Could you give me a hint? Generating inference graphs for defeasible reasoning

Aman Madaan [*], **Dheeraj Rajagopal** [*], **Niket Tandon** [*†], **Yiming Yang, Eduard Hovy**
Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[†] Allen Institute for Artificial Intelligence, Seattle, WA, USA
{dheeraj,amadaan,yiming,hovy}@cs.cmu.edu
{nikett}@allenai.org

Abstract

Defeasible reasoning is a mode of reasoning where conclusions can be overturned by taking into account new evidence. A commonly used method in cognitive science and logic literature is to handcraft argumentation supporting inference graphs. While humans find inference graphs very useful for reasoning, constructing them at scale is difficult. In this paper, we automatically generate such inference graphs through transfer learning from a related NLP task that shares the kind of reasoning that inference graphs support. Through automated metrics and human evaluation, we find that our method generates meaningful graphs for the defeasible inference task. Human accuracy on this task improves by 20% by consulting the generated graphs. Our findings open up exciting new research avenues for cases where machine reasoning can help human reasoning.[1]

1 Introduction

Defeasible inference (Rudinger et al., 2020) is a mode of reasoning in which given a premise **P** (Rob went for a hike), a hypothesis **H** (Rob saw an elephant, it was pink) may be weakened or overturned in light of new evidence i.e., an update **U** (Rob often has hallucinations). Given the non-monotonic nature of this reasoning, humans find it challenging to master this task (Morgan, 2004). This problem has been widely studied in classical AI through logic (Israel, 1980; McCarthy, 1981), and in cognitive science through argumentative models (Pollock, 1987). A prominent approach is to support defeasible inference through argumentations by constructing an *inference graph* (Pollock, 2009).

Despite their prominence (Bentahar et al., 2010), argumentative models are not scalable because an inference graph needs to be handcrafted for every example. Recently, Rudinger et al. (2020) proposed two auxiliary tasks related to defeasible inference: (i) an NLI task to predict whether an update **U** would weaken or strengthen a hypothesis **H**, and (ii) a generative task to generate an update **U** given a premise **P** and a hypothesis **H**. However, this only addresses a part of the problem because their inference is still not supported by the line of reasoning that a human typically uses to solve this task, namely mediators (e.g., hallucinations can be deceptive) and contextualizers (some elephants can have mutated gene which makes them look different) that are inherently embedded in an inference graph, limiting their utility for humans (figure 1).

In this paper, we adopt the concept of an inference graph for defeasible reasoning from cognitive science and provide a computational model to make their generation scalable. Training such a model would require a large amount of annotated inference graphs, which will be too expensive to obtain. Instead, our solution is to draw a parallel to a related reasoning task in NLP (Tandon et al., 2019), where the reasoning is supported by a graph that we find has similarities with the kind of reasoning that an inference graph supports. We train a model that can learn from the NLP task and effectively transfer it to generate inference graphs. Such transfer learning is made possible due to the powerful seq-to-seq neural language models that did not exist before.

The contributions of this paper are the answers to the following two research questions:

RQ1 Can we automate the construction of the argumentation supporting inference graphs? In §2, we show that we can effectively construct meaningful graphs using transfer learning.

RQ2 Can our generated graphs help improve hu-

[*] Equal Contribution

[1] A dataset of 230,000 influence graphs for each defeasible query is located at: https://tinyurl.com/defeasiblegraphs.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5138–5147
August 1–6, 2021. ©2021 Association for Computational Linguistics

