ExplaGraphs: An Explanation Graph Generation Task for Structured Commonsense Reasoning

Swarnadeep Saha **Prateek Yadav** **Lisa Bauer** **Mohit Bansal**
UNC Chapel Hill
{swarna, prateek, lbauer6, mbansal}@cs.unc.edu

Abstract

Recent commonsense-reasoning tasks are typically *discriminative* in nature, where a model answers a multiple-choice question for a certain context. Discriminative tasks are limiting because they fail to adequately evaluate the model's ability to reason and explain predictions with underlying commonsense knowledge. They also allow such models to use reasoning shortcuts and not be "right for the right reasons". In this work, we present ExplaGraphs, a new *generative* and *structured* commonsense-reasoning task (and an associated dataset) of explanation graph generation for stance prediction. Specifically, given a belief and an argument, a model has to predict if the argument supports or counters the belief and also generate a commonsense-augmented graph that serves as non-trivial, complete, and unambiguous explanation for the predicted stance. We collect explanation graphs through a novel *Create-Verify-And-Refine* graph collection framework that improves the graph quality (up to 90%) via multiple rounds of verification and refinement. A significant 79% of our graphs contain external commonsense nodes with diverse structures and reasoning depths. Next, we propose a multi-level evaluation framework, consisting of automatic metrics and human evaluation, that check for the structural and semantic correctness of the generated graphs and their degree of match with ground-truth graphs. Finally, we present several structured, commonsense-augmented, and text generation models as strong starting points for this explanation graph generation task, and observe that there is a large gap with human performance, thereby encouraging future work for this new challenging task.[1]

1 Introduction

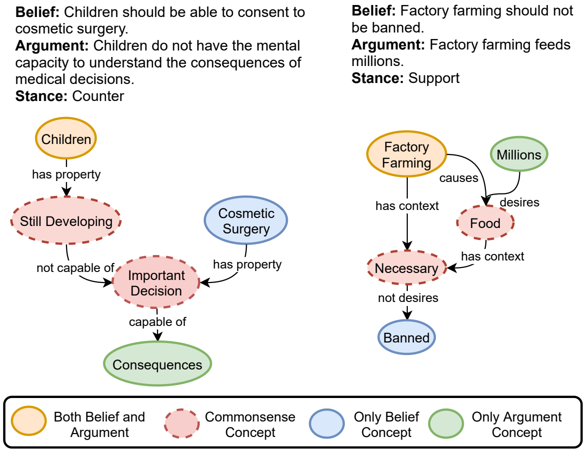


Figure 1: Two representative examples from our dataset. Explanation graphs are read and reasoned through by following the edges that explain why the argument supports or counters the belief.

Current state-of-the-art commonsense reasoning (CSR) (Davis and Marcus, 2015) models are typically trained and evaluated on *discriminative* tasks, in which a model answers a multiple-choice question for a certain context (Zellers et al., 2018; Sap et al., 2019b; Bisk et al., 2020). While pretrained language models perform well on these tasks (Lourie et al., 2021), this setup limits the exploration and evaluation of a model's ability to reason and explain its predictions with relevant commonsense knowledge, thereby allowing models to solve tasks by using shortcuts, statistical biases or annotation artifacts (Gururangan et al., 2018; McCoy et al., 2019). Thus, we emphasize the importance of *generative* CSR capability, in which a model has to compose and reveal the plausible commonsense knowledge required to solve a reasoning task. Moreover, structured (e.g., graph-based) commonsense explanations, unlike unstructured natural language explanations, can more explicitly explain and evaluate the reasoning structures of the model by visually laying out the relevant context and commonsense knowledge edges, chains, and subgraphs.

We propose ExplaGraphs, a new *generative* and *structured* CSR task (in English) of explana-

[1] ExplaGraphs dataset will be publicly available at https://explagraphs.github.io/.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7716–7740
November 7–11, 2021. ©2021 Association for Computational Linguistics

tion graph generation for stance prediction on debate topics. Specifically, our task requires a model to predict whether a certain argument supports or counters a belief, but correspondingly, also generate a commonsense explanation graph that explicitly lays out the reasoning process involved in inferring the predicted stance. Consider Fig. 1 showing two examples with belief, argument, and stance (support or counter) from our benchmarking dataset collected for this task. Each example requires understanding social, cultural, or taxonomic commonsense knowledge about debate topics in order to infer the correct stance. The example on the left requires the knowledge that "children" are "still developing" and hence not capable of making an "important decision" like "cosmetic surgery" which has "consequences". Given this knowledge, one can understand that the argument is counter to the belief. We represent this knowledge in the form of a commonsense explanation graph.

Graphs are efficient for representing explanations due to multiple reasons: (1) unlike a chain of facts (Khot et al., 2020; Jhamtani and Clark, 2020; Inoue et al., 2020; Geva et al., 2021), they can capture complex dependencies between facts, while also avoiding redundancy (e.g., "Factory farming causes food and millions desire food" forms a "V-structure"), (2) unlike natural language explanations (Camburu et al., 2018; Rajani et al., 2019; Narang et al., 2020; Brahman et al., 2021; Zhang et al., 2020), it is easier to impose task-specific constraints on graphs (e.g., connectivity, acyclicity), that eventually help in better quality control during data collection (Sec. 4) and designing structural validity metrics for model-evaluation (Sec. 6), and (3) unlike semi-structured templates (Ye et al., 2020; Mostafazadeh et al., 2020) or extractive rationales (Zaidan et al., 2007; Lei et al., 2016; Yu et al., 2019; DeYoung et al., 2020), they allow for more flexibility and expressiveness. Graphs can encode any reasoning structure and the nodes are not limited to just phrases from the context. As shown in Fig. 1, our explanations are connected directed acyclic graphs (DAGs), in which the nodes are either internal concepts (short phrases from the belief or argument), or external commonsense concepts (dashed-red), essential for connecting the internal concepts in a way that the stance is inferred. The edges are labeled with commonsense relations chosen from a pre-defined set. While some edges might not necessarily be factual (e.g., "Factory farming; has context; necessary"), note that such edges are essential in the context for composing an explanation that is indicative of the stance. Semantically, our graphs are extended structured arguments, augmented with commonsense knowledge.

We construct a benchmarking dataset for our task through a novel *Create-Verify-And-Refine* graph collection framework. These graphs serve as non-trivial (not paraphrasing the belief as an edge), complete (explicitly connects the argument to the belief) and unambiguous (infers the target stance) explanations for the task (Sec. 3). The graph quality is iteratively improved (up to 90%) through multiple verification and refinement rounds. 79% of our graphs contain external commonsense nodes, indicating that commonsense is a critical component of our task. Explanation graph generation poses several syntactic and semantic challenges like predicting the internal nodes, generating the external concepts and predicting and labeling the edges in a way that leads to a connected DAG. Finally, the graph should unambiguously infer the target stance.

We next present a multi-level evaluation framework for our task (Sec. 6, Fig. 4), consisting of diverse automatic metrics and human evaluation. The evaluation framework checks for stance and graph consistency along with the structural and semantic correctness of explanation graphs, both locally by evaluating the importance of each edge and globally by the graph's ability to reveal the target stance. Furthermore, we propose graph-matching metrics like Graph Edit Distance (Abu-Aisheh et al., 2015) and ones that extend text-generation metrics for graphs (based on multiple test graphs in our dataset). Lastly, as some strong initial baseline models for this new task, we propose a commonsense-augmented structured prediction model that predicts nodes and edges jointly and enforces global graph constraints (e.g., connectivity) through an Integer Linear Program (ILP). We also experiment with BART (Lewis et al., 2019) and T5 (Raffel et al., 2020) based models, and show that all these models have difficulty in generating meaningful graph explanations for our challenging task, leaving a large gap between model and human performance. Overall, our main contributions are:

- We propose ExplaGraphs, a *generative* and *structured* commonsense-reasoning task of explanation graph generation for stance prediction.
- We construct a benchmarking dataset for our task and propose a novel *Create-Verify-And-Refine*

graph collection framework for collecting graphs that serve as explanations for the task. Our framework is generalizable to any crowdsourced collection of graph-structured data.

- We propose a multi-level evaluation framework with automatic metrics and human evaluation, that compute structural and semantic correctness of graphs and match with human-written graphs.
- We propose a commonsense-augmented structured model and BART/T5 based models for this task, and find that they are relatively weak at generating reasoning graphs, obtaining 20% accuracy (compared to human performance of 84%).

We encourage researchers to use our benchmark as a way to improve and explore structured commonsense reasoning capabilities of models.

2 Related Work

Structured Explanations in NLP: Explanation datasets in NLP (Wiegreffe and Marasović, 2021) take three major forms: (1) extractive rationales (Zaidan et al., 2007; Lei et al., 2016; Yu et al., 2019; DeYoung et al., 2020), (2) free-form or natural language explanations (Camburu et al., 2018; Rajani et al., 2019; Narang et al., 2020; Brahman et al., 2021; Zhang et al., 2020), and (3) structured explanations consisting of explanations graphs (Jansen et al., 2018; Jansen and Ustalov, 2019; Xie et al., 2020; Saha et al., 2020; Kalyanpur et al., 2020; Saha et al., 2021), chain of facts (Khot et al., 2020; Jhamtani and Clark, 2020; Inoue et al., 2020; Geva et al., 2021) or semi-structured text (Ye et al., 2020). Our commonsense explanations bear most similarity to WorldTree's (Jansen et al., 2018) explanation graphs. However, while they connect lexically-overlapping words, we connect concepts to create facts and diverse reasoning structures in fully-structured graphs with carefully designed constraints for explainability. EXPLAGRAPHS's explanations also share similarities with visual scene graphs from the vision community (Johnson et al., 2015; Xu et al., 2017), in which the image entities are connected via edges to represent relationships.

Commonsense Reasoning Benchmarks: A large variety of CSR tasks have been developed recently, including commonsense extraction (Li et al., 2016; Xu et al., 2018), next situation prediction (Zellers et al., 2018, 2019), cultural, social, and physical commonsense understanding (Lin et al., 2018; Sap et al., 2019a,b; Bisk et al., 2020; Hwang et al., 2020; Forbes et al., 2020), pronoun disambiguation (Sakaguchi et al., 2020; Zhang et al., 2020), abductive commonsense reasoning (Bhagavatula et al., 2019) and general commonsense (Talmor et al., 2019; Huang et al., 2019; Wang et al., 2019; Boratko et al., 2020). While there is an abundance of discriminative commonsense tasks, there are few recent works in generative commonsense tasks. E.g., CommonGen (Lin et al., 2020) generates unstructured commonsense text, and EIGEN (Madaan et al., 2020) considers event influence graph generation. Instead, our work focuses on generating commonsense-augmented explanation graphs.

Stance Prediction and Argumentation: Previous stance prediction works have been largely applied to online content, for political, ideological debates, rumor and fake news detection (Mohammad et al., 2016; Derczynski et al., 2017; Hardalov et al., 2021). Other recent works on argumentation deal with convincingness of claims and arguments Habernal and Gurevych (2016); Gleize et al. (2019) and reasons (Hasan and Ng, 2014). However, to the best of our knowledge, our work is the first to explore explicit commonsense-augmented graph-based explanations for stance prediction.

3 EXPLAGRAPHS Task Definition

We propose EXPLAGRAPHS, a new *generative* and *structured* commonsense-reasoning task, where given a belief about a topic and an argument, a model has to (1) infer the stance (support/counter), and (2) generate the corresponding commonsense explanation graph that explains the inferred stance (Fig. 1). Our primary focus in this work is on the second sub-task that requires generative commonsense reasoning. The explanation graph is a connected and directed acyclic graph, where each node is a concept (short English phrase). Concepts are either internal (part of the belief or the argument) or external (part of neither but essential for filling in any knowledge gap between the belief and the argument). Each directed edge connects two concepts and is labeled with one of the pre-defined commonsense relations. These relations are chosen based on ConceptNet (Liu and Singh, 2004) with three modifications – (1) removing some generic relations like "related to", (2) merging some relations that have similar meanings (e.g. "synonym of" and "similar to"), (3) adding a negated counterpart ("not desires") for every non-negated relation ("desires"), to enable easy construction of support and counter explanations and a balanced set between

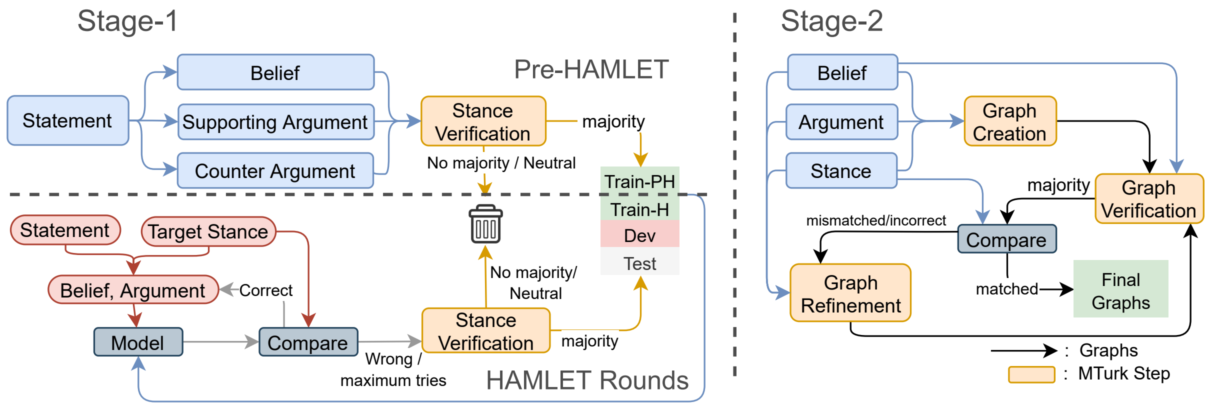


Figure 2: Interface for our data collection framework consisting of two stages. In Stage 1, we collect (belief, argument, stance) triples in pre-HAMLET and multiple HAMLET (human-and-model-in-the-loop) rounds. In each HAMLET round, we collect harder examples by asking the annotators to fool a stance prediction model. In Stage 2, we collect the corresponding explanation graphs through a Create-Verify-And-Refine framework.

negated and non-negated relations (see appendix for full list). Semantically, our explanation graphs are commonsense-augmented structured arguments that explicitly support or counter the belief. All subjective claims in the graph are assumed to be true for inferring the stance. An explanation graph is correct if it is both structurally and semantically correct.

