"I'm Not Mad": Commonsense Implications of Negation and Contradiction

Liwei Jiang[1] **Antoine Bosselut**[2,3] **Chandra Bhagavatula**[2] **Yejin Choi**[1,2]

[1]Paul G. Allen School of Computer Science & Engineering, University of Washington
[2]Allen Institute for Artificial Intelligence
[3]Stanford University
{lwjiang,yejin}@cs.washington.edu,{chandrab}@allenai.org

Abstract

Natural language inference requires reasoning about contradictions, negations, and their commonsense implications. Given a simple premise (e.g., "I'm mad at you"), humans can reason about the varying shades of contradictory statements ranging from straightforward negations ("I'm not mad at you") to commonsense contradictions ("I'm happy"). Moreover, these negated or contradictory statements shift the commonsense implications of the original premise in nontrivial ways. For example, while "I'm mad" implies "I'm unhappy about something," negating the premise (*i.e.*, "I'm *not* mad") does not necessarily negate the corresponding commonsense implications.

In this paper, we present the first comprehensive study focusing on commonsense implications of negated statements and contradictions. We introduce ANION[1], a new commonsense knowledge graph with 624K *if-then* rules focusing on negated and contradictory events. We then present joint generative and discriminative inference models for this new resource, providing novel empirical insights on how logical negations and commonsense contradictions reshape the commonsense implications of their original premises.

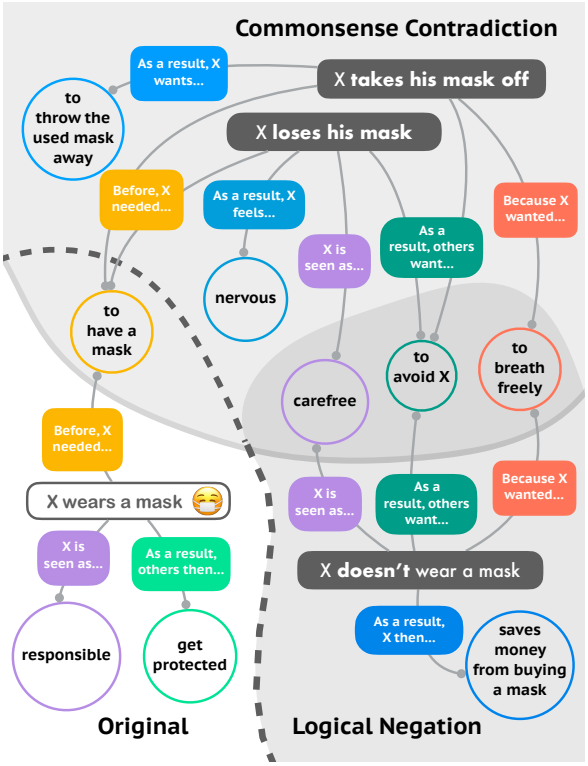

Figure 1: Commonsense inferences for the event "X wears a mask," its logical negation and commonsense contradiction events, and their associated inferences.

1 Introduction

Humans reason about underlying causes and effects of events described in text. For example, in Figure 1, the event "X wears a mask" is associated with many causal inferences such as "X is seen as responsible," or "Others get protected." Hypothesizing and reasoning about commonsense inferences is used for understanding complex situations encountered in everyday life (Sap et al., 2019; Bisk et al., 2020; Bhagavatula et al., 2020; Sakaguchi et al., 2020). This ability eludes AI systems, and has motivated the design of a wealth of commonsense knowledge resources, such as Cyc (Lenat, 1995), ConceptNet (Speer et al., 2017), and ATOMIC (Sap et al., 2020; Hwang et al., 2020), to provide structured reasoning capabilities to AI systems (Lin et al., 2019; Feng et al., 2020).

However, reasoning about negated observations remains a challenge (Hossain et al., 2020). While negation is often considered a poorer form of meaning than affirmation[2] (Ackrill, 1963; Horn and Wansing, 2020), negated statements can still imply expressive commonsense inferences. In Figure 1, the negated event "X doesn't wear a mask,"

[1]Data and code available at https://github.com/liweijiang/anion

[2]Following Horn and Wansing (2020), we classify declarative expressions as affirmations or negations/contradictions based on whether they affirm or deny an action or object.

is connected to rich commonsense inferences, despite describing the *absence* of action. However, negated observations are rarely found in commonsense knowledge resources. For example, negated examples make up only ∼3% of examples in the ConceptNet knowledge graph (Li et al., 2016).

