracy on a QA task

We compare against the existing SOTA on WIQA no-explanation task. Table 3 shows that QUARTET improves over the previous SOTA BERT-NO-EXPL by 7%, achieving a new SOTA results. Both these

---

models are trained on the same dataset[6]. The major difference between BERT-NO-EXPL and QUARTET is that BERT-NO-EXPL solves only the QA task, whereas QUARTET solves explanations, and the answer to the QA task is derived from the explanation. Multi-tasking (i.e., explaining the answer) provides the gains to QUARTET.

	QA accuracy
RANDOM	33.33
MAJORITY	41.80
q_eONLY	44.82
TAGGING	58.34
BERT-NO-EXPL	75.19
BERT-W/-EXPL	75.06
DATAAUG	78.50
QUARTET	**82.07**
HUMAN	96.30

Table 3: QUARTET improves accuracy on the QA (end task) by 7% points.

All the models get strong improvements over RANDOM and MAJORITY. The least performing model is TAGGING. The space of possible sequences of correct labels is large, and we believe that the current training data is sparse, so a larger training data might help. QUARTET avoids this sparsity problem because rather than a sequence it learns on four separate explanation components.

Table 4 presents the accuracy based on question types. QUARTET achieves large gains over BERT-NO-EXPL on the most challenging out-of-para questions. This suggests that QUARTET improves the alignment of q_p and x_i that involves some common-sense reasoning.

6.2 Correlation between QA and Explanation

QUARTET not only improves QA accuracy but also the explanation accuracy. We find that QA accuracy (acc_{d_e} in Table 2) is positively correlated (Pearson coeff. 0.98) with explanation accuracy (acc_{expl}). This shows that if a model is optimized for explanations, it leads to better performance on end-task. Thus, with this result we establish that (at least on

---

Model	in-para	out-of para	no-effect	overall
RANDOM	33.33	33.33	33.33	33.33
MAJORITY	00.00	00.00	100.0	41.80
q_eONLY	20.38	20.85	78.41	44.82
BERT-NO-EXPL	71.40	53.56	90.04	75.19
BERT-W/-EXPL	72.83	58.54	92.03	75.06
QUARTET	**73.49**	**65.65**	**95.30**	**82.07**

Table 4: QUARTET improves accuracy over SOTA BERT-NO-EXPL across question types.

our task) models can make better predictions when forced to generate a sensible *explanation structure*. An educational psychology study (Dunlosky et al., 2013) hypothesizes that student performance improves when they are asked to explain while learning. However, their hypothesis is not conclusively validated due to lack of evidence. Results in Table 2 hint that, at least on our task, machines that learn to explain, ace the end task.

7 Error analysis

We analyze our model's errors (marked in red) over the dev set, and observe the following phenomena.

1. Multiple explanations: As mentioned in Section 3, more than one explanations can be correct. 22% of the incorrect explanations were reasonable, suggesting that overall explanation accuracy scores might under-estimate the explanation quality. The following example illustrates that while `gathering firewood` is appropriate when `fire is needed for survival`, one can argue that `going to wilderness` is less precise but possibly correct.

Gold: need fire for survival → (MORE/+) gather firewood → (MORE/+) build fire for warmth → (MORE/+) extensive camping trip
Pred: need fire for survival → (MORE/+) go to wilderness → (MORE/+) build fire for warmth → (MORE/+) extensive camping trip

2. i, j errors: Fig. 3 shows that predicted and gold distributions of i and j are similar. Here, sentence id $= -1$ indicates no effect. The model has learned from the data to never predict $j < i$ without any hard constraints.

The model is generally good at predicting i, j and in many cases when the model errs, the explanation seems plausible. Perhaps for the same underlying reason, human upper bound is not high on i (75.9%) and on j (66.1%). We show an example where i, j are incorrectly predicted (in red), but
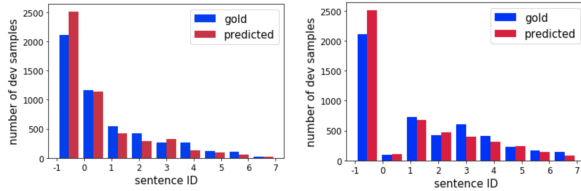
Figure 3: Gold vs. predicted distribution of i & j resp.

sound plausible.