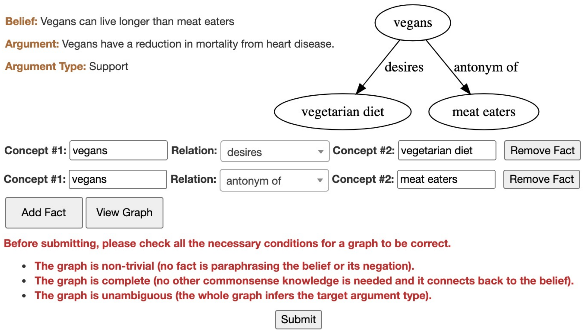
Structural Correctness of Graphs: In order to ensure the structural validity of an explanation graph, we define certain constraints on the graph which not only ensure better quality control during our data collection (Sec. 4) but also simplify the evaluation (Sec. 6), given the open-ended nature of our task. Note that most of these constraints are only possible to impose because of the explicit graphical structure of these explanations.

- Each concept should contain a maximum of three words and each relation should be chosen from the pre-defined set of relations.
- The total number of edges should be between 3 and 8, to ensure a good balance between under-specified and over-specified explanations.
- The graph should contain at least two concepts from the belief and at least two from the argument. This ensures that the graph uses important parts of the belief and argument (exactly, without paraphrasing) to construct the explanation.
- The graph should be a connected DAG to ensure the presence of explicit reasoning chains between the belief and argument and also avoid redundancy or circular explanations. E.g., having “(vegans; antonym of; meat eaters)” makes “(meat eaters; antonym of; vegans)” redundant.

Semantic Correctness of Graphs: We define the semantic correctness of explanation graphs as follows. First, all facts in the graph, individually, should be *semantically coherent*. Second, the graph should be *non-trivial*, *complete* and *unambiguous*. We call a graph *non-trivial* if it uses the argument to arrive at the belief and does not use fact(s) which are mere paraphrases of the belief. E.g., for a belief “Factory farming should be banned”, if the explanation graph contains facts like “(Factory farming; desires; banned)”, then it is only paraphrasing the belief to explain why the belief holds, hence making the graph incorrect. Instead, it should be augmenting the argument with commonsense knowledge like our graph in Fig. 1. A *complete* graph is one which explicitly connects the argument to the belief and no other commonsense knowledge is needed to understand why it supports or counters the belief. E.g., in Fig. 1, the fact “(necessary; not desires; banned)” makes the explanation complete by explicitly connecting back to the belief. We call a graph *unambiguous* if it, as a whole, infers the target stance and only that stance. We revisit these definitions of structural and semantic correctness when evaluating the quality of human-written graphs (Sec. 4.2) as well as model-generated graphs (Sec. 6).

4 Dataset Collection

We collect EXPLAGRAPHS data in two stages via crowdsourcing on Amazon Mechanical Turk (Fig. 2). In Stage 1 (left of Fig. 2), we collect instances of belief, argument and their corresponding stance. In Stage 2 (right of Fig. 2), we collect the corresponding commonsense explanation graph for each



Belief: Vegans can live longer than meat eaters

Argument: Vegans have a reduction in mortality from heart disease.

Argument Type: Support

Concept #1: vegans **Relation:** desires **Concept #2:** vegetarian diet

Concept #1: vegans **Relation:** antonym of **Concept #2:** meat eaters

Before submitting, please check all the necessary conditions for a graph to be correct.

- **The graph is non-trivial (no fact is paraphrasing the belief or its negation).**
- **The graph is complete (no other commonsense knowledge is needed and it connects back to the belief).**
- **The graph is unambiguous (the whole graph infers the target argument type).**

Figure 3: Explanation Graph Creation Interface.

(belief, argument, stance) sample.

4.1 Stage 1: (Belief, Argument, Stance)

In Stage 1, annotators are given prompts that express beliefs about various debate topics, extracted from evidences in Gretz et al. (2019). We use 71 topics in total (see appendix for the list), randomly assigning 53/9/9 disjoint topics to our train/dev/test splits. Given the prompt, annotators write the belief expressed in the prompt and subsequently, a supporting and a counter argument for the belief. Since we focus on commonsense-augmented explanations, we want to ensure that most of our belief, argument pairs require some implicit background commonsense knowledge for understanding why a certain argument supports or refutes the belief. For collecting such pairs, we use Human-And-Model-in-the-Loop Enabled Training (HAMLET) (Nie et al., 2019), a multi-round adversarial data collection procedure that enables the collection of trickier examples with more background commonsense knowledge. Due to space constraints, we discuss this in detail in appendix Sec. 1.1. After stance label verification, we obtain a high fleiss-kappa inter-annotator agreement of 0.61.

4.2 Stage 2: Commonsense Explanation Graph Collection

Given the (belief, argument, stance) triples from Stage 1, we next collect the corresponding commonsense explanation graphs through a generic *Create-Verify-And-Refine* iterative framework.

Graph Creation: Annotators are given a belief, an argument, and the stance (support or counter) and are asked to construct a commonsense-augmented explanation graph that explicitly explains the stance (Fig. 3). A graph is constructed by writing multiple facts, each consisting of two concepts and a chosen relation that connects the two concepts. The annotators write 3-8 facts such that the facts lead to a connected DAG with at least two concepts from the belief and two from the argument. The graphical representation of the explanation provides an explicit structure, thereby allowing us to automatically perform in-browser checks for these structural constraints. Clicking on the "View Graph" button shows the graph written so far. Before submitting, we remind the annotators that they reason through the graph and verify that it is non-trivial, complete and unambiguous (marked red in Fig. 3). See appendix for the graph creation instructions.

Graph Verification: Here, we verify the semantic correctness of graphs (as defined in Sec. 3) because by construction, they are all structurally correct. The explanation graphs should be complete and hence are treated as extended structured arguments with commonsense. Thus, in our graph verification step, we provide annotators with only the belief and the corresponding explanation graph and ask them to reason through it to infer the stance. Additionally, we include a third category of "incorrect" graphs which is broadly aimed at identifying the ill-formed graphs with either semantically incoherent facts, trivial belief-paraphrased facts, or no explicit connection back to the belief (incomplete or ambiguous). Each graph is annotated by three verifiers into one of support/counter/incorrect. A graph is considered correct if and only if the majority label matches the original stance (already known from Stage 1). All other graphs are sent for refinement (described next, also see Fig. 2) because they are either incorrect or infer the wrong stance. See appendix for the graph verification interface.

Graph Refinement: During graph refinement, in addition to the belief, argument, and the target stance, annotators are provided with the initial incorrect graph along with the verification label from the previous stage. Then another qualified annotator who is not the author of the initial graph is asked to refine it. Refinement is defined in terms of three edit operations on the graph: (1) adding a new fact, (2) removing an existing fact, and (3) replacing an existing fact. We again ensure that the refined graph adheres to the structural constraints. See appendix for the instructions and interface.

Graph Quality: The refined graphs are again sent to the verification stage and the process iterates between the verification and refinement stages until we obtain a high percentage of correct graphs. We perform two rounds of refinement, and obtain a high *90% of semantically correct graphs*

	Train			Dev			Test (2 graphs/sample)		
Round	S / C	Total	Topics	S / C	Total	Topics	S / C	Total	Topics
Pre-HAMLET	541 / 457	998	33	-	-	-	-	-	-
HAMLET R1	347 / 226	573	20	79 / 76	155	9	84 / 80	164	9
HAMLET R2	234 / 181	415	20	66 / 63	129	9	64 / 59	123	9
HAMLET R3	213 / 169	382	20	54 / 60	114	9	52 / 61	113	9
ExplaGraphs	1335 / 1033	2368	53	199 / 199	398	9	200 / 200	400	9

Table 1: ExplaGraphs dataset statistics: S = Support, C = Counter, Topics = Number of spanning topics.

	#N	#E	#EN	D	%Non-linear	%EN
Train	5.1	4.2	1.3	3.3	58.8	78.2
Dev	5.4	4.5	1.6	3.8	47.0	88.4
Test	5.2	4.3	1.4	3.3	63.9	78.4
Total	5.2	4.3	1.3	3.4	58.6	79.4

Table 2: Graph Statistics: #N, #E, #EN = Average number of nodes, edges and external nodes respectively. D = Average depth of graphs. %Non-Linear = percentage of graphs which are not linear chains. % EN = percentage of graphs with external node(s) in the graph.

(67%, 81% and 90% after rounds 1, 2 and 3 respectively). Our *Create-Verify-And-Refine* framework is generic and allows for iterative improvement of graphs. See appendix for various quality control mechanisms for complex graph collection, which we believe will be helpful for similar future efforts.

5 Dataset Analysis

ExplaGraphs consists of a total of 3166 samples (see Table 1).[2] We collect two graphs for each sample in the test set. Table 2 shows statistics concerning the average number of nodes, edges, and external commonsense nodes present in our graphs. Approximately, 79% of graphs contain external nodes, indicating that most of our samples require background commonsense knowledge to explicitly support or refute a belief. Additionally, our graphs have diverse reasoning structures, with 58% of non-linear graphs. A large presence of non-linear structures and an average depth of 4 indicates complex reasoning involved in our task. We also find that the most frequently used relations are causal (like "capable of", "causes", "desires", and their negative counterparts), which further supports our graphs as explanations (details in appendix).

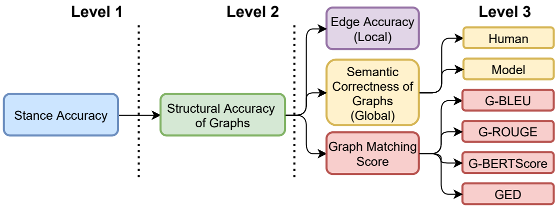


Figure 4: Our multi-level evaluation framework.

6 Evaluation Metrics

Explanation graphs can be represented in multiple correct ways with varying levels of specificity and different graphical structures. A single concept can also be paraphrased differently. Thus, we design a 3-level evaluation pipeline (see Fig. 4).

Level 1 – Stance Accuracy (SA): All models for our task predict both the stance label and the commonsense explanation graph. In Level 1, we report the stance prediction accuracy which ensures that the explanation graph is consistent with the predicted stance. Samples with a correctly predicted stance are then passed to the next levels that check for the quality of the generated explanation graphs.

Level 2 – Structural Correctness Accuracy of Graphs (StCA): As per our task definition in Sec. 3, for an explanation graph to be correct, it first has to be structurally correct. Hence, we compute the fraction of structurally correct graphs (connected DAGs with at least three edges and at least two concepts from the belief and at least two from the argument). Samples with correct stances and structurally correct graphs are then evaluated in Level 3 for: (1) semantic correctness, (2) match with GT graphs, and (3) edge importance.

Level 3 – Semantic Correctness Accuracy of Graphs (SeCA): Identifying semantic correctness of a graph requires following our human verification process discussed in Sec. 4.2. A graph is semantically correct if all its edges are semantically coherent and given the belief, the unambiguously inferred stance from the graph matches the original stance. However, both these aspects are

[2] Like prior structured data collection efforts (Geva et al., 2021), graph collection is challenging due to the difficulty in training annotators to create (connected/acyclic) graphs and verifying them for semantic consistency and stance inference.

challenging because they require understanding the underlying semantics and reasoning through the graph. Carrying this out by humans at a large scale is also expensive. Thus, following previous works (Zhang* et al., 2020; Sellam et al., 2020; Pruthi et al., 2020), we propose an automatic model-based metric that given a belief-graph pair, predicts the label between incorrect, support, and counter. Specifically, we fine-tune RoBERTa (Liu et al., 2019) on the beliefs and corresponding human-verified graphs from our data collection phase. Graphs are fed as concatenated edges to the model. Since the space of incorrect graphs is potentially huge, we augment our training data with synthetically created incorrect graphs (by randomly adding, removing, or replacing edges) from already correct (support/counter) graphs. Note that our automatic metric is not meant to replace human evaluation. Thus, for completeness, we still perform human evaluation and show human-metric correlation for SeCA (Sec. 8).