This scarcity poses downstream issues for systems that must understand negated situations. Commonsense knowledge models (Bosselut et al., 2019; Hwang et al., 2020) trained on resources of largely affirmative instances struggle particularly with negation examples. Their ability to hypothesize inferences for negated events is 35% lower than for affirmative events (§4.2). Furthermore, since negated statements are asymmetrically mentioned in text compared to affirmative statements (Jowett et al., 1892; Horn and Wansing, 2020), large-scale pretraining does not implicitly learn negation scoping (Kim et al., 2019). As a result, when presented with negated concepts, pretrained neural language models (PLMs) often exhibit the same associations as affirmative statements (Kassner et al., 2020). Motivated by these observations, our work focuses on improving the ability of knowledge models to make commonsense inferences about events that convey denial, rejection or contradiction of actions.

We define our contributions as follows. First, we crowdsource a new large scale resource, **A**rray of commonse**N**se **I**nferences for **O**ppositions and **N**egations (ANION), which contains inferences for different types of negated events.This new resource can be used to train knowledge models on commonsense inferences associated with the absence of actions. Second, we propose a new class of negation discriminators that can be applied to generated commonsense inferences. These discriminators partition inferences based on logical consistency, thereby mitigating the effects of common affirmative associations that violate negation constraints. Discriminators are trained using contrastive samples from paired affirmative and negated events in ANION. Finally, we conduct an empirical study of both of these techniques and show that using training- and discriminator-based approaches for modeling negation cuts the performance difference between affirmative and negated events by 73% - 85% depending on the negation variety.

2 Commonsense Negation

Negation in Language In *Categories* and *De Interpretatione*, Aristotle classifies declarative statements into affirmation and negation, which respectively affirms or denies observations about an event (Ackrill, 1963). Despite this seeming simplicity, natural language often expresses negation in complex and subtle ways, using diverse syntactic, semantic and pragmatic formulations (Horn and Wansing, 2020). For example, syntactically, different negation determiners (*i.e.*, negation cues) such as *no*, *few* and *only* result in distinct explicit and implicit negative perceptions (Xiang et al., 2016).

Despite their diversity, however, negated language expressions are much less likely to appear in text than affirmative statements (Reitan et al., 2015). Consequently, PLMs, which rely on large-scale textual corpora as training data, are prone to decreased performance when confronted with negated constructions. In machine translation, for example, the presence of negation may heavily affect the quality of produced translations (Fancellu and Webber, 2015; Hossain et al., 2020). In factual knowledge understanding tasks, PLMs memorize positive and negative sentences seen during training, but generalize more poorly to unseen negated instances (Kassner and Schütze, 2020).

Negation in Commonsense Reasoning Understanding negation and oppositional expressions is critical for reasoning about commonsense knowledge, particularly in counterfactual scenarios (Qin et al., 2019). However, negation is rarely explicitly modeled in NLP studies on commonsense reasoning. As a result, in many NLP tasks, these models experience a performance drop when presented with examples exhibiting negated characteristics.

As a case study, the ATOMIC (Sap et al., 2020) knowledge graph encodes social commonsense knowledge about event pre-conditions, event post-conditions, and static attributes in the form of natural language *if-then* rules. However, despite the fact that ATOMIC provides a rich set of seed events, it comprises an unbalanced set of affirmative events (97.9%) and negated events (2.1%). As a result, when systems link to ATOMIC to retrieve relevant social commonsense inferences, they are likely to recover inferences of affirmative events even when searching for negated instances. Furthermore, knowledge models that use this resource (*e.g.*, COMET; Bosselut et al., 2019) are unlikely to learn implicit differences between inferences of affirmative and negated events. When given negated events, these models often produce associations of counterpart affirmative events. For example, for