Gold: ear	is	not	clogged	by	infection	→
(OPP/−)	sound hits ear		→		(OPP/−)	
electrical impulse reaches brain	→	(OPP/−)	more			
sound detected						
Pred: ear	is	not	clogged	by	infection	→
(OPP/−)	sound hits ear		→		(OPP/−)	
drum converts sound to electrical impulse	→	(OPP/−)				
more sound detected						

3. d_i, d_e errors: When the model incorrectly predicts d_i, a major source of error is when '−' is misclassified. 70% of the '−' mistakes, should have been classified as '+'. A similar trend is observed for d_e but the misclassification of '− is less skewed. Table 5 shows the confusion matrix of d_i and of d_e *in* { + − • } .

	•	+	−		•	+	−
•	1972	91	47	•	1972	89	49
+	295	883	358	+	261	909	295
−	226	492	639	−	252	346	830

Table 5: Confusion matrix for d_i (left) and d_e overall (right). (gold is on x-axis, predicted on y-axis.)

The following example shows an instance where '−' is misclassified as '+'. It implies that there is more scope for improvement here.

Gold: less	seeds	fall	to	the	ground	→
(OPP/−)	seed falls to the ground		→		(OPP/−)	
seeds germinate	→	(MORE/+)	fewer plants			
Pred: less	seeds	fall	to	the	ground	→
(OPP/−)	seed falls to the ground		→		(OPP/−)	
seeds germinate	→	(OPP/−)	fewer plants			

4. in-para vs. out-of-para: The model performs better on in-para questions (typically, linguistic variations) than out-of-para questions (typically, commonsense reasoning). Also see empirical evidence of this in Table 4.

The model is challenged by questions involving commonsense reasoning, especially to connect q_p with x_i in out-of-para questions. For example, in the following passage, the model incorrectly

predicts • (no effect) because it fails to draw a connection between `sleep` and `noise`:

Pack up your camping gear, food. Drive to your campsite. Set up your tent. Start a fire in the fire pit. Cook your food in the fire. Put the fire out when you are finished. Go to sleep. Wake up ...
q_p: less noise from outside
q_e: you will have more energy

Analogous to i and j, the model also makes more errors between labels '+' and '−' in out-of-para questions compared to in-para questions (39.4% vs 29.7%) – see Table 6.

	•	+	−		•	+	−
+	29	295	78	+	266	588	280
−	49	130	259	−	177	362	380

Table 6: Confusion matrix d_i for in-para & out-of-para

(Tandon et al., 2019) discuss that some in-para questions may involve commonsense reasoning similar to out-of-para questions. The following is an example of an in-para question where the model fails to predict d_i correctly because it cannot find the connection between `protected ears` and `amount of sound entering`.

Gold: ears less protected → (MORE/+) sound enters ear
→ (MORE/+) sound hits ear drum → (MORE/+)
more sound detected
Pred: ears less protected → (OPP/−)
sound enters the ear → (OPP/−) sound hits ear drum
→ (MORE/+) more sound detected

5. Injecting background knowledge: To study whether additional background knowledge can improve the model, we revisit the out-of-para question that the model failed on. The model fails to draw a connection between `sleep` and `noise`, leading to an incorrect (no effect) '•' prediction.

By adding the following relevant background knowledge sentence to the paragraph "`sleep requires quietness and less noise`", the model was able to correctly change probability mass from $d_e = $ '•' to '+'. This shows that providing commonsense through Web paragraphs and sentences is a useful direction.