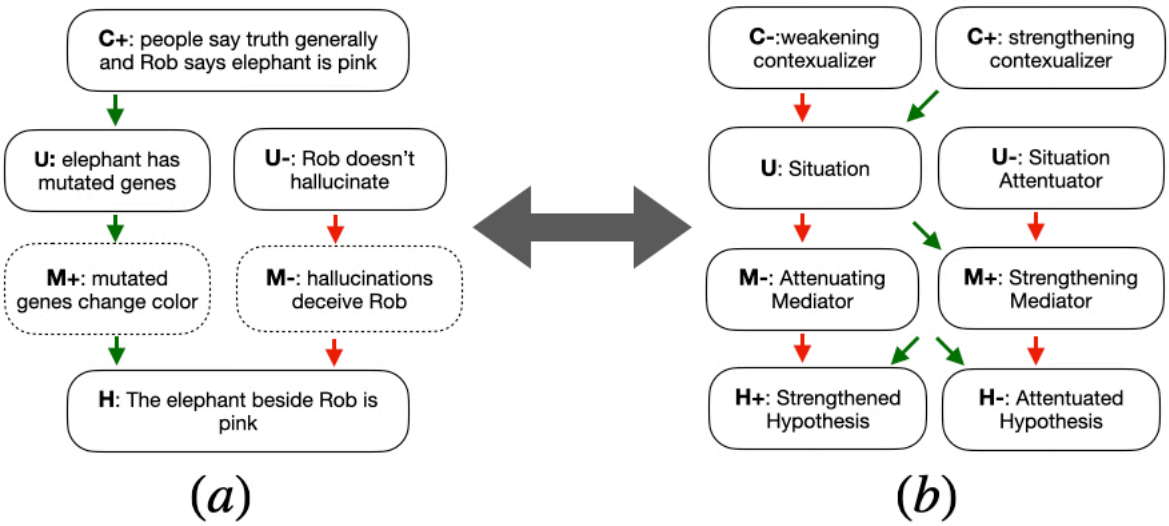

Figure 1: (**a**) An example of an Inference Graph adapted from Pollock (2009) and (**b**) Structure of an Influence Graph adapted from WIQA (Tandon et al., 2019) dataset. The adapted influence graph incorporates the contextualizers, mediators, hypotheses and situations, making them useful for defeasible reasoning.

man performance? In §3, we show that humans leverage generated graphs to improve their performance on a previously reported benchmark.

2 RQ1: Generating argumentation supporting Inference Graphs

We start by drawing parallels to a counterfactual reasoning task in NLP - the WIQA (Tandon et al., 2019) task. WIQA consists of a set of procedural passages, each accompanied by a human-curated *influence graph*. The influence graph captures the causal influences between the events in the context of the process described by the passage. We draw a connection between inference graphs (Pollock, 2009) and influence graphs (Tandon et al., 2019) by drawing parallels between their reasoning structures. In essence, each inference graph from Pollock (1987) can be instantiated via an influence graph from Tandon et al. (2019) by interpreting the nodes in both the graphs as follows (Figure 1):

i. **Contextualizers (C):** these nodes set the context around a situation and connect to the **P** in some way.

ii. **Updates (U):** these nodes are new situations that emerge which might overturn an inference.

iii. **Hypothesis (H):** Hypothesis nodes describes the outcome/conclusion of the situation.

iv. **Mediators (M):** Mediators are nodes that help bridge the knowledge gap between a situation and a hypothesis node by explaining their connection explicitly.

Figure 1 presents an example to highlight the similarities between the two graphs by labeling an example node adapted from (Pollock, 2009), and the structure of the influence graph from (Tandon et al., 2019) with the four node types that we defined above. A green edge indicates that the source node has a positive influence on the target node, and a red edge indicates a negative influence. Further, each node can either act as a *strengthener* (**+**) or a *weakener* (**-**) for the hypothesis. Consequently, these graphs can support similar type of reasoning e.g., the effect of **U** on **H** and how this can change in light of external influences (**C**) is captured by graph paths **C+** to **U** and from **U** via a mediator node (**M+**/**M-**) to **H**. Inspired by these similarities, we hypothesize that influence graphs can be used to supplement defeasible reasoning.

2.1 Influence Graphs Generation

To obtain an influence graph for each defeasible query, we perform a zero-shot transfer from WIQA (Tandon et al., 2019), a corpus of 2100 (passage, influence graphs) pairs.[2].

Training : We treat influence graph generation as a sequence-to-sequence mapping task. We leverage WIQA to derive parallel data $\{(\mathbf{seq}^i_{ip}, \mathbf{seq}^i_{op})\}_{i=1}^N$ for the task. Let $(\mathbf{T}_i, \mathbf{G}_i)$ be a sample in WIQA,

[2]Dataset details in the Appendix §E.

where $\mathbf{T}_i$ is the passage text (e.g. describing how viruses spread), and $\mathbf{G}_i$ is the corresponding influence graph (e.g., Figure 2). To create tokens of the input sequence $\mathbf{seq}^i_{ip}$, the model trains best with explicit markers:[3]

$$\mathbf{seq}^i_{ip} = \text{Premise: } \mathbf{T}_i \mid \text{Update: } \mathbf{U}_i \mid \text{less/ more: } \mathbf{H}_i \quad (1)$$

where $\mathbf{T}_i$ is the passage text (e.g. steps describing how viruses spread) and $\mathbf{U}_i$ and $\mathbf{H}_i$ are nodes of $\mathbf{G}_i$ (these are phrases as shown in Figure 2).