Level 3 – G-BERTScore (G-BS): We also introduce a matching metric that quantifies the degree of match between the ground-truth and the predicted graphs. We call this G-BERTScore, designed as an extension of a text generation metric, BERTScore (Zhang* et al., 2020) for graph-matching. We consider graphs as a set of edges and solve a matching problem that finds the best assignment between the edges in the gold graph and those in the predicted graph. Each edge is treated as a sentence and the scoring function between a pair of gold and predicted edges is given by BERTScore.[3] Given the best assignment and the overall matching score, we compute precision, recall and report F1 as our G-BERTScore metric. On the test set, we consider the best match across all ground-truth graphs.

Level 3 – Graph Edit Distance (GED): As a more interpretable graph matching metric, we use Graph Edit Distance (Abu-Aisheh et al., 2015) to compute the distance between the predicted graph and the gold graph. Formally, GED measures the number of edit operations (addition, deletion, and replacement of nodes and edges) for transforming the predicted graph to a graph isomorphic to the gold graph. The cost of each edit operation is chosen to be 1. The GED for each sample is normalized between 0 and 1 by an appropriate normalizing constant (upper bound of GED). Thus, the samples with either incorrect stances or structurally incorrect graphs will have a maximum normalized GED of 1 while samples whose graphs match exactly will have a score of 0. The overall GED is given by the average of the sample-wise GEDs. Lower GED indicates that the predicted graphs match more closely with the gold graphs.

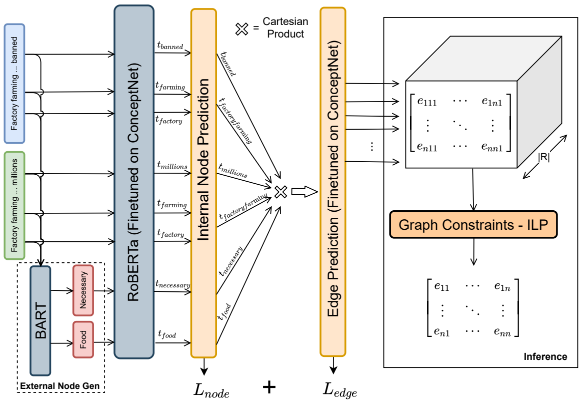


Figure 5: Our Commonsense-Augmented Structured Prediction Model for explanation graph generation.

Level 3 – Edge Importance Accuracy (EA): While SeCA assesses the correctness of a graph at a global level, we also propose a local model-based metric, named "Edge Importance Accuracy" which computes the macro-average of important edges in the predicted graphs. An edge is defined as important if not having it as part of the graph causes a decrease in the model's confidence for the target stance. We first fine-tune a RoBERTa model that given a *(belief, argument, graph)* triple, predicts the probability of the target stance. Next, we remove one edge at a time from the corresponding graph and query the same model with the belief, argument and the graph but with the edge removed. If we observe a drop in the model's confidence for the target stance, the edge is considered important.

7 Models

Following prior work on explanation generation (Rajani et al., 2019), we experiment with two broad families of models – (1) **Reasoning** (First-Graph-Then-Stance) models that first predict the explanation graph by conditioning on the belief and the argument. Then it augments the belief and the argument with the generated graph to predict the stance, (2) **Rationalizing** (First-Stance-Then-Graph) mod-

[3] We choose BERTScore over BLEU or ROUGE because they have been shown to correlate poorly with humans for prior natural language explanation studies (Camburu et al., 2018; Marasović et al., 2020). However, for completeness sake, our code reports them.

	SA↑	StCA↑	SeCA↑	G-BS↑	GED↓	EA↑	Hu↑
RE-BA	68.5	18.7	11.2	15.6	0.86	10.0	5.7
RA-BA	**87.2**	25.7	13.0	22.0	0.81	12.8	8.5
RE-T5	69.0	32.5	13.5	28.3	0.75	17.3	8.7
RA-T5	**87.2**	38.7	19.0	33.6	0.71	20.8	10.5
RE-SP	72.5	**62.5**	**20.0**	**50.0**	**0.60**	**26.2**	**12.5**
UB	91.0	91.0	83.5	71.1	0.38	46.8	80.3

Table 3: Results of our models across all metrics on EXPLAGRAPHS test set. UB = Metric Upper Bound, Hu = Human verification of semantic correctness of graphs.

els that first predict the stance, followed by generating graphs as post-hoc explanations. In both these types of models, the stance prediction happens through a fine-tuned RoBERTa. For graph generation, we first propose a commonsense-augmented structured model (described next). We also experiment with state-of-the-art text generation models like BART and T5 that generate graphs as linearized strings. During training, edges in the graphs are ordered according to the depth-first-traversal (DFS) order of the nodes. See appendix for details on fine-tuning BART and T5 for graph generation.

Commonsense-Augmented Structured Prediction Model: Next, as another baseline, we present a commonsense-augmented structured prediction model. As shown in Fig. 5, it has the following four modules: (a) *Internal Nodes Prediction:* It identifies the concepts (nodes) from the belief or the argument. We pose this task as a sequence-tagging problem where given a sequence of tokens from the belief and argument, each token is classified into one of the three classes {B-N, I-N, O} denoting the beginning, inside and outside of a node respectively. We build this module on top of a pre-trained RoBERTa (Liu et al., 2019) by feeding in the concatenated belief and argument and having a standard 2-layer classifier at the top. (b) *External Commonsense Nodes Generation:* We build this module separately by fine-tuning a pre-trained BART (Lewis et al., 2019) model that conditions on the concatenated belief and argument and generates a sequence of commonsense concepts. (c) *Edge Prediction:* We pose this as a multi-way classification problem in which given a pair of nodes, the module has to classify the edge into one relation (or no edge). This module is conditioned on the node prediction module to enable learning edges between the set of chosen nodes only and also for optimizing both modules jointly. Specifically, given the set of node representations from the node module, we construct the edge representations for all possible edges which are then passed to a 2-layer classifier for prediction. To augment our model with external commonsense, we first fine-tune RoBERTa and the edge module on ConceptNet (Liu and Singh, 2004). (d) *Enforcing Graph Constraints:* Our final loss sums the cross-entropy losses from the node and edge module. Following Saha et al. (2020), during inference, we ensure connectivity and acyclicity in the explanation graph through an Integer Linear Program. See appendix Sec. C.3 for a more formal description of the model.[4]

8 Experiments and Analysis

In Table 3, we compare our Reasoning-SP (RE-SP) model that generates graphs using the structured model with Rationalizing-BART/T5 (RA-BART/T5) and Reasoning-BART/T5 (RE-BART/T5) models that generate graphs using BART/T5. Besides our automatic metrics, the last column shows human evaluation of semantic correctness of graphs. Below, we summarize our key findings.

SP vs BART/T5: BART and T5, used out-of-the-box, fail to generate a high percentage of structurally correct graphs (StCA) due to the lack of explicit constraints. Overall, RE-SP is the best performing model across all automatic metrics and human evaluation. It obtains a much higher StCA due to the constraints-enforcing ILP module and eventually a higher SeCA. Its superior performance is also reflected through the other metrics (G-BS, GED, and EA). See appendix for more analysis (like per-

[4] Our EXPLAGRAPHS task also encourages future work based on other related structured models such as deep generative models for graphs (You et al., 2018; Simonovsky and Komodakis, 2018; Grover et al., 2019; Liao et al., 2019; Shi et al., 2020), but with adaptation to the unique challenges involved in our task, e.g., learning a good representation of the context using some pre-trained language model, identifying the internal nodes, generating the external/commonsense nodes and inferring the relations between nodes.

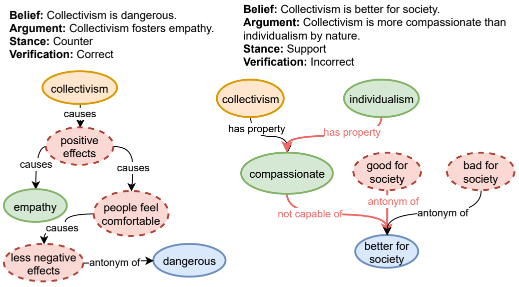


Figure 6: Predicted graphs from the RE-SP model. The first graph is correct, while the second one is not.

formance at varying reasoning depths, structures and the effect of edge ordering on BART/T5).

Explanation Impact (RA vs RE): RA models predict the stance first without the graph, while the RE models predict the stance conditioned on the generated graph. RE models' drop in SA points to their overall limitations in generating helpful explanations. In fact, conditioning on such graphs makes the model less confident of its stance predictions.

Metrics' Upper Bound: While the stance accuracy (SA) is sufficiently high for all models, they obtain a significantly low semantic correctness accuracy (SeCA) for graphs (between 10-20%). To obtain an upper bound on our metrics (last row), we treat ground-truth graphs as predictions and find that they not only aid in stance prediction (SA increases from 87% to 91%) but also obtain a high 83% SeCA. Given the large gap (>60%) between human and model performance, we hope our dataset will encourage future work on better model development for explanation graph generation.

Human-Metric Correlation for SeCA: While we develop an initial automatic metric for SeCA, it still is a challenging problem and hence human evaluation for the same is necessary. In order to show human-metric correlation for SeCA, we perform human evaluation (using the exact mechanism of human-written graph verification, discussed in Sec. 4.2) of all structurally-correct generated graphs. Encouragingly, we find that our model-based metric (SeCA column) correlates well with humans (last column), with RE-SP being the best model. The human verification labels match with the SeCA model's predictions 68% of the time.

Analysis of Generated Graphs: Fig. 6 shows two randomly chosen graphs generated by RE-SP containing external commonsense nodes like "positive effects", "good for society". While the first graph is correct, the second graph chooses the wrong relations for certain edges (in red), thus pointing to its lack of commonsense. Overall, we find a large fraction of incorrect graphs contain incoherent facts or facts not adhering to human commonsense. Table 4 shows that RE-SP generates more nodes, edges, external nodes and non-linear structures, due to its individual components.

	#N	#E	#EN	D	%NL	%EN
RA-T5	4.3	3.3	0.3	3.3	3.5	24.8
RE-T5	4.4	3.4	0.3	3.3	7.6	28.8
RE-SP	6.2	5.2	2.2	2.6	99.6	97.6

Table 4: Statistics for the generated explanation graphs. NL = Non-Linear graphs, EN = External Nodes.

9 Discussion and Future Work

We show the promise of explanation graphs by considering the task of stance prediction as a motivating use-case because it is representative of many sentence-pair inference tasks (consider the belief as the premise, argument as the hypothesis and the support/counter labels as entailment/contradiction). We believe that our definition of explanation graphs (Sec. 3) is quite generic and should extend naturally to any NLU task, e.g., the internal nodes are concepts that are part of a context (context could mean premise-hypothesis for NLI, passage for sentiment classification, passage-question for QA, etc), the external nodes refer to concepts that are not part of the context, the edges are semantic relations between concepts, and the DAG-like constraints ensure the presence of explicit reasoning structures. Although we choose a pre-defined set of relations for our task that can adequately represent most commonsense facts, the relations can be updated/adapted for a different task. Given the potential of explanation graphs in improving the explainability of many reasoning tasks, we hope future work can further explore their applicability in different scenarios.

10 Conclusion

We proposed ExplaGraphs, a new *generative* and *structured* commonsense-reasoning task (and a benchmarking dataset) on explanation graph generation for stance prediction. Additionally, we proposed automatic evaluation metrics and an initial structured model for ExplaGraphs, demonstrating its difficulty in generating high-quality commonsense-augmented graphical explanations, and encouraging future work on better graph-based commonsense explanation generation.

Ethical Considerations

We select crowdworkers from Amazon Mechanical Turk (AMT) who are located in the US and Australia with a HIT approval rate higher than 96% and at least 1000 HITs approved. To ensure high data quality, we perform multiple on-boarding tests (details in the appendix) and manually verify a lot of the initial explanation graphs. We also provide personal feedback to a number of annotators. A total of 198 workers took part in our data collection and human verification process. We compensated annotators at the rate of $12-15 per hour. The payments per HIT for each of our tasks are listed in the appendix. To estimate this, we first post small pilot studies to evaluate average time of completion, and pay users accordingly. Annotators who annotated high-quality graphs were regularly compensated with bonuses, throughout the duration of our data collection process. Also, our dataset mostly reflects the views of a set of English-speaking US annotators about some of the debate topics. However, for completeness, we collect both support and counter sides of the arguments. While some of the beliefs may span controversial topics, we as authors do not promote or stand with either side of the argument. Instead, we focus on the explainability aspect of these arguments through background commonsense knowledge.

Acknowledgements

We thank the reviewers as well as Yejin Choi, Peter Clark, Peter Hase, Hyounghun Kim, and Jie Lei for their helpful feedback, and the annotators for their time and effort. This work was supported by DARPA MCS Grant N66001-19-2-4031, NSF-CAREER Award 1846185, Microsoft Investigator Fellowship, Munroe & Rebecca Cobey Fellowship, and an NSF Graduate Research Fellowship. The views in this article are those of the authors and not the funding agency.

References

Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. Protoqa: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136.

Faeze Brahman, Vered Shwartz, Rachel Rudinger, and Yejin Choi. 2021. Learning to rationalize for nonmonotonic reasoning with distant supervision. In *AAAI*.