Types	Example Negation Cues	Example Sentences
Affixes	un-, ir-, non-, il-, im-, -less, etc.	X addresses an *irr*elevant point X is *un*likely to be a spy X *un*saddles the horse
Single-word	not, no, nothing, nobody, few, little, without, never, hardly, rarely, barely, seldomly, etc.	X does *not* tell the truth to the public X *never* eats ice cream X went to a movie *without* his friends
Multi-word	no longer, barely/hardly ever, not at all, a lack of, be deprived of, in the absence of, on no condition, by no means, not by any means, under no circumstances, make no attempt to, etc.	X *no longer* wants to buy a car X is *not at all* impressed by Y's ideas X *under no circumstances* smokes X is *by no means* cheating on Y
Negative Verbs	oppose, refuse, resist, avoid, disapprove, lack, discontinue, stop, cease, halt, prohibit, forbid, prevent, reject, fail, etc.	X *denies* the existence of god X *restrains* himself from eating with Y X *refuses* to be in a relationship

Table 1: Negation cues and examples from ANION.

the negated event, "X opposes racism," COMET infers "X intends to be a racist," an association of the affirmative statement, "X supports racism."

At the heart of this problem is that inferring commonsense knowledge about negations often requires implicit reasoning. In factual knowledge reasoning, applying logical rules over statements can be effective for handling negative queries (Asai and Hajishirzi, 2020; Ren and Leskovec, 2020). However, directly manipulating affirmative forms with logic-guided rules may fail for commonsense reasoning: the boundary of commonsense inferences between affirmative and negated statements is not always wholly contrastive. Many inferences can be relevant to both forms. The events "X puts the potato in the oven" and "X doesn't put the potato in the oven," could both have an associated inference: "X wants to make dinner." The affirmative event clearly implies this inference. For the negated event to be worth mentioning on its own (Grice et al., 1975), an implicit complementary event (*e.g.*, "X puts the potato in the microwave") would likely hold, which might validate the inference w.r.t. the negated event. To model the defeasibility of commonsense reasoning (Pratt, 1994; Rudinger et al., 2020), modeling both common and contrastive inferences of negated forms is necessary.

3 ANION: Commonsense Inferences of Oppositions and Negations

To provide a rich resource of commonsense inferences for opposition and negation events, we design ANION. Using the same schema as the ATOMIC knowledge graph (Sap et al., 2020), we initialize 22,483 negated forms paired to original ATOMIC events and crowdsource 627,042 new inferences for these negated events. Consistent with ATOMIC, ANION is constructed using English formulations of events and inferences. We briefly recap ATOMIC and describe the construction of ANION below.

ATOMIC Background The ATOMIC knowledge graph contains ∼24K base events (*e.g.*, "X plays the piano") with 877K accompanying social commonsense inferences (*e.g.*, "Before, X needs to buy a piano.") along nine dimensions (*e.g.*, *xNeed*). The full description of ATOMIC relation types can be found in Table 12 in the Appendix.

3.1 Overview of ANION Construction

Our knowledge construction pipeline consists of two steps. First, we collect negated and contradictive events by deriving oppositions of events in ATOMIC. Inspired by the distinction made between negation contributed by semantic assertion (explicit negation) or non-asserted content (implicit negation) (Xiang et al., 2016), we define three varieties of negated events: logical negations, semi-logical negations, and commonsense contradictions, which we describe in detail below. Logical and semi-logical negations were heuristically formulated from ATOMIC events. Commonsense contradiction events were crowdsourced from Amazon Mechanical Turk (MTurk). Negated events in ANION are assigned to the same data split as the corresponding affirmative event from which they are derived (*e.g.*, negated events for ATOMIC training set events are found in the ANION training set).

Once a list of negated events is compiled, we crowdsource inferences of these new events on MTurk using similar annotation templates as Sap et al. (2020). We design qualifying tasks to filter out unreliable workers and screen their answers manually for quality control purposes.