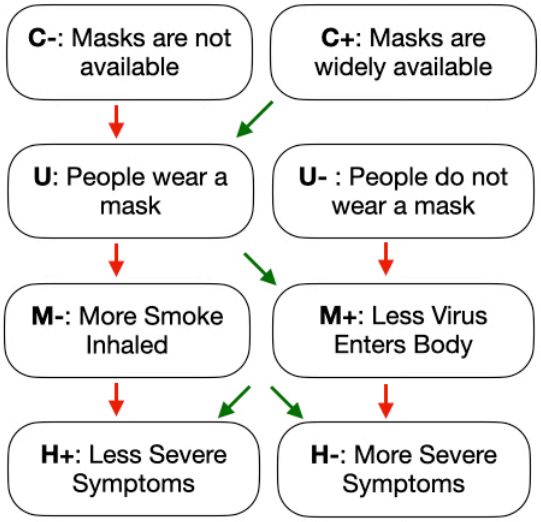

Figure 2: An example of an influence graph similar to ones in WIQA that we train on.

The output $\mathbf{seq}^i_{op}$ is set to a DOT-string representation of the corresponding influence graph $\mathbf{G}_i$, as such a representation was shown to be effective at extracting high-quality graphs (Madaan and Yang, 2021) from free-form text using language models (examples in the appendix). Thus, each passage-graph pair $(\mathbf{T}_i, \mathbf{G}_i)$ from WIQA is mapped to an input-output pair $\mathcal{D} = (\mathbf{seq}^i_{ip}, \mathbf{seq}^i_{op})$. We use this corpus to fine-tune an autoregressive language model $\mathcal{L}$ for graph generation. Essentially, the fine-tuned $\mathcal{L}$ allows us to efficiently sample an influence graph for a given input sequence $\mathbf{seq}^j_{ip}$ by drawing samples from $\mathbf{G}_j \sim P_\theta(y \mid \mathbf{seq}^j_{ip})$ using greedy sampling, where θ denotes the parameters of the language model.

Zero-shot Transfer to Defeasible Inference : We use the model $\mathcal{L}$ trained on WIQA to generate inference graphs on the defeasible inference dataset by Rudinger et al. (2020). We obtain an influence graph for each defeasible input (**P**, **H**, **U**) by converting it to an input sequence that can be fed to $\mathcal{L}$ by filling the template (1). This conversion from (**P**, **H**, **U**) to template (1) is done by setting the premise **P** as the context passage **T**, the update **U** as the node **U**, and the attenuated and strengthened outcomes are simulated by prefixing the hypothesis **H** with the tokens *Less* and *More* respectively. This input is then passed to the $\mathcal{L}$ to generate an influence graph.

Results on Influence Graph Generation We use T5-11B (Raffel et al., 2020) fine-tuned on $\mathcal{D}$ derived from WIQA (§2.1) as our graph generation language model ($\mathcal{L}$). All the graphs generated by our model were in valid DOT format. We use the standard generation metrics BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) to evaluate $\mathcal{L}$ on the test split of WIQA. Each node N_i in the reference graph is compared with the corresponding generated node $\hat{N}_i$ using $\texttt{BLEU}(N_i, \hat{N}_i)$ (Node-BLEU). Further, node-edge-node pairs (neighbors) (N_i, N_j) and $(\hat{N}_i, \hat{N}_j)$ are compared using Rel-BLEU = $\texttt{HM}(\texttt{BLEU}(N_i, \hat{N}_i), \texttt{BLEU}(N_j, \hat{N}_j))$ where HM is the harmonic mean. These metrics are averaged over the graph (i.e., across the nodes and the edges), and further averaged across the corpus. We perform these experiments across two different language models: GPT-2-MEDIUM (Radford et al., 2019) and T5-11B. Finally, we calculate the overlap in the edge structures of the reference and generated graphs match as Edge-MATCH%. We report the numbers in Table 1, and include a random baseline for reference. A random baseline will correctly generate the nodes **S**, **H+**, and **H-** as they are part of the query ($\frac{3}{8}$ nodes). As neither of these nodes are connected to another, the random baseline will likely not generate any node pair correctly (Rel-BLEU ~ 0). Since two unique graph structures are possible (Tandon et al., 2019), a random baseline would get Edge-match $\sim 50\%$. Table 1 shows that our T5-based model is able to generate syntactically valid (high edge-match) and semantically meaningful graphs. Additionally, we find that our generated graphs are helpful to humans on a downstream task, as described next.