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

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Maxwell Forbes, Jena D Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. 2019. Are you convinced? choosing the more convincing evidence with a siamese network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 967–976.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. 2019. A large-scale dataset for argument quality ranking: Construction and analysis. *arXiv preprint arXiv:1911.11408*.

Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*, pages 2434–2444. PMLR.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 107–112. Association for Computational Linguistics (ACL).

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1589–1599.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. A survey on stance detection for mis-and disinformation identification. *arXiv preprint arXiv:2103.00242*.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 751–762.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2020. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. *arXiv preprint arXiv:2010.05953*.

Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. R4c: A benchmark for evaluating rc systems to get the right answer for the right reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6740–6750.

Peter Jansen and Dmitry Ustalov. 2019. Textgraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 63–77.

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *arXiv preprint arXiv:1802.03052*.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 137–150.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. 2015. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678.

Aditya Kalyanpur, Tom Breloff, David Ferrucci, Adam Lally, and John Jantos. 2020. Braid: Weaving symbolic and neural knowledge into coherent logical explanations. *arXiv preprint arXiv:2011.13354*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Tom Leighton and Satish Rao. 1999. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L Hamilton, David Duvenaud, Raquel Urtasun, and Richard S Zemel. 2019. Efficient graph generation with graph recurrent attention networks. *arXiv preprint arXiv:1910.00760*.

Bill Yuchen Lin, Frank F Xu, Kenny Zhu, and Seungwon Hwang. 2018. Mining cross-cultural differences and similarities in social media. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 709–719.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1823–1840.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. *arXiv preprint arXiv:2103.13009*.

Aman Madaan, Dheeraj Rajagopal, Yiming Yang, Abhilasha Ravichander, Eduard Hovy, and Shrimai Prabhumoye. 2020. Eigen: Event influence generation using pre-trained language models. *arXiv preprint arXiv:2010.11764*.

Ana Marasović, Chandra Bhagavatula, Jae sung Park, Ronan Le Bras, Noah A Smith, and Yejin Choi. 2020. Natural language rationales with full-stack visual reasoning: From pixels to semantic frames to commonsense graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2810–2829.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2020. Evaluating explanations: How much do explanations from the teacher aid students?

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(140):1–67.

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

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2021. multiPRover: Generating multiple proofs for improved interpretability in rule reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3662–3677.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019b. Socialiqa: Commonsense reasoning about social interactions. In *Conference on Empirical Methods in Natural Language Processing*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. 2020. Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv preprint arXiv:2001.09382*.

Martin Simonovsky and Nikos Komodakis. 2018. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pages 412–422. Springer.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4020–4026.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. *arXiv preprint arXiv:2102.12060*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. Worldtree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5456–5473.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419.

Frank F Xu, Bill Yuchen Lin, and Kenny Zhu. 2018. Automatic extraction of commonsense locatednear knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 96–101.

Qinyuan Ye, Xiao Huang, Elizabeth Boschee, and Xiang Ren. 2020. Teaching machine comprehension with compositional explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1599–1615.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pages 5708–5717. PMLR.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. 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 4085–4094.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. Winowhy: A deep diagnosis of essential commonsense knowledge for answering winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745.

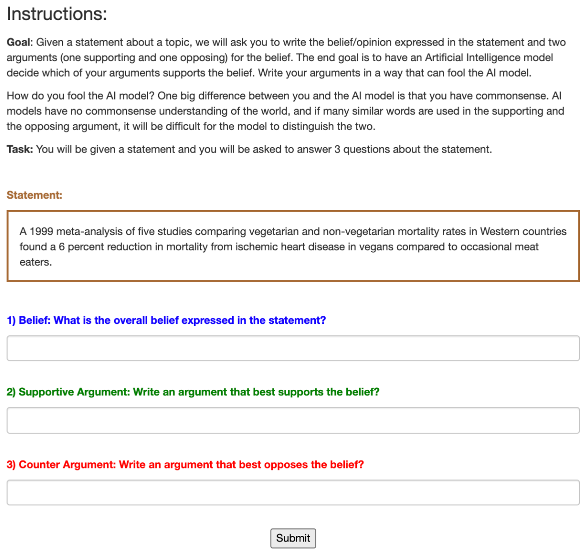
Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Data Collection

A.1 Stage 1: (Belief, Argument, Stance) Collection

Our stage 1 data collection consists of one pre-HAMLET round and three rounds of HAMLET (Nie et al., 2019).

Pre-HAMLET: The complete instructions for pre-HAMLET data collection is shown in Fig. 7. Briefly, annotators write the belief expressed in the



Instructions:

Goal: Given a statement about a topic, we will ask you to write the belief/opinion expressed in the statement and two arguments (one supporting and one opposing) for the belief. The end goal is to have an Artificial Intelligence model decide which of your arguments supports the belief. Write your arguments in a way that can fool the AI model.

How do you fool the AI model? One big difference between you and the AI model is that you have commonsense. AI models have no commonsense understanding of the world, and if many similar words are used in the supporting and the opposing argument, it will be difficult for the model to distinguish the two.

Task: You will be given a statement and you will be asked to answer 3 questions about the statement.

Statement:

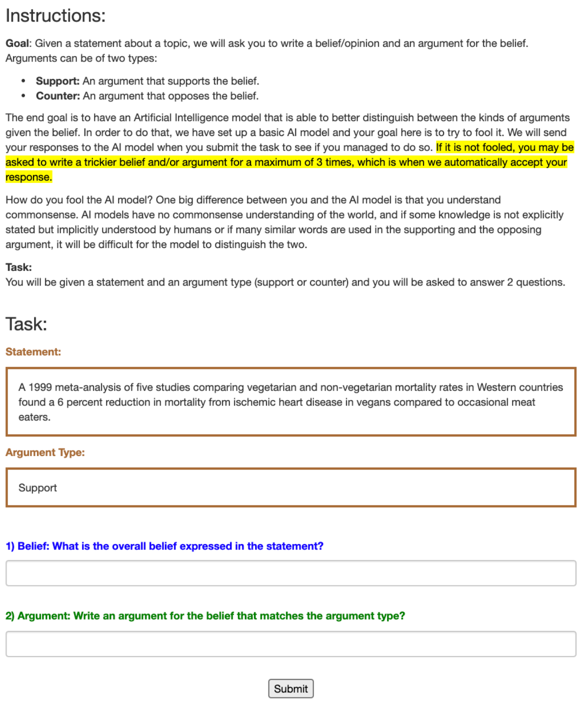
A 1999 meta-analysis of five studies comparing vegetarian and non-vegetarian mortality rates in Western countries found a 6 percent reduction in mortality from ischemic heart disease in vegans compared to occasional meat eaters.

1) Belief: What is the overall belief expressed in the statement?

2) Supportive Argument: Write an argument that best supports the belief?

3) Counter Argument: Write an argument that best opposes the belief?

Figure 7: Interface showing the instructions for collecting belief and argument (support and counter) pairs on MTurk for the pre-HAMLET stage, given a prompt about one of the debate topics.



Instructions:

Goal: Given a statement about a topic, we will ask you to write a belief/opinion and an argument for the belief. Arguments can be of two types:

- **Support:** An argument that supports the belief.
- **Counter:** An argument that opposes the belief.

The end goal is to have an Artificial Intelligence model that is able to better distinguish between the kinds of arguments given the belief. In order to do that, we have set up a basic AI model and your goal here is to try to fool it. We will send your responses to the AI model when you submit the task to see if you managed to do so. If it is not fooled, you may be asked to write a trickier belief and/or argument for a maximum of 3 times, which is when we automatically accept your response.

How do you fool the AI model? One big difference between you and the AI model is that you understand commonsense. AI models have no commonsense understanding of the world, and if some knowledge is not explicitly stated but implicitly understood by humans or if many similar words are used in the supporting and the opposing argument, it will be difficult for the model to distinguish the two.

Task:
You will be given a statement and an argument type (support or counter) and you will be asked to answer 2 questions.

Task:

Statement:

A 1999 meta-analysis of five studies comparing vegetarian and non-vegetarian mortality rates in Western countries found a 6 percent reduction in mortality from ischemic heart disease in vegans compared to occasional meat eaters.

Argument Type:

Support

1) Belief: What is the overall belief expressed in the statement?

2) Argument: Write an argument for the belief that matches the argument type?

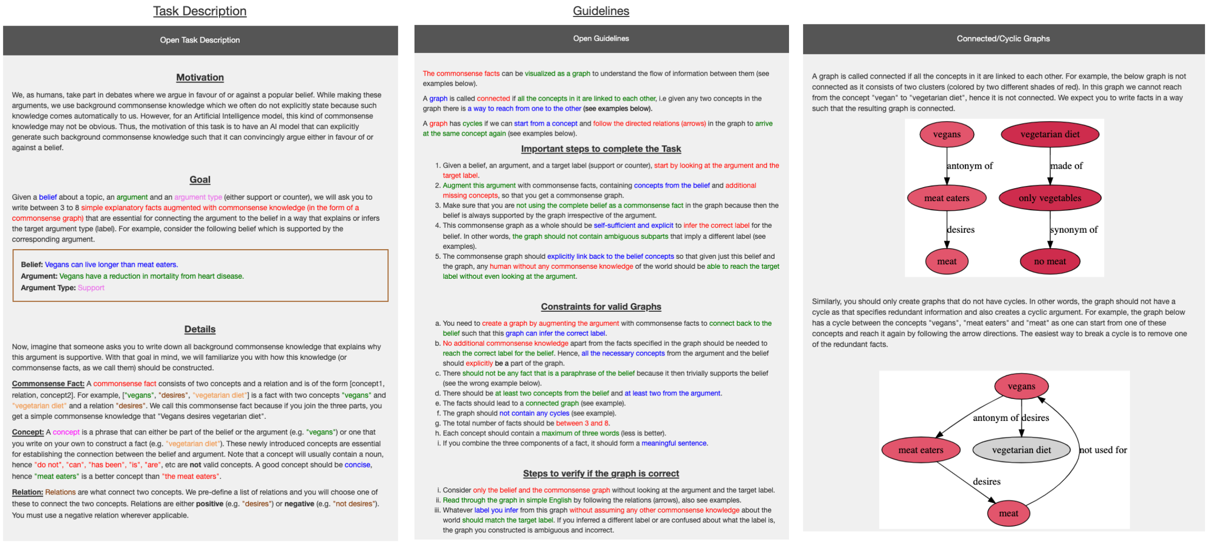
Figure 8: Interface showing the instructions for collecting belief and argument pairs on MTurk for the HAMLET stage, given a prompt about one of the debate topics and the target stance label (support or counter).

prompt along with a supporting and a counter argument. The beliefs and arguments are typically one-sentence long. We collect a total of 998 samples from randomly chosen 33 topics out of the 53 train topics with an average of 30 samples per topic. Note that we do not include the dev and test topics as part of the pre-HAMLET collection to ensure that the examples in these splits are sufficiently hard for the models.

HAMLET: We follow the initial pre-HAMLET collection round with 3 rounds of HAMLET collection to reduce any annotation artifacts and most importantly, collect harder examples with implicit background knowledge. Fig. 8 shows the instructions for the HAMLET rounds. At each round of HAMLET collection, we ask annotators to write (belief, argument) pairs in a way that a stance prediction model is fooled. In the first round, we start by fine-tuning a RoBERTa model (Liu et al., 2019) on the pre-HAMLET data that given a (belief, argument) pair predicts the stance label. After each round, we divide the collected HAMLET data into train, dev and test splits based on their respective topics and update the RoBERTa model by training on the pre-HAMLET data and the train splits of the HAMLET rounds collected so far. We collect data in each round from the remaining 38 topics (20 train, 9 dev, 9 test) equally. In contrast to the pre-HAMLET round, here we also provide the target stance label along with the prompt and annotators are asked to write the belief and an argument that adhere to the target label. Once they construct a pair, in real-time, it is sent to the stance prediction model and if the model is able to predict the stance correctly, we prompt the annotators to rewrite either the belief or the argument. We provide annotators 3 tries in Round 1 and 4 tries in Round 2 and Round 3 to fool the model, following which we accept the final pair. Our HAMLET collection comprises of a total of 2170 samples with 892, 667 and 611 samples in rounds 1, 2, and 3 respectively.

Quality Control: We apply the following mechanisms to control the quality of the collected data.

- **Onboarding Test:** Each annotator is required to successfully pass an onboarding quiz before they can start writing belief and argument pairs. In this test, we evaluate their understanding of supportive and counter arguments by providing them with 10 (belief, argument) pairs and they are asked to choose if the argument supports or counters the belief.
- **Stance Label Verification:** We verify the stance labels of all the examples collected in pre-HAMLET and HAMLET rounds. This is particularly necessary for the HAMLET rounds where the annotators are constrained to fool the model



Task Description

Open Task Description

Motivation

We, as humans, take part in debates where we argue in favour of or against a popular belief. While making these arguments, we use background commonsense knowledge which we often do not explicitly state because such knowledge comes automatically to us. However, for an Artificial Intelligence model, this kind of commonsense knowledge may not be obvious. Thus, the motivation of this task is to have an AI model that can explicitly generate such background commonsense knowledge such that it can convincingly argue either in favour of or against a belief.