Type		#Words	Total	Train	Development	Test
ATOMIC	event	4.61	25,096	20,322	2,282	2,492
	inference	-	795,059	643,571	72,227	79,261
ANION - Logical (L)	event	4.47	8,285	4,175	1,903	2,207
	inference	-	225,635	110,864	57,170	57,601
ANION - Semi-logical (S)	event	4.52	5,019	2,457	1,223	1,339
	inference	-	138,587	66,087	33,030	39,470
ANION - Commonsense Contradiction (C)	event	4.46	9,179	3,267	2,808	3,104
	inference	-	262,820	93,419	95,685	73,716

Table 2: Statistics of ATOMIC and different subsets of ANION (ANION-L + ANION-S + ANION-C).

Logical Negation We define logical negation events as events with the negation cue *not* added to their original formulation (*e.g.*, "X does not play the piano"). However, different positions of the *not* modifier in a clause can result in different *negation scopes*, which can alter the semantics of the event (Councill et al., 2010). To be consistent, we systematically insert *not* after the subject of the event clause. If necessary, we change verb forms and add auxiliary words (*e.g.*, do, does, did, is, was, can, could, would, should, may, might). For quality control, we have human workers validate each logically negated event form and exclude events that annotators identify as uninterpretable or awkwardly worded. For each created event, we then collect the same nine dimensions of inferences as defined in ATOMIC. Consequently, we collected 8,285 logically negated events with 225K corresponding inferences (as shown in Table 2). Appendix A.1 provides further details of the compilation of logical negation events.

Semi-logical Negation We define semi-logical negation using explicit cues other than *not*. We categorize these negation cues (words or phrases) into four subtypes: affixes (*e.g.*, legal/illegal), single-word cues (*e.g.*, never), multi-word cues (*e.g.*, no longer), and negative verbs (*e.g.*, refuse). See Table 1 for examples. We create semi-logical negation events by heuristically adding these cues to different positions of ATOMIC events. Similar to logically-negated events, we avoid grammatically incorrect or semantically awkward events by removing auto-generated instances of low quality. The final set of data includes 5,019 semi-logical negation events. We then crowdsource a total of 138K inferences for these new events. Appendix A.1 provides further details of the compilation of semi-logical negation events.

Event	Commonsense Contradiction
X buys a bicycle	X buys a car X donates a bicycle
X walks in the door	X stops at the door X walks out of the building
X works hard all day	X plays games all day X puts in minimal effort all day
X finishes the story	X starts the story X stops halfway through the story
X turns the air blue	X secretly curses X speaks appropriately

Table 3: Contradictions of events from ATOMIC

Commonsense Contradiction We formulate commonsense contradiction as contradictory statements without negation cues. Commonsense contradiction events are not identifiable as negations on their own, but demonstrate reversed semantic or pragmatic meaning when paired with their affirmative counterparts (*e.g.*, "X eats a hamburger" vs. "X eats a salad"). To obtain commonsense contradictions, we crowdsource two oppositional events for each ATOMIC event, excluding events with blank placeholders representing generic objects, resulting in 40K new commonsense contradiction events. For 9,179 of these events, we crowdsource an additional 262K commonsense inferences. Appendix A.1 provides further details of the crowdsourcing of commonsense contradiction events.

4 Knowledge Models of Negated Events

ANION can be used as training data for commonsense models to make inferences about negated events. Here, we recap COMET (Bosselut et al., 2019), a commonsense knowledge model, and evaluate how training knowledge models on ANION affects their ability to hypothesize commonsense knowledge for negated and oppositional events.

Eval Set	Train Set	PPL ↓	BL2 ↑	P@10 ↑
ATOMIC	ATOMIC	9.30	**14.18**	**55.18**
	ATOMIC + ANION	**9.28**	14.05	*53.61
ANION-L	ATOMIC	10.87	10.86	35.84
	ATOMIC + ANION	**9.08**	**11.96**	**45.42
ANION-S	ATOMIC	11.69	12.07	36.89
	ATOMIC + ANION	**9.80**	**13.22**	**46.88
ANION-C	ATOMIC	12.02	14.32	46.70
	ATOMIC + ANION	**11.20**	**14.64**	**50.65

Table 4: Evaluations of COMET models trained on ATOMIC and ANION KGs. Training on examples of negated events leads to large improvements in the quality of generated inferences with minimal dropoff in the quality of inferences for affirmative events. Single (*) and double asterisks (**) indicate significance at p<0.05 and p<0.01, respectively.