3 RQ2: Do generated graphs help humans at defeasible reasoning?

Human Evaluation Rudinger et al. (2020) performed a human evaluation on 2000 defeasible queries, where given (**P**, **H**, **U**), the task was to label the nature of the effect of **U** on **H** as *Intensifies*

[3] An example shown in Appendix §A.

Model	Random	GPT-2-MEDIUM	T5-11B
Node-BLEU	37.5	46.05	**50.94**
Rel-BLEU	0.0	19.34	**33.01**
Edge-match%	50.0	92.86	**97.63**

Table 1: Results on automated metrics showing that our T5-11B model is able to generate very accurate graph structure and meaningful nodes that sufficiently match the reference nodes.

or *Attenuates*. Three human judges labeled each query, and the majority label was then compared with the ground-truth to ascertain the accuracy. In their setup, human judges were collectively right on 1745 samples (correct pool) and wrong on 255 samples (wrong pool). We create a challenging pool of 510 queries for the human judges by combining the 255 queries in the wrong pool with 255 queries sampled from the correct pool, giving a baseline accuracy of 50% for this eval pool. Each query in this pool is supplemented with a generated influence graph (§2).[4] We found that our generated influence graphs showed high-levels of redundancy in contextualizers and mediators, with about 46% of the generated influence graphs repeating these nodes. We found that humans find it simpler to follow positive chains of influence, so to reduce their cognitive load, we post-process each influence graph to only retain the strengthening contextualizer (Figure 1), the situation (**U**), the strengthening mediator (**M+**), and the hypothesis (**H**).

In order to establish comparable gains, we replicate the evaluation setup of Rudinger et al. (2020) by using use the same Amazon Mechanical Turk template and the instruction set, and the same pool of 230 qualified annotators that Rudinger et al. (2020) selected based on a paid qualification test, in which the workers were asked to answer SNLI queries of varying levels of difficulty. We paid slightly above $15 per hour for the tasks.

For each query, in addition to answering the defeasible question, three judges were asked to evaluate the augmented influence graphs on two aspects:

i) **Is the influence graph useful?** The judges were allowed to select from the following:

   (a) *helpful*: the graph was crucial in helping towards answering the question

   (b) *relevant but not helpful*: the graph had the right topic (relevant to the question) but did not help in answering the question.

   (c) *irrelevant or misleading*: the graph was irrelevant to the question or misled the human judge to a wrong answer.

ii) **Why is the influence graph useful?** The judges were given an option to highlight the most useful aspect of the generated influence graph. They were allowed to tag one or more of the following aspects as the most helpful: i) Extraneous node, ii) Mediating node, and iii) Structure of the graph.

We summarize the key findings below.

Finding 1: influence graphs are helpful and relevant As Table 2 shows, a large majority of the human judges found the influence graphs to be helpful or relevant. We calculate the inter-annotator agreement for this question using majority-agreement $= \frac{1}{N}\sum_{i=1}^{N} \texttt{ma}_i$ where $\texttt{ma}_i$ indicates a majority agreement for the ith sample (i.e., at least 2 out of 3 judges agreed on the label for the sample). The majority-agreement (ma) on these labels was 0.83. The judges marked about 25% of the graphs as relevant but not helpful. The graphs in such cases were on topic but not helpful in answering the query, thereby distinguishing the cases when the graph was crucial in reaching the correct answer. Finally, we note that the graphs provided as hints could have been helpful in two ways: by helping the human annotators arrive at the answer, or by reinforcing their mental picture that helped them in making the right decision. Future research in this direction is needed to study these aspects in depth.

Helpful	47.25
Relevant but not helpful	25.09
Irrelevant or misleading	10.58
No majority agreement	17.05

Table 2: Helpfulness of the augmentations.

Finding 2: Mediators are the most helpful for defeasible queries For every sample, we asked the human judges to mark which parts of the graph was the most helpful (as shown in Figure 6 in Appendix §D.1). The judges could select more than one aspect of the graph if they found multiple useful aspects. Table 3 shows the percentage of human

[4] Discussion on IRB exemption in Section §B.

judges that selected the particular graph aspect as most helpful. We observe that 49.48% of the judges who found the graphs useful indicated the mediator node as the most helpful. This indicates that while there may be other events that impact **U** and **H**, the mediating events are the most informative in determining the type of link between them.