Goal

Given a belief about a topic, an argument and an argument type (either support or counter), we will ask you to write between 3 to 8 simple explanatory facts augmented with commonsense knowledge (in the form of a commonsense graph) that are essential for connecting the argument to the belief in a way that explains or infers the target argument type (label). For example, consider the following belief which is supported by the corresponding argument.

Belief: Vegans can live longer than meat eaters.
Argument: Vegans have a reduction in mortality from heart disease.
Argument Type: Support

Details

Now, imagine that someone asks you to write down all background commonsense knowledge that explains why this argument is supportive. With that goal in mind, we will familiarize you with how this knowledge (or commonsense facts, as we call them) should be constructed.

Commonsense Fact: A commonsense fact consists of two concepts and a relation and is of the form [concept1, relation, concept2]. For example, ["vegans", "desires", "vegetarian diet"] is a fact with two concepts "vegans" and "vegetarian diet" and a relation "desires". We call this commonsense fact because if you join the three parts, you get a simple commonsense knowledge that "Vegans desires vegetarian diet".

Concept: A concept is a phrase that can either be part of the belief or the argument (e.g. "vegans") or one that you write on your own to construct a fact (e.g. "vegetarian diet"). These newly introduced concepts are essential for establishing the connection between the belief and argument. Note that a concept will usually contain a noun, hence "do not", "can", "has been", "is", "are", etc are **not** valid concepts. A good concept should be concise, hence "meat eaters" is a better concept than "the meat eaters".

Relation: Relations are what connect two concepts. We pre-define a list of relations and you will choose one of these to connect the two concepts. Relations are either **positive** (e.g. "desires") or **negative** (e.g. "not desires"). You must use a negative relation wherever applicable.

Guidelines

Open Guidelines

The commonsense facts can be visualized as a graph to understand the flow of information between them (see examples below).

A graph is called connected if all the concepts in it are linked to each other, i.e given any two concepts in the graph there is a way to reach from one to the other (see examples below).

A graph has cycles if we can start from a concept and follow the directed relations (arrows) in the graph to arrive at the same concept again (see examples below).

Important steps to complete the Task

1. Given a belief, an argument, and a target label (support or counter), start by looking at the argument and the target label.
2. Augment this argument with commonsense facts, containing concepts from the belief and additional missing concepts, so that you get a commonsense graph.
3. Make sure that you are not using the complete belief as a commonsense fact in the graph because then the belief is always supported by the graph irrespective of the argument.
4. This commonsense graph as a whole should be self-sufficient and explicit to infer the correct label for the belief. In other words, the graph should not contain ambiguous subparts that imply a different label (see examples).
5. The commonsense graph should explicitly link back to the belief concepts so that given just this belief and the graph, any human without any commonsense knowledge of the world should be able to reach the target label without even looking at the argument.

Constraints for valid Graphs

a. You need to create a graph by augmenting the argument with commonsense facts to connect back to the belief such that this graph can infer the correct label.
b. No additional commonsense knowledge apart from the facts specified in the graph should be needed to reach the correct label for the belief. Hence, all the necessary concepts from the argument and the belief should explicitly **be a** part of the graph.
c. There should not be any fact that is a paraphrase of the belief because it then trivially supports the belief (see the wrong example below).
d. There should be at least two concepts from the belief and at least two from the argument.
e. The facts should lead to a connected graph (see example).
f. The graph should not contain any cycles (see example).
g. The total number of facts should be between 3 and 8.
h. Each concept should contain a maximum of three words (less is better).
i. If you combine the three components of a fact, it should form a meaningful sentence.

Steps to verify if the graph is correct

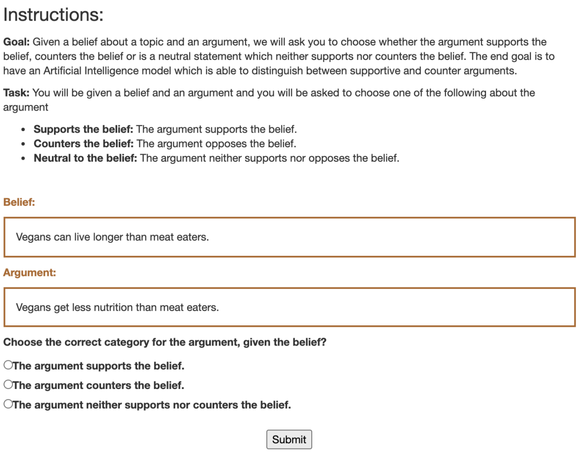
i. Consider only the belief and the commonsense graph without looking at the argument and the target label.
ii. Read through the graph in simple English by following the relations (arrows), also see examples.
iii. Whatever label you infer from this graph without assuming any other commonsense knowledge about the world should match the target label. If you inferred a different label or are confused about what the label is, the graph you constructed is ambiguous and incorrect.

Connected/Cyclic Graphs

A graph is called connected if all the concepts in it are linked to each other. For example, the below graph is not connected as it consists of two clusters (colored by two different shades of red). In this graph we cannot reach from the concept "vegan" to "vegetarian diet", hence it is not connected. We expect you to write facts in a way such that the resulting graph is connected.

Similarly, you should only create graphs that do not have cycles. In other words, the graph should not have a cycle as that specifies redundant information and also creates a cyclic argument. For example, the graph below has a cycle between the concepts "vegans", "meat eaters" and "meat" as one can start from one of these concepts and reach it again by following the arrow directions. The easiest way to break a cycle is to remove one of the redundant facts.

Figure 9: Instructions for commonsense explanation graph creation: We start by explaining the overall motivation and goal of this task, followed by the definitions of commonsense fact, concept, and relation. As part of the guidelines, we provide the detailed steps to perform this task and the list of structural constraints on the explanation graphs. We also remind the workers to verify their own graphs before submitting by following three basic steps of stance inference from the graphs. Since workers are required to fix their graphs if they are not connected DAGs, we also provide examples of disconnected and cyclic graphs.



Instructions:

Goal: Given a belief about a topic and an argument, we will ask you to choose whether the argument supports the belief, counters the belief or is a neutral statement which neither supports nor counters the belief. The end goal is to have an Artificial Intelligence model which is able to distinguish between supportive and counter arguments.

Task: You will be given a belief and an argument and you will be asked to choose one of the following about the argument

- **Supports the belief:** The argument supports the belief.
- **Counters the belief:** The argument opposes the belief.
- **Neutral to the belief:** The argument neither supports nor opposes the belief.

Belief:

Vegans can live longer than meat eaters.

Argument:

Vegans get less nutrition than meat eaters.

Choose the correct category for the argument, given the belief?

○ **The argument supports the belief.**
○ **The argument counters the belief.**
○ **The argument neither supports nor counters the belief.**

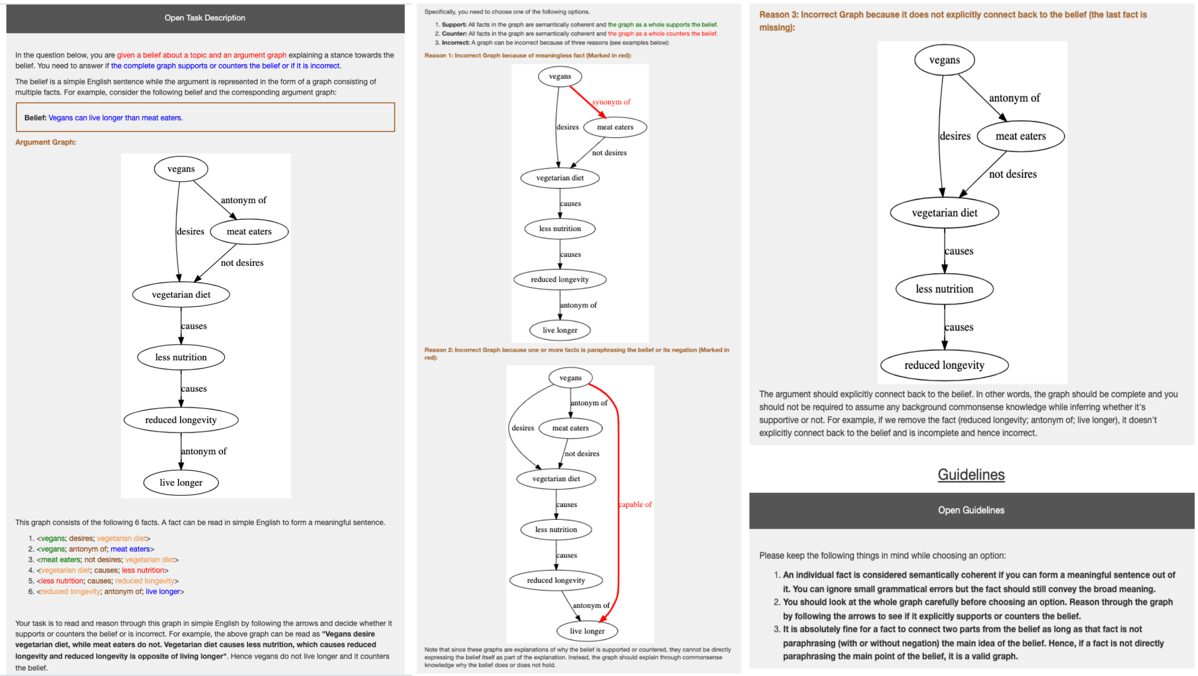
Figure 10: Interface showing the instructions for verifying the stance labels for belief and argument pairs on MTurk. We keep only those pairs which have majority stance label support or counter across five verifiers.

and it is hard to create such samples and hence verification is required. Fig. 10 shows the interface for our stance label verification, given the belief and the argument. For each (belief, argument) pair, we ask five annotators to choose the correct label between "support", "counter", and "neutral". We choose the majority label as the final label and keep only those examples that have majority labels either "support" or "counter".

A.2 Stage 2: Commonsense Explanation Graph Collection

Graph Creation: Fig. 9 shows the detailed instructions provided to the annotators for commonsense explanation graph creation. We start by explaining the overall motivation and the goal of our task, followed by the definitions of commonsense fact, concept, and relation. As part of the guidelines, we provide the detailed steps to perform this task and the list of structural constraints on the explanation graphs. We remind the workers to verify their own graphs before submitting, by following three basic steps of stance inference from the graphs. We also provide examples of disconnected and cyclic graphs to help them understand structurally incorrect graphs.

Graph Verification: In Fig. 11, we show the instructions provided for verifying the semantic correctness of our commonsense explanation graphs. In this stage, we refer to explanation graphs as argument graphs since our graphs are extended structured arguments. We provide annotators will only the belief and the argument graph, and ask them to choose between incorrect, support and counter labels. We also provide examples of semantically incorrect graphs. Fig. 12 shows the interface for graph verification.



Open Task Description

In the question below, you are given a belief about a topic and an argument graph explaining a stance towards the belief. You need to answer if the complete graph supports or counters the belief or if it is incorrect.

The belief is a simple English sentence while the argument is represented in the form of a graph consisting of multiple facts. For example, consider the following belief and the corresponding argument graph:

Belief: Vegans can live longer than meat eaters.

Argument Graph:

This graph consists of the following 6 facts. A fact can be read in simple English to form a meaningful sentence.

1. <vegans; desires; vegetarian diet>
2. <vegans; antonym of; meat eaters>
3. <meat eaters; not desires; vegetarian diet>
4. <vegetarian diet; causes; less nutrition>
5. <less nutrition; causes; reduced longevity>
6. <reduced longevity; antonym of; live longer>

Your task is to read and reason through this graph in simple English by following the arrows and decide whether it supports or counters the belief or is incorrect. For example, the above graph can be read as **"Vegans desire vegetarian diet, while meat eaters do not. Vegetarian diet causes less nutrition, which causes reduced longevity and reduced longevity is opposite of living longer"**. Hence vegans do not live longer and it counters the belief.

Specifically, you need to choose one of the following options.

1. **Support:** All facts in the graph are semantically coherent and the graph as a whole supports the belief.
2. **Counter:** All facts in the graph are semantically coherent and the graph as a whole counters the belief.
3. **Incorrect:** A graph can be incorrect because of three reasons (see examples below):

Reason 1: Incorrect Graph because of meaningless fact (Marked in red):

Reason 2: Incorrect Graph because one or more facts is paraphrasing the belief or its negation (Marked in red):

Note that since these graphs are explanations of why the belief is supported or countered, they cannot be directly expressing the belief itself as part of the explanation. Instead, the graph should explain through commonsense knowledge why the belief does or does not hold.

Reason 3: Incorrect Graph because it does not explicitly connect back to the belief (the last fact is missing):

The argument should explicitly connect back to the belief. In other words, the graph should be complete and you should not be required to assume any background commonsense knowledge while inferring whether it's supportive or not. For example, if we remove the fact (reduced longevity; antonym of; live longer), it doesn't explicitly connect back to the belief and is incomplete and hence incorrect.