Pack up your camping gear, food ... ***Sleeping requires quietness and less noise.*** Go to sleep. Wake up ...
q_p: less noise from outside
q_e: you will have more energy

8   Assumptions and Generality

QUARTET makes two simplifying assumptions: (1) explanations are assembled from the provided sentences (question + context), rather than generated, and (2) explanations are chains of qualitative, causal influences, describing how an end-state is influenced by a perturbation. Although these (helpfully) bound this work, the scope of our solution is still quite general: Assumption (1) is a common approach in other work on multihop explanation (e.g., HotpotQA), where authoritative sentences support an answer. In our case, we are the first to apply the same idea to chains of influences. Assumption (2) bounds QUARTET to explaining the effects of qualitative, causal influences. However, this still covers a large class of problems, given the importance of causal and qualitative reasoning in AI. The WIQA dataset provides the first large-scale dataset that exemplifies this class: given a qualitative influence, assemble a causal chain of events leading to a qualitative outcome. Thus QUARTET offers a general solution within this class, as well as a specific demonstration on a particular dataset.

9   Conclusion

Explaining the effects of a perturbation is critical, and we have presented the first system that can do this reliably. QUARTET not only predicts meaningful explanations, but also achieves a new state-of-the-art on the end-task itself, leading to an interesting finding that models can make better predictions when forced to explain. Our work opens up new directions for future research: 1) Can additional background context from the Web improve explainable reasoning? 2) Can such structured explanations be applied to other NLP tasks? We look forward to future progress in this area.

Acknowledgements

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. *ICCV*.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5642–5650, Online. Association for Computational Linguistics.

Jonathan Baxter. 1997. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39.

Mario Bollini, Stefanie Tellex, Tyler Thompson, Nicholas Roy, and Daniela Rus. 2012. Interpreting and executing recipes with a cooking robot. In *ISER*.

Antoine Bosselut, Omer Levy, Ari Holtzman, Corin Ennis, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. *ICLR*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *NeurIPS*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Yin Cui, Guandao Yang, Andreas Veit, Xun Huang, and Serge J. Belongie. 2018. Learning to evaluate image captioning. *CVPR*.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: A challenge dataset and models for process comprehension. *NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *NAACL*.

John Dunlosky, Katherine A. Rawson, Elizabeth Marsh, Mitchell J. Nathan, and Daniel T. Willingham. 2013. Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological science*.

Kenneth D. Forbus. 1984. Qualitative process theory. *Artificial Intelligence*, 24:85–168.

Shalini Ghosh, Giedrius Burachas, Arijit Ray, and Avi Ziskind. 2019. Generating natural language explanations for vqa using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on HLT*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*, pages 107–112.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. In *ICLR*.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In *LREC 2018*.

David Lewis. 2013. *Counterfactuals*. John Wiley & Sons.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. *arXiv*.

Sheshera Mysore, Zach Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. *arXiv preprint arXiv:1905.06939*.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. *CVPR*.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *ACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Hanie Sedghi and Ashish Sabharwal. 2018. Knowledge completion for generics using guided tensor factorization. *ACL*.

Ramprasaath Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ICLR*.

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

Niket Tandon, Bhavana Dalvi Mishra, Joel Grus, Wentau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. *EMNLP*.

Niket Tandon, Bhavana Dalvi Mishra, Keisuke Sakaguchi, Antoine Bosselut, and Peter Clark. 2019. Wiqa: A dataset for "what if..." reasoning over procedural text. *EMNLP*.

Gerhard Weikum and Martin Theobald. 2010. From information to knowledge: harvesting entities and relationships from web sources. In *PODS*.

Daniel S Weld and Johan De Kleer. 2013. *Readings in qualitative reasoning about physical systems*. Morgan Kaufmann.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

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. In *EMNLP*.

Zi Yang and Eric Nyberg. 2015. Leveraging procedural knowledge for task-oriented search. In *SIGIR*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2018. From recognition to cognition: Visual commonsense reasoning. *ArXiv*, abs/1811.10830.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Appendix

9.1 Hyperparameter Tuning

QUARTET fine-tunes BERT, allowing us to re-use the same hyperparameters as BERT with small adjustments in the recommended range (Devlin et al., 2018). We use the BERT-base-uncased version with a hidden size of 768. We found the best hyperparameter settings by searching the space using the following hyperparameters.

1. weight decay = { 0.1, 0.01, 0.05 }

2. dropout = {0.1, 0.2, 0.3 }

3. learning rate = {1e-05, 2e-05, 5e-05}