Aspect	% marked useful
Mediator	49.48
Extraneous	32.03
Structure	12.81
None helpful	5.68

Table 3: Most useful aspects of an influence graph.

Finding 3: Machine generated influence graphs help humans in defeasible reasoning Table 4 shows that performance improves across all three tasks when the defeasible query is augmented with an influence graph. On our challenging set of 510 queries, the overall accuracy jumps nearly 20 points from 0.50 to 0.698. Figure 3 highlights that 113 queries that were previously given the wrong answers were marked correctly when augmented with the influence graphs.

Dataset	Human (Rudinger et al., 2020)	Human (ours)
SNLI	0.461 ± 0.11	0.553 ± 0.11
SOCIAL	0.628 ± 0.07	0.814 ± 0.06
ATOMIC	0.418 ± 0.06	0.657 ± 0.06
overall	0.500 ± 0.04	$\mathbf{0.698 \pm 0.04}$

Table 4: Human performance (accuracy) on the three tasks with and without generated influence graphs along with Wilson's score intervals for $\alpha = 95\%$. We tested the statistical significance of these results using the McNemar's test (McNemar, 1947) and found the results to be statistically highly significant ($p < 1e-6$).

4 Discussion and Conclusion

Our work takes the idea of using inference graphs for defeasible inference and scales up its usability by automatically generating and augmenting them to a downstream defeasible task that both humans and machines are known to find difficult. We identify that the contextualizer and mediator nodes are crucial to defeasible inference, and show that our generated graphs generate these critical nodes

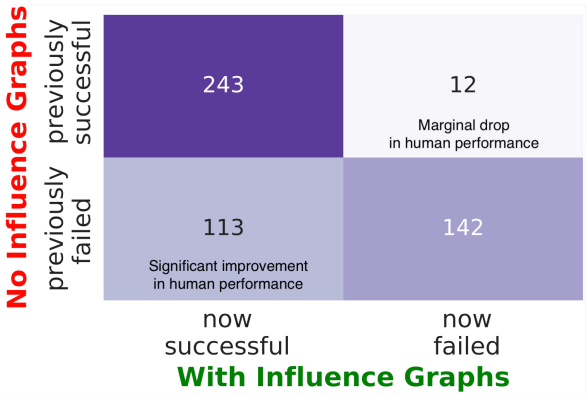

Figure 3: Human performance before and after the human judges were provided with the influence graph.

effectively. Humans perform significantly better (20% absolute improvement) across diverse defeasible datasets and overwhelmingly attribute their success to the mediator nodes – giving insights into what helps and why. In this case study, we show that machines can fill the gaps in human knowledge when for defeasible reasoning. While we establish that humans are helped by these graphs, a further investigation on how (and if) the graphs reinforced their beliefs, and what additional information in the graphs was beneficial to their understanding is essential. Furthermore, a deeper understanding of the trade-offs (time spent in answering these questions with and without the graphs) also forms important future work.

Acknowledgments

We would like to thank Peter Clark for the thoughtful discussions, and the anonymous reviewers for valuable feedback.

This material is partly based on research sponsored in part by the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government. We would like to thank Google for providing the TPU machines for conducting experiments.

References

J. Bentahar, B. Moulin, and M. Bélanger. 2010. A taxonomy of argumentation models used for knowledge representation. *Artificial Intelligence Review*, 33:211–259.

D. Israel. 1980. What's wrong with non-monotonic logic? In *AAAI*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Aman Madaan and Yiming Yang. 2021. Neural language modeling for contextualized temporal graph generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 864–881, Online. Association for Computational Linguistics.

J. McCarthy. 1981. Some philosophical problems from the standpoint of artificial intelligence. *Machine intelligence*.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

C. Morgan. 2004. The nature of nonmonotonic reasoning. *Minds and Machines*, 10:321–360.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

J. Pollock. 1987. Defeasible reasoning. *Cogn. Sci.*, 11:481–518.

J. Pollock. 2009. A recursive semantics for defeasible reasoning. In *Argumentation in Artificial Intelligence*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Rachel Rudinger, Vered Shwartz, Jena D. Hwang, Chandra Bhagavatula, Maxwell Forbes, Ronan Le Bras, Noah A. Smith, and Yejin Choi. 2020. Thinking like a skeptic: Defeasible inference in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4661–4675, Online. Association for Computational Linguistics.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. Wiqa: A dataset for "what if..." reasoning over procedural text. 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 6078–6087.