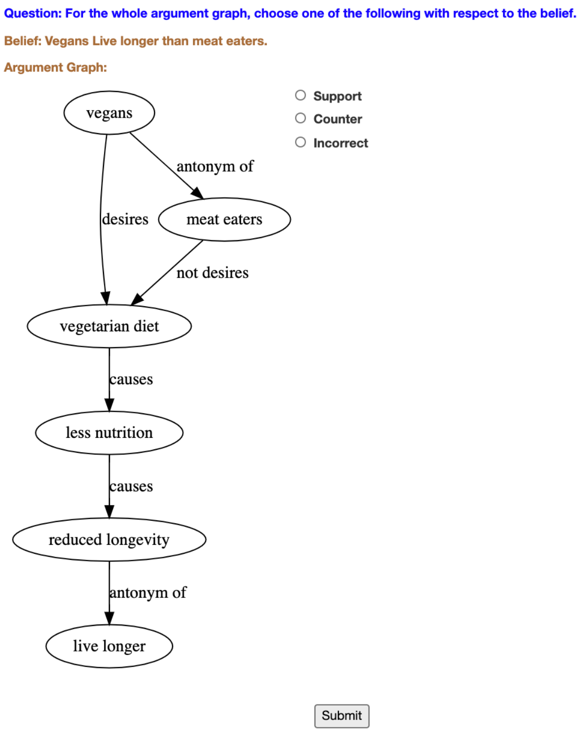
Guidelines

Open Guidelines

Please keep the following things in mind while choosing an option:

1. **An individual fact is considered semantically coherent if you can form a meaningful sentence out of it. You can ignore small grammatical errors but the fact should still convey the broad meaning.**
2. **You should look at the whole graph carefully before choosing an option. Reason through the graph by following the arrows to see if it explicitly supports or counters the belief.**
3. **It is absolutely fine for a fact to connect two parts from the belief as long as that fact is not paraphrasing (with or without negation) the main idea of the belief. Hence, if a fact is not directly paraphrasing the main point of the belief, it is a valid graph.**

Figure 11: Instructions for commonsense graph verification: Explanation graphs are treated as augmented structured arguments for this task and hence referred to as argument graphs. Given a belief and the argument graph, workers are required to choose between incorrect, support and counter labels. We begin by visually explaining what an argument graph is, and also show examples of incorrect graphs. To ensure good inter-annotator agreement and that the semantically incorrect graphs are identified correctly, we also provide some general guidelines for performing this task.



Question: For the whole argument graph, choose one of the following with respect to the belief.

Belief: Vegans Live longer than meat eaters.

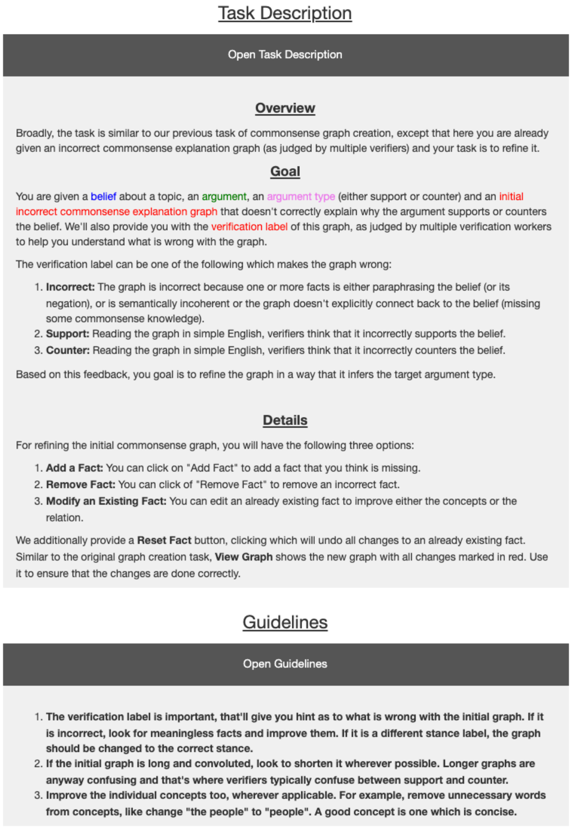
Argument Graph:

Figure 12: Interface for graph verification.

Graph Refinement: In Fig. 13, we show the instructions of graph refinement in which we also provide some broad guidelines of how to refine the graphs. Our refinement interface is shown in Figure 14. They refine the initial graph by adding, removing or replacing facts and the "View Graph" button shows the updated graph, with the changes marked in red.

Quality Control: Quality control of crowd-sourced data is challenging, more so when the task involves creating graphs with associated constraints like acyclicity, connectivity, etc and then reasoning through the graph to infer the target label. Verifying these graphs for completeness, semantic coherence and non-triviality also requires understanding the overall motivation of the underlying task and hence is significantly more challenging than our Stage 1 stance label verification. In the light of these challenges, we employ carefully designed quality control mechanisms, which we believe will be helpful for similar graph collection tasks in the future.

- **2-level Onboarding Test:** Since the three stages of graph creation, verification and refinement are closely tied to one another, we choose a single pool of annotators to perform all the graph-



Task Description

Open Task Description

Overview

Broadly, the task is similar to our previous task of commonsense graph creation, except that here you are already given an incorrect commonsense explanation graph (as judged by multiple verifiers) and your task is to refine it.

Goal

You are given a belief about a topic, an argument, an argument type (either support or counter) and an initial incorrect commonsense explanation graph that doesn't correctly explain why the argument supports or counters the belief. We'll also provide you with the verification label of this graph, as judged by multiple verification workers to help you understand what is wrong with the graph.

The verification label can be one of the following which makes the graph wrong:

1. **Incorrect:** The graph is incorrect because one or more facts is either paraphrasing the belief (or its negation), or is semantically incoherent or the graph doesn't explicitly connect back to the belief (missing some commonsense knowledge).
2. **Support:** Reading the graph in simple English, verifiers think that it incorrectly supports the belief.
3. **Counter:** Reading the graph in simple English, verifiers think that it incorrectly counters the belief.

Based on this feedback, you goal is to refine the graph in a way that it infers the target argument type.

Details

For refining the initial commonsense graph, you will have the following three options:

1. **Add a Fact:** You can click on "Add Fact" to add a fact that you think is missing.
2. **Remove Fact:** You can click of "Remove Fact" to remove an incorrect fact.
3. **Modify an Existing Fact:** You can edit an already existing fact to improve either the concepts or the relation.

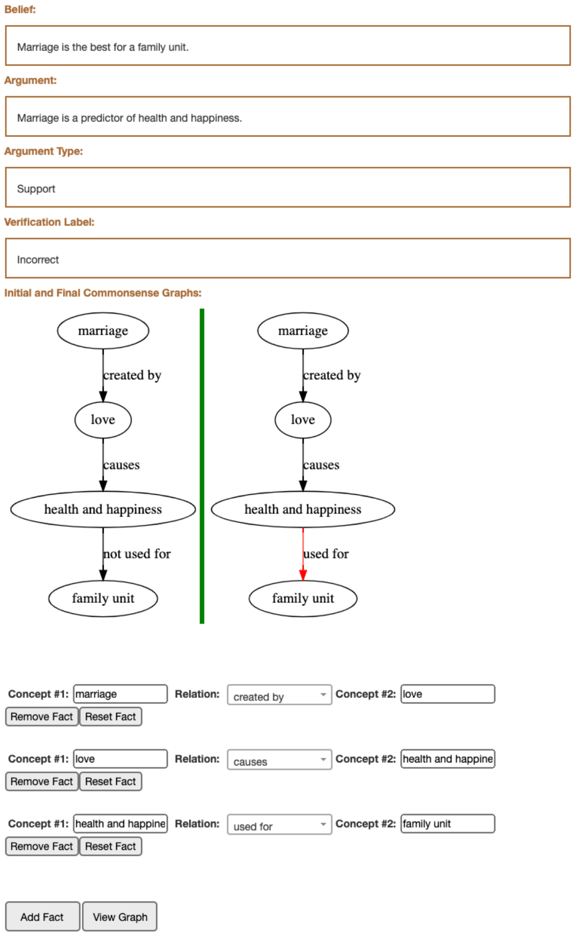
We additionally provide a **Reset Fact** button, clicking which will undo all changes to an already existing fact. Similar to the original graph creation task, **View Graph** shows the new graph with all changes marked in red. Use it to ensure that the changes are done correctly.

Guidelines

Open Guidelines

1. **The verification label is important, that'll give you hint as to what is wrong with the initial graph. If it is incorrect, look for meaningless facts and improve them. If it is a different stance label, the graph should be changed to the correct stance.**
2. **If the initial graph is long and convoluted, look to shorten it wherever possible. Longer graphs are anyway confusing and that's where verifiers typically confuse between support and counter.**
3. **Improve the individual concepts too, wherever applicable. For example, remove unnecessary words from concepts, like change "the people" to "people". A good concept is one which is concise.**

Figure 13: Instructions for graph refinement.



Belief:

Marriage is the best for a family unit.

Argument:

Marriage is a predictor of health and happiness.

Argument Type:

Support

Verification Label:

Incorrect

Initial and Final Commonsense Graphs:

Concept #1: marriage **Relation:** created by **Concept #2:** love

Remove Fact | Reset Fact

Concept #1: love **Relation:** causes **Concept #2:** health and happine

Remove Fact | Reset Fact

Concept #1: health and happine **Relation:** used for **Concept #2:** family unit

Remove Fact | Reset Fact

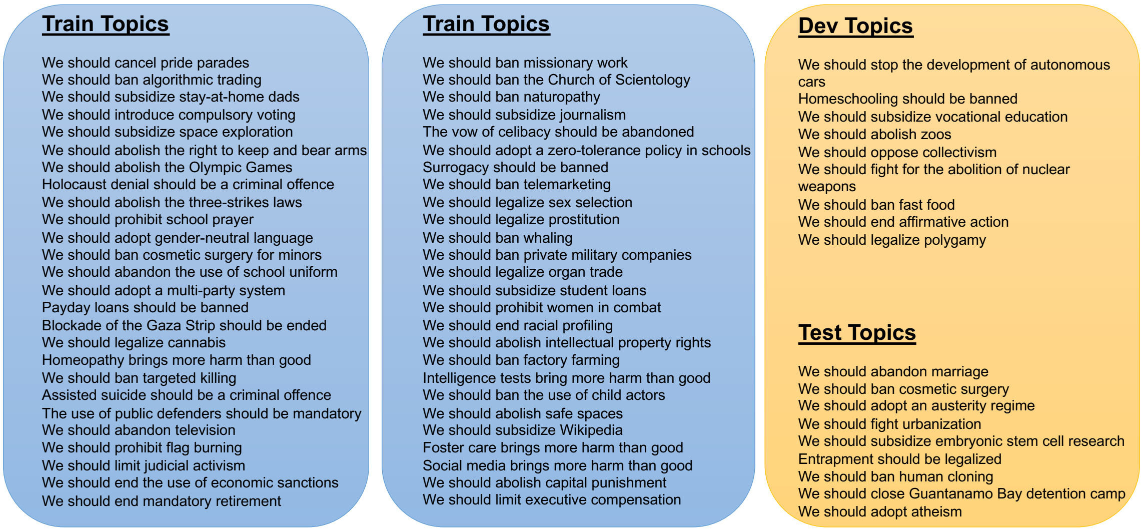
Add Fact | View Graph

Figure 14: Interface for commonsense explanation graph refinement: Annotators are provided with the belief, argument, the stance label, the initial incorrect explanation graph and the majority verification label. They refine the graph by adding, removing or replacing facts and the changes to the initial graph are shown in red.

related tasks. We also prohibit annotators from verifying their own graphs. We design a 2-level onboarding test where in the first level, we test the annotators' understanding of a commonsense fact because that is the basic building block of our graphs. Annotators are tested on 10 multiple choice questions, half of which require choosing the correct relation given the two concepts and another half require choosing the right pair of concepts, given the relation. Successful annotators from the first level qualify for the second level, where they are required to take two other tests. In one, we ask them to create a graph given a (belief, argument, stance) triple, whose quality we manually verify and in another, we ask them to verify the correctness of some already provided explanation graphs.

- **Intensive Training and Feedback:** We begin by providing detailed feedback and explanations of the correct answers from the onboarding tests to every qualified annotator. Every new annotator who starts creating graphs for the first time is initially requested to submit only a small number of graphs. We then verify these graphs manually and provide detailed feedback and suggest improvements wherever there are some incoherent facts in the graph or the graph is a trivial explanation or is incomplete. Over time, we find such personal feedback to be highly effective towards improving the quality of the graphs.

- **High-performing annotators for Refinement:** While it is theoretically possible to run multiple iterations of graph verification and refinement, under most practical scenarios due to time and budget constraints, we want to ensure that a few rounds of refinement is enough to obtain a high percentage of correct graphs. Hence, we qualify only the high-performing annotators (whose graphs have been verified as correct the most) for our refinement task.



Train Topics

We should cancel pride parades
We should ban algorithmic trading
We should subsidize stay-at-home dads
We should introduce compulsory voting
We should subsidize space exploration
We should abolish the right to keep and bear arms
We should abolish the Olympic Games
Holocaust denial should be a criminal offence
We should abolish the three-strikes laws
We should prohibit school prayer
We should adopt gender-neutral language
We should ban cosmetic surgery for minors
We should abandon the use of school uniform
We should adopt a multi-party system
Payday loans should be banned
Blockade of the Gaza Strip should be ended
We should legalize cannabis
Homeopathy brings more harm than good
We should ban targeted killing
Assisted suicide should be a criminal offence
The use of public defenders should be mandatory
We should abandon television
We should prohibit flag burning
We should limit judicial activism
We should end the use of economic sanctions
We should end mandatory retirement

Train Topics

We should ban missionary work
We should ban the Church of Scientology
We should ban naturopathy
We should subsidize journalism
The vow of celibacy should be abandoned
We should adopt a zero-tolerance policy in schools
Surrogacy should be banned
We should ban telemarketing
We should legalize sex selection
We should legalize prostitution
We should ban whaling
We should ban private military companies
We should legalize organ trade
We should subsidize student loans
We should prohibit women in combat
We should end racial profiling
We should abolish intellectual property rights
We should ban factory farming
Intelligence tests bring more harm than good
We should ban the use of child actors
We should abolish safe spaces
We should subsidize Wikipedia
Foster care brings more harm than good
Social media brings more harm than good
We should abolish capital punishment
We should limit executive compensation

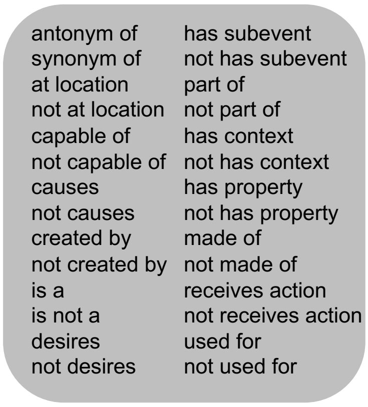
Dev Topics

We should stop the development of autonomous cars
Homeschooling should be banned
We should subsidize vocational education
We should abolish zoos
We should oppose collectivism
We should fight for the abolition of nuclear weapons
We should ban fast food
We should end affirmative action
We should legalize polygamy

Test Topics

We should abandon marriage
We should ban cosmetic surgery
We should adopt an austerity regime
We should fight urbanization
We should subsidize embryonic stem cell research
Entrapment should be legalized
We should ban human cloning
We should close Guantanamo Bay detention camp
We should adopt atheism

Figure 15: The complete list of debate topics used in our data collection process.



antym of
synonym of
at location
not at location
capable of
not capable of
causes
not causes
created by
not created by
is a
is not a
desires
not desires
has subevent
not has subevent
part of
not part of
has context
not has context
has property
not has property
made of
not made of
receives action
not receives action
used for
not used for

Figure 16: The complete list of commonsense relations used for our explanation graphs.

Task	Pay/HIT (in cents)
Pre-HAMLET Collection	25
HAMLET Collection	25
Stance Verification	5
Graph Creation	45
Graph Refinement	45
Graph Verification	10

Table 5: Payment per HIT (in cents) for each of our tasks on MTurk (with additional bonuses).

B Data Analysis

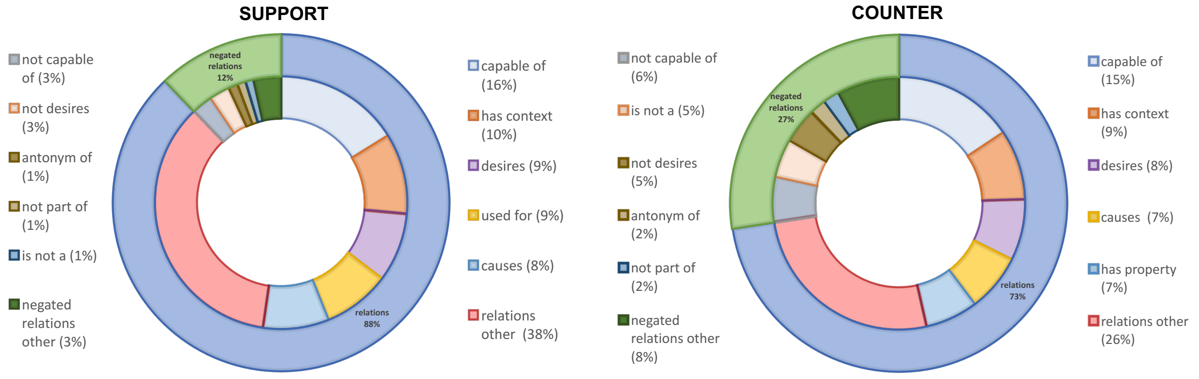
In Figure 15, we show the full list of debate topics used in our data collection process. The train split consists of 53 topics, while the dev and the test splits contain 9 topics each. Figure 16 shows all the commonsense relations used for our explanation graph creation. We broadly choose the relation set from ConceptNet (Liu and Singh, 2004), while removing generic relations like "related to" and adding a negative counterpart for every positive relation to enable the composition of supportive and counter graphs.

Due to this, the relations used to construct the facts in our graphs can be divided into two categories – with and without negations ("not capable of" vs "capable of"). We analyze the presence of these relations separately for the support and counter graphs. Fig. 17 illustrates that while non-negated relations are used more frequently in both kinds of graphs, they broadly follow a similar distribution of negated vs non-negated relations, demonstrating that the usage of a type of relation is not indicative of the stance label and actually depends on the specific context they are being used in. Interestingly, we also observe that the most frequently used relations in both stances are causal in nature (like "capable of", "causes", "desires", and their negative counterparts), which further supports our graphs as explanations.

C Models

C.1 Reasoning Model (First-Graph-Then-Stance)

Our first approach towards generating both stance and explanation graphs is through a reasoning model that first predicts the explanation graph by conditioning on the belief and the argument and then uses the generated graph, augmented with the belief and the argument, to predict the stance label.



(a) Relation Distribution for Support Graphs

(b) Relation Distribution for Counter Graphs

Figure 17: Relation percentages for Gold Graphs: Frequencies of occurrence of positive and negative relation for support and counter graph along with sub-classification into relation level statistics.

The explanation graph, in this case, provides additional commonsense knowledge and structure for the stance prediction task. For the BART (Lewis et al., 2019) or T5-based (Raffel et al., 2020) graph prediction models, the input is the concatenated belief, argument (separated by separator) and the output is the explanation graph. We represent and predict graphs as linearized strings formed by concatenating the constituent edges. Since our explanation graphs are connected DAGs, during training, the edges are concatenated according to the depth-first-search (DFS) order of the nodes. In our experiments, we perform an empirical study showing that DFS marginally outperforms other edge orderings and is significantly better than a random ordering (see Results). Next, for the stance prediction model, we fine-tune a pre-trained sequence classification model, RoBERTa (Liu et al., 2019), which conditions on the concatenated belief, argument and the linearized graph to predict the stance label.[5]

C.2 Rationalizing Model (First-Stance-Then-Graph)

Our second approach is via a rationalizing model which generates graphs as post-hoc explanations. Specifically, we first fine-tune a RoBERTa model to predict the stance label by conditioning on the belief and argument. The predicted labels are then concatenated with the belief and argument to fine-tune BART and T5 models for generating the explanation graph in a post-hoc manner. Similar to the reasoning models, graphs are represented as linearized strings according to the DFS order of the nodes.

C.3 Commonsense-Augmented Structured Prediction Model

Our model consists of the following four components.

Internal Nodes Prediction: It involves predicting the nodes which are either part of the belief or the argument. We build this module on top of RoBERTa (Liu et al., 2019), which takes in the concatenated belief and the argument (separated by a separator). The task is posed as a sequence-tagging problem, where given a sequence of tokens $s = (t_1, t_2, \dots, t_n)$, each token is classified into one of the three classes, {B-N, I-N, O}, where B-N indicates the start token of the node, I-N denotes the intermediate node tokens and O denotes tokens which are not part of any node. For example, given a belief “Factory farming should not be banned”, the gold sequence tag is {B-N, I-N, O, O, O, B-N}. Given the representation of each token from RoBERTa, we classify them into one of the three classes using two fully-connected layers with dropout. The module is trained using standard cross-entropy loss over all tokens.

External Commonsense Nodes Prediction: For generating external commonsense nodes which are neither part of the belief nor the argument, we separately fine-tune a BART model.[6] We construct samples, where the input is again the concatenated belief and the argument and the

[5] The stance prediction model can possibly be improved with better encoding of the explanation graph (e.g., through graph neural networks). We hope our challenging dataset encourages such model development as part of the future work by the community.

[6] We also experiment with T5, but find BART to perform better.

output is a comma-separated list of external nodes. For example, we construct samples like X = *Factory farming should not be banned <s> Factory farming feeds millions,* y = *Food, Necessary*, where "Food" and "Necessary" are the commonsense nodes identified from the gold graph. The generated nodes from the BART model are fed to RoBERTa (from the previous module) and concatenated with the belief and the argument as part of the input, so as to have an unified model.

Edge Prediction: We model edge prediction as a multi-way classification problem over 29 classes (one class for each of our 28 relations and one for no edge, if no edge exists between the two nodes). Given the representation of each token from RoBERTa, we construct the representation of each node by mean-pooling over the representations of the constituent tokens. These node representations are used to construct the edge representations. Specifically, given two node representations n_i and n_j and the representation of a relation r, we construct the edge representation for that relation by concatenating the relation representation, the individual node representations along with their element-wise difference to capture the directionality of the edge. Similar to the node module, the edge embeddings are also passed to a standard 2-layer classifier which predicts the probability of each edge belonging to any one of the classes. The module is trained with cross-entropy loss over all edges. Our final loss is the summation of the node loss and the edge loss. Given that our training data is not sufficient to learn commonsense relations between concepts from scratch, we initially fine-tune the RoBERTa pre-trained weights and the edge classifier on ConceptNet (Liu and Singh, 2004) triples. Specifically, we consider facts like *(man, capable of, eating)* from ConceptNet and create training data consisting of X = *man <s> eating* and y = *capable of* where *<s>* is a separator used for separating the two concepts. We find that augmenting knowledge from ConceptNet improves the edge prediction capability of our model.

ILP Inference for Graph Constraints: Our inference procedure operates in two steps. Note that our edge prediction module is conditioned on the node module which means that edges will be predicted between the chosen nodes only. Thus, predicting edges requires predicting the nodes first. Once we obtain the internal and external nodes from their respective modules, in the second step, we predict the edge probabilities using the edge module. During edge inference, we want to enforce additional constraints such that the edges are predicted in a way that the final explanation graph is a connected DAG. Following prior work (Saha et al., 2020), we achieve this through an Integer Linear Program (ILP) by maximizing a global score over the edge probabilities as described below.

Checking for graph connectivity can be reduced to solving a max-flow problem in an augmented graph. Specifically, to ensure connectivity in an explanation graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, we first define an augmented graph $\mathcal{G}_{aug} = (\mathcal{N}_{aug}, \mathcal{E}_{aug})$ with two additional nodes s_o and s_i representing a source node and a sink node respectively. We further add an edge from the source s_o to any one of the nodes n in $\mathcal{G}$ and from all nodes in $\mathcal{G}$ to the sink s_i. Now, for a graph to be connected, there should be a maximum total flow of $|\mathcal{N}|$ from s_o to s_i.

In the reduced maximum-flow formulation (Leighton and Rao, 1999) in $\mathcal{G}_{aug}$, we define a capacity variable $c_{(m,n)}$ for each edge, $m \rightarrow n$ in G_{aug}, as follows.

$$\begin{gathered} c_{(s_o,x)} = |\mathcal{N}| \text{ and } c_{(x,s_o)} = 0 \\ \forall n \in \mathcal{N}, c_{(n,s_i)} = 1 \text{ and } c_{(s_i,n)} = 0 \end{gathered}$$

Our final optimization problem is as follows. From our edge module, we obtain $e_{(m,n,r)}$, the probability that an edge $m \rightarrow n$ has the relation r. Additionally, we also obtain $e_{(m,n,-)}$, the probability that no edge exists between the nodes m and n. Given these probabilities, we define binary optimization variables $\phi_{(m,n)}$, where 1 means that an exists between the nodes m and n, while 0 means no such edge exists. Our final optimization function is:

$$\underset{\phi_{(m,n)}, f_{(m,n)}}{argmax} \sum_{m,n,m \neq n} (\phi_{(m,n)} \max_{r}(e_{(m,n,r)}) + \\ (1 - \phi_{(m,n)})(e_{(m,n,-)}))$$

subject to constraints:

$$\forall m, n \in \mathcal{N}_{aug}, 0 \leq f_{(m,n)} \leq c_{(m,n)} \quad (1)$$

$$\forall n \in \mathcal{N}_{aug}, \sum_{m:(m,n) \in \mathcal{E}_{aug}} f_{(m,n)} = \sum_{o:(n,o) \in \mathcal{E}_{aug}} f_{(n,o)} \quad (2)$$

$$f_{(s_o,x)} = |\mathcal{N}| \quad (3)$$

	SA↑	StCA↑	SeCA↑	G-BS↑	GED↓	EA↑
RE-BART	73.9	20.8	12.5	15.6	0.85	13.0
RA-BART	**86.2**	21.6	11.0	16.1	0.85	10.3
RE-T5	70.8	29.6	12.2	22.8	0.79	18.0
RA-T5	**86.2**	35.4	15.5	27.7	0.75	19.8
RE-SP	72.3	**62.3**	**18.5**	**47.0**	**0.62**	**27.1**

Table 6: Results of our Rationalizing and Reasoning models with structured, BART and T5 variants across all metrics on EXPLAGRAPHS dev set.