A Sample input-output sequence for training $\mathcal{L}$

We now present a sample input-output sequence used to train out $\mathcal{L}$ for graph generation. The input-output sample ($\mathbf{seq_{ip}}, \mathbf{seq_{op}}$) is presented below. As mentioned in Section

1. As described in section 2.1, each input sequence $\mathbf{seq_{ip}}$ is formatted in a special template to be fed to the language model (Template (1)). We show an example of the same next for a sample from our training data. **Premise:** `Sunlight shines on plants. Cells with chlorophyll in them ... other parts of the plant.` | **Situation** : `more minerals are absorbed` | **Less** : `LESS sugar and oxygen being produced` | **More** : `MORE sugar and oxygen being produced`

2. Each output graph is encoded in as a DOT string. The output DOT sequence $\mathbf{seq_{op}}$ corresponding to the input shown above is: `strict digraph "C+ : less minerals in the soil [OR] less root system" -> "S : more minerals are absorbed" [label=hurts]; "C- :more minerals in the soil [OR] a better root system" -> "S : more minerals are absorbed" [label=helps]; "S : more minerals are absorbed" -> "M- : less conversion into sugars [OR] less oxygen produced" [label=hurts]; "S : more minerals are absorbed" -> "M+ : more conversion into sugars" [label=helps]; "S- : less minerals absorbed [OR] less root system" -> "M+ : more conversion into sugars" [label=hurts]; "M- : less conversion into sugars [OR] less oxygen produced" -> "H- : LESS sugar and oxygen being produced" [label=helps]; "M- : less conversion into sugars [OR] less oxygen produced" -> "H+ : MORE sugar and oxygen being produced" [label=hurts]; "M+ : more conversion into sugars" -> "H+ : MORE sugar and oxygen being produced" [label=helps]; "M+ : more conversion into sugars" -> "H- : LESS sugar and oxygen being produced" [label=hurts];`

B IRB Exemption

Our study was not an experimentation on humans (posed no identifiable risk to the human judges), did not collect any identifying information, and ensured it involved only adults. As per the IRB guidelines, this falls under the purview of human research, and we are not publishing individual workers' answers but rather the data is tallied up, much like a "benign behavioral intervention." This exempts us from IRB (category 3 of Federal Regulations for Protection of Human Research Subjects https://www.hhs.gov/ohrp/regulations-and-policy/regulations/45-cfr-46/).

C Infrastructure and hyperparameters

To train the T5-11B model, comprising of 11 billion parameters, we used v3-8 TPUs. The average time to train was 7 hours for about 10 epochs. We used the same hyperparameters as provided with the T5 checkpoint at `gs://t5-data/pretrained_models/11B`. We used maximum block size of 512 tokens, and max generation length set to 512. For decoding, we sample according to predicted distribution. We train the GPT-2 model on a Nvidia GTX 2080 Ti, and training the model takes about 30 minutes per epoch.

We use the medium (355M) variant of GPT-2 (Radford et al., 2019) with 24 layers, 1024 hidden size, 16 attention heads.

D Details of our Mechanical Turk Setup

We follow the same instructions for humans as (Rudinger et al., 2020)[5], and only additionally provided instructions for the inference graph. We used a pool of 230 annotators that were previously qualified and selected to do the defeasible inference task, thus providing a fair comparison to their setup. Eventually 12 workers out of these 230 workers

[5]We are grateful to the authors of (Rudinger et al., 2020) for sharing their mechanical turk setup template with us.