	SA↑	StCA↑	SeCA↑	G-BS↑	GED↓	EA↑
Random	70.3	9.3	4.0	7.1	0.93	5.6
Topological	69.6	27.6	11.0	21.2	0.81	16.1
BFS	70.3	28.9	10.2	22.4	0.80	16.1
DFS	**70.8**	**29.6**	**12.2**	**22.8**	**0.79**	**18.0**

Table 7: Effect of edge ordering on Reasoning-T5 model. Having a random ordering leads to a significant drop in performance. Between fixed orderings, DFS performs better than BFS and Topological ordering.

Equations 1 and 2 define the flow constraints which state that flow for each edge is bounded by its capacity and that the total flow at each node is conserved. Finally, Equation 3 ensures connectivity in the explanation graph, by enforcing the total flow to be $|\mathcal{N}|$. When an edge exists, we choose the relation r with the maximum probability.

D Experimental Setup

We train all our models using the Hugging Face transformers library (Wolf et al., 2019).[7] For all RoBERTa-based models (including the commonsense-augmented structured model and the stance prediction models and our model-based metrics), we use RoBERTa-large (Liu et al., 2019) with a batch size of 32, an initial learning rate of 10^{-5} with linear decay, a weight decay of 0.1 and a maximum sequence length of 128 for training up to a maximum of 10 epochs. As for BART (Lewis et al., 2019) and T5 (Raffel et al., 2020), we use their base models with a batch size of 8, an initial learning rate of $3 * 10^{-5}$ and train for a maximum of 6 epochs. The maximum input and output sequence lengths are set to 100 and 150 respectively. Graphs are generated from these models using standard beam search decoding with beam size of 4. Batch size and learning rate are manually tuned in the range $\{8, 16, 32\}$ and $\{10^{-5}, 2*10^{-5}, 3*10^{-5}\}$ respectively and the best models are chosen based on our validation set performance. The random seed is chosen as 42 in all our experiments. The total number of parameters of our structured model is similar to that of RoBERTa-large (355M). All of our models have an average runtime between 30 mins to 1 hour. The ILP inference is modeled using PuLP.[8] All experiments are performed on one V100 Volta GPU.

E Additional Experiments and Analysis

Table 6 shows the results of all models on the EXPLAGRAPHS dev set.

E.1 Effect of Edge Ordering in BART/T5

In order to evaluate the effect of a particular edge ordering on BART and T5 fine-tuning for graph generation, we compare the performance of the Reasoning-T5 model with edges ordered according to (1) a random order, (2) Topological, (3) Breadth First Search (BFS), and (4) Depth First Search (DFS). From Table 7, we observe that having a pre-defined ordering enables the model to learn the graph structure significantly better. This, however, is not surprising; due to the auto-regressive nature of these text generation models, an un-ordered edge set confuses the model and it is not able to learn the structural properties of graphs. We observe that the random model often generates cycles and hence has a significantly low percentage of structurally correct graphs. Having a fixed ordering also enables the model to learn an inductive bias towards generating graphs in a manner than can be read and reasoned through by humans. Owing to the slightly better performance of DFS, we conduct all our experiments with the same ordering.

[7] https://github.com/huggingface/transformers

[8] https://pypi.org/project/PuLP/

	SA↑	StCA↑	SeCA↑	G-BS↑	GED↓	EA↑
Low (1-3)	73.1	63.1	18.4	45.4	0.66	26.3
Medium (4-5)	74.1	64.8	19.1	50.6	0.65	29.7
High (6-8)	63.0	50.0	17.4	38.9	0.73	20.8

Table 8: Comparison of Reasoning-SP model on the subset of examples in EXPLAGRAPHS dev set with varying reasoning depths (low, medium, high). Performance on the graph-related metrics drop significantly at higher depth.

	SA↑	StCA↑	SeCA↑	G-BS↑	GED↓	EA↑
Linear	72.1	63.3	23.9	47.7	0.66	28.5
Non-linear	72.7	61.0	11.6	45.4	0.68	25.2

Table 9: Comparison of Reasoning-SP model on the subset of examples in EXPLAGRAPHS dev set with linear vs non-linear graph structures. The semantic correctness accuracy (SeCA) of graphs drops significantly for non-linear graphs due to the complex reasoning process involved in such graphs.

E.2 Analysis with Reasoning Depths

We refer to the depth of a graph as the reasoning depth involved in inferring the stance label. As part of ablation analysis, in Table 8, we analyze the performance of the Reasoning-T5 model on the subset of examples requiring varying depths of reasoning from low (depth $<= 3$) to high (depth > 5). Unsurprisingly, we find that our task of explanation graph generation becomes challenging at higher depth, as demonstrated by a drop in all graph-related metrics at depth $>= 6$. This reveals the hardness of our task and encourages future work on better model development of explanation graph generation.

E.3 Analysis with Reasoning Structures

Our next ablation analyzes the effect of linear vs non-linear reasoning structures. We call a reasoning structure linear when the explanation graph contains a single chain of nodes. A non-linear reasoning structure adds complexity to the inference process and we validate this through our results in Table 9. Similar to the previous result, we observe that our task becomes challenging with non-linear structures as demonstrated by a significant drop in semantic correctness accuracy.

E.4 Quantitative Analysis of Generated Explanation Graphs from RE-T5

In order to gain a better understanding of the explanation graphs generated by our Reasoning-T5 model, we show sample explanation graphs generated by the model in Figure 18. Unlike our RE-SP model, it typically generates linear chains with much fewer number of external commonsense nodes.

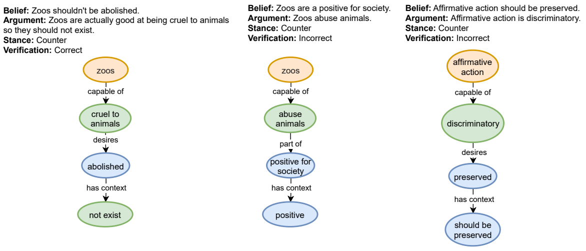


Figure 18: Examples of predicted graphs from the Reasoning-T5 model. The verification term stands for the outcome of human verification while stance refers to the gold label for the (belief, argument) pair.

F Examples from EXPLAGRAPHS

We also show some randomly chosen examples from EXPLAGRAPHS in Figures 19, 20, 21, 22, 23, 24, 25, 26, 27. Each example contains a belief, an argument, the stance and the corresponding commonsense explanation graph.

Belief: Celibacy should be respected as an expression of belief.
Argument: Vows of celibacy are often related to religious beliefs.
Stance: Support

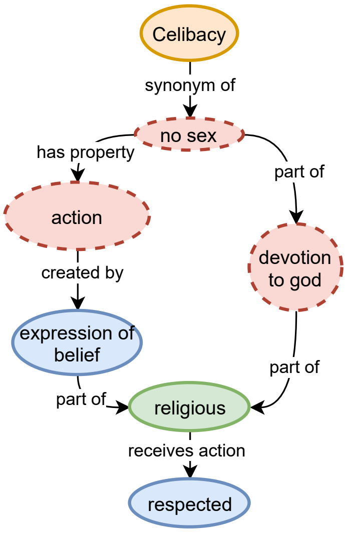


Figure 19: Example 1

Belief: Entrapment should be legal.
Argument: Entrapment catches terrible people.
Stance: Support

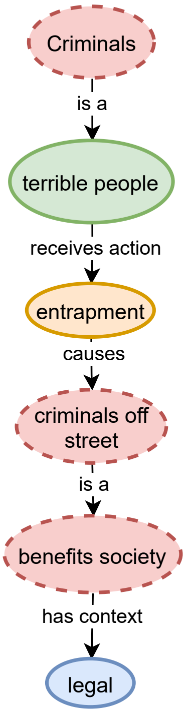


Figure 20: Example 2

Belief: Organ transplant is important.
Argument: A patient with failed kidneys might not die if he gets organ donation.
Stance: Support

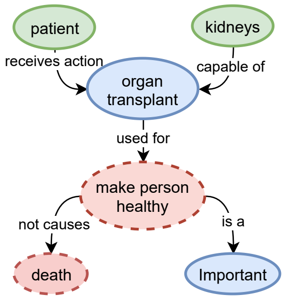


Figure 21: Example 3

Belief: Autonomous car development should end.
Argument: Autonomous cars would be better than humans.
Stance: Counter

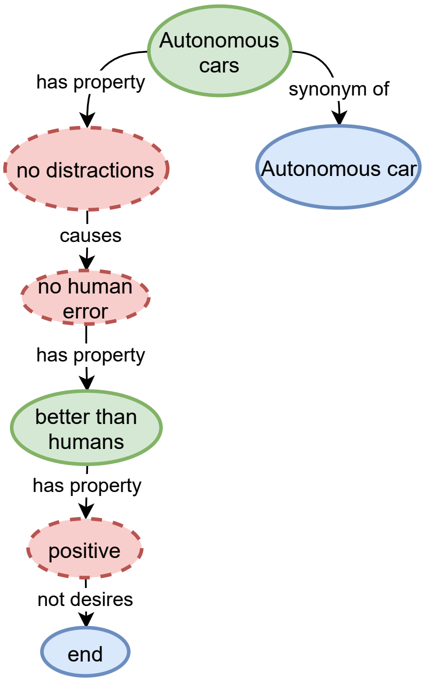


Figure 22: Example 4

Belief: Allowing organ trade does harm to the poor.
Argument: If we allow organ trade, the poor can more easily pay to acquire needed resources.
Stance: Counter

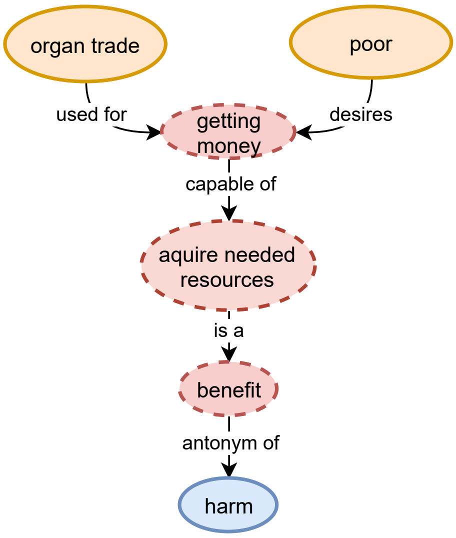


Figure 23: Example 5

Belief: Marriage is extremely important for strong families.
Argument: Marriage has been a staple in society for centuries.
Stance: Support

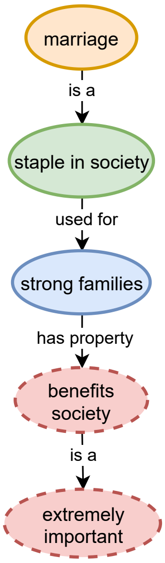


Figure 24: Example 6

Belief: Cosmetic surgery should not have an age requirement.
Argument: Young people with traumatic accidents may need reconstructive surgery just as much as an adult would.
Stance: Support

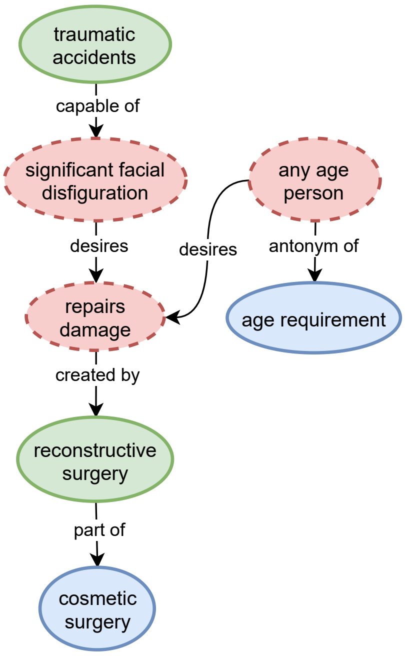


Figure 25: Example 7

Belief: Plastic surgery should not be shamed.
Argument: Plastic surgery is harmful to one's self esteem.
Stance: Counter

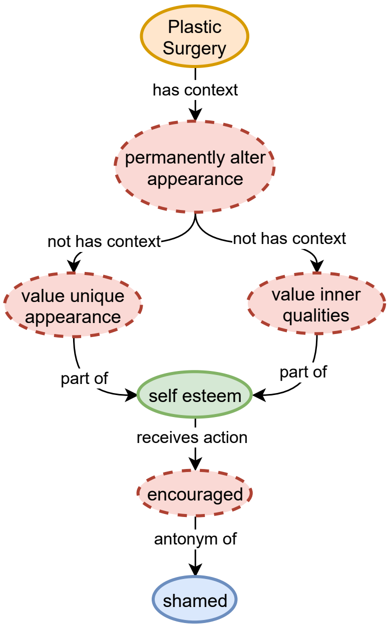


Figure 26: Example 8

Belief: Marriage does not mean much.
Argument: Marriage is the backbone of society.
Stance: Counter

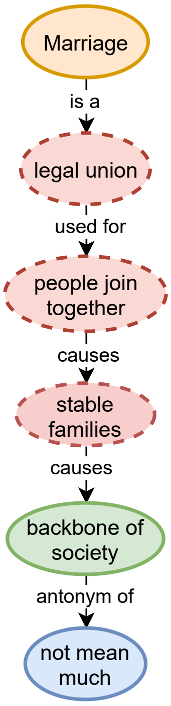


Figure 27: Example 9