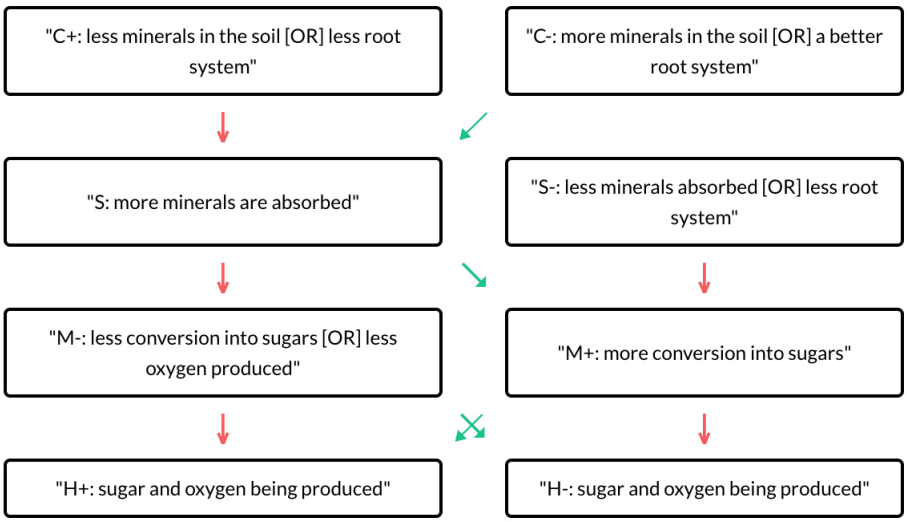

Figure 4: The influence graph corresponding to dot code shown in $\mathbf{seq_{op}}$

worked on our HITs. The graph we showed to humans was a subgraph of the inference graph, where the selected path has the relevant content from the inference graph to avoid showing redundant opposite edges. These redundant edges are useful in training a model as the model must jointly predict all the nodes, but this is redundant for humans. Figure 5 shows this subgraph.

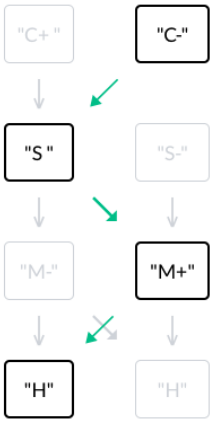

Figure 5: Part of the generated influence graph that is presented in the hit.

D.1 A sample HIT

We now show a sample HIT in Figure 6. We had two set of annotations in every HIT.

D.2 Examples that helped humans

Next, we show two examples (Figure 7, Figure 8) where humans were previously unsuccessful on this answer (in the original setup of (Rudinger et al., 2020)), and were successful now having looked at the inference graphs. The humans marked that the mediator nodes and the contextualizer nodes provide useful information.

E Dataset

Dataset	Split	# Samples	Total
WIQA	train	1522	2107
	test	189	
	dev	152	
ATOMIC	train	35,001	42,977
	test	4137	
	dev	3839	
SOCIAL	train	88,675	92,295
	test	1836	
	dev	1784	
SNLI	train	77,015	95,795
	test	9438	
	dev	9342	

Table 5: Number of samples in each dataset by split. ATOMIC, SNLI, SOCIAL are available at https://github.com/rudinger/defeasible-nli, WIQA is avilable at https://allenai.org/data/wiqa

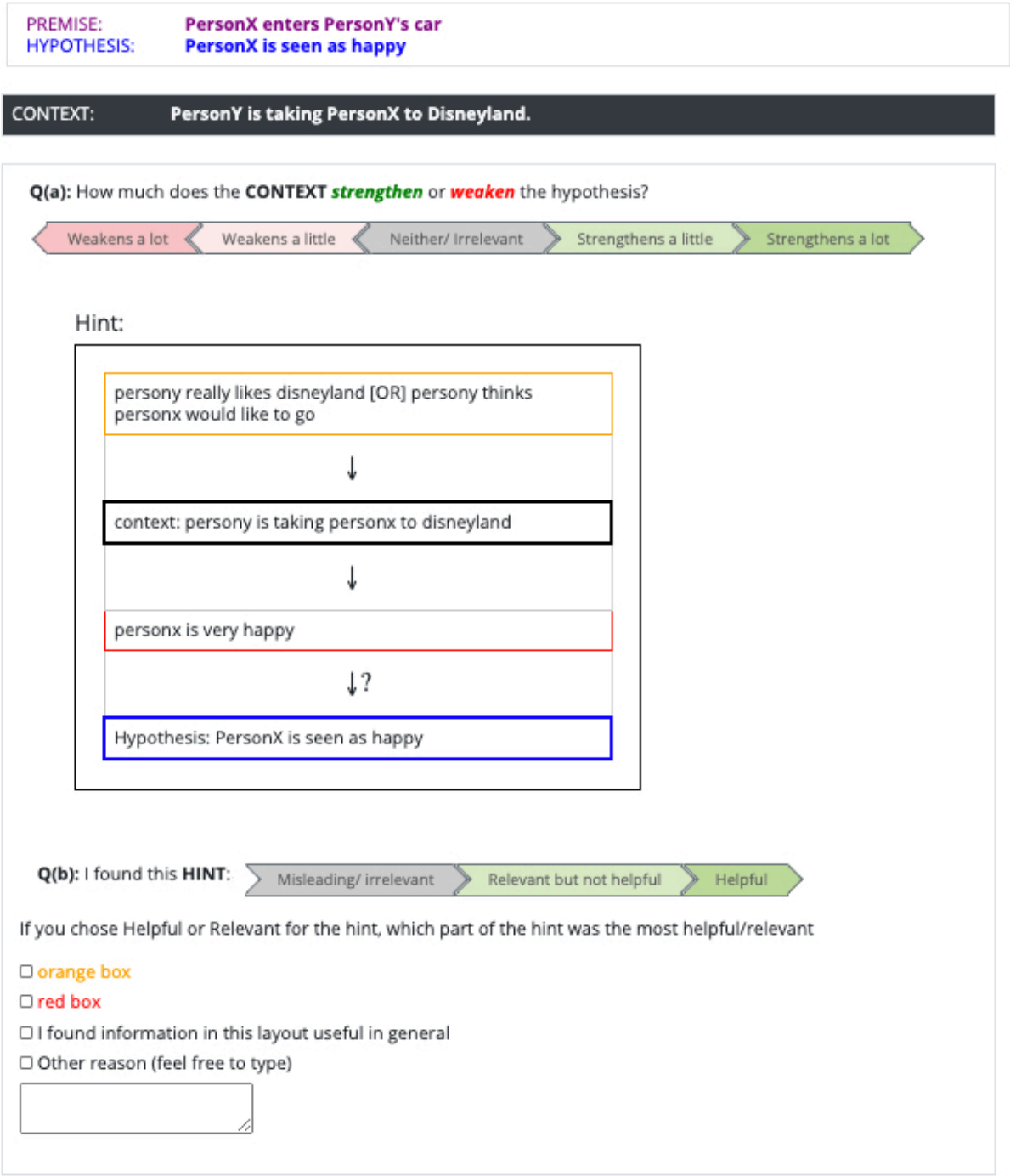

Figure 6: A sample HIT in mechanical turk.

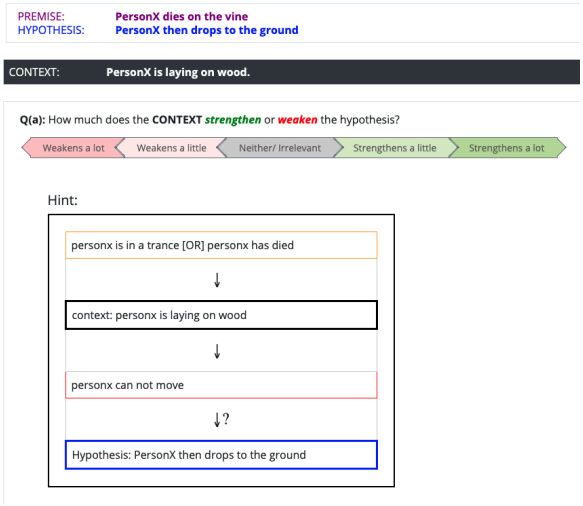

Figure 7: An example where the graph helped the human in getting the correct answer, that humans were unsuccessful on, in the past.

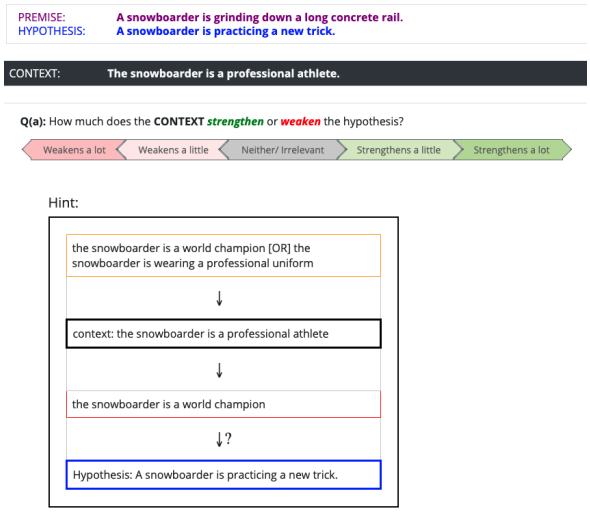

Figure 8: Another example where the graph helped the human in getting the correct answer, that humans were unsuccessful on, in the past.