4.1 Setup

Commonsense transformers (COMET) are generative knowledge models that learn to hypothesize commonsense inferences by training on examples from a knowledge graph. Specifically, COMET receives knowledge tuples in $\{h, r, t\}$ form during training, where h is a head entity, r is a relation type, and t is a tail entity. The model is trained to maximize the conditional loglikelihood of predicting the tokens of the tail entity t given the tokens of the head entity h and relation r:

$$\mathcal{L}_G = -\sum \log P(t|h, r) \quad (1)$$

In ATOMIC and ANION, h corresponds to events, such as "X has a nightmare," t corresponds to commonsense inferences about those events, such as "X wakes up," and r corresponds to commonsense inference types, such as "As a result, X does...".

Following Bosselut et al. (2019) and Sap et al. (2020), for each event and relation type in ATOMIC, 10 candidate inferences are decoded from COMET using beam search with b=10.

4.2 Experiments

As oppositional instances remain challenging to knowledge models such as COMET, we evaluate how ANION can be used to augment the type of examples seen by COMET during training.

Evaluation Metrics Following Bosselut et al. (2019), we evaluate the quality of generated inferences using BLEU-2 (Papineni et al., 2002) as an automatic evaluation. We also compute the perplexity of models on their reference generations.

For the human evaluation, we employ human judges from MTurk to identify whether generated commonsense inferences are plausible. We randomly sample 100 events from the original ATOMIC test set along with their negated counterparts from ANION. For each event, we present every decoded inference to five crowdworkers and ask them to identify whether the inference is plausible given the event. For each model trained on a different combination of ATOMIC and ANION (*i.e.*, ANION-L, ANION-S, ANION-C), we evaluate the same events for comparison. We calculate Precision @ 10 (P@10) across these human ratings, *i.e.*, the average number of correct options per event-relation prompt. Specifically, we average the results from 45K ratings to compute the final human score (100 events $\times$ 9 relations $\times$ 10 options $\times$ 5 annotators). The pairwise agreement score of human evaluation is 63.6, which is on par with other similar commonsense reasoning annotation tasks (Rashkin et al., 2016).

Does negated event training improve commonsense inference for negated situations? We train a COMET model on the events from ATOMIC (*i.e.*, COMET-ATOMIC), and another on the examples from both ATOMIC and ANION (*i.e.*, COMET-FULL). The combined dataset is shuffled so that the original and negated examples are uniformly mixed during training.

We report our comparison of these two models in Table 4. The performance of the original COMET model trained only on the ATOMIC knowledge graph drops significantly across all types of oppositional instances. Most surprisingly, a drop in performance is also observed on commonsense contradictions (ANION-C), which have no explicit negation cues. However, commonsense contradiction events can often be richer in content (see Table 3), making them more challenging for knowledge models. Meanwhile training on all negated examples in the ANION knowledge graph produces significant improvements across all negation categories (ANION-{L,S,C}), though we do observe a slight drop in human ratings on the examples from the original ATOMIC test set.

Does negated event training deteriorate commonsense inference of affirmative situations? We note in Table 4 that training on ATOMIC + ANION hurts inference performance on the original ATOMIC evaluation set. To analyze why COMET-

FULL does not improve on this set of examples, we perform a case study on inferences generated by COMET-ATOMIC and COMET-FULL under the same event and relation prompt, and note two qualitative patterns.

First, we observe that COMET-FULL tends to generate inferences that are less generic, but that may require additional implicit context. For example, for the event "X is really sad" and the relation *xEffect* (*i.e.*, the effect of the event on X), COMET-ATOMIC generates inferences such as "cries," "gets depressed" and "takes medication." Conversely, COMET-FULL generates context-specific inferences such as "thinks about the past" and "thinks about what they did," which, while plausible in some context, may be less straightforward when evaluated broadly (not all feelings of sadness lead to reflection on the past or one's own actions).

Second, we find an overall improvement for certain compositional events in ATOMIC that contain conjunction words: "and" or "but." On these examples, COMET-FULL outperforms COMET-ATOMIC with 12.41 and 12.22 BLEU-2 scores respectively. For example, for the event "X is hot and humid" and the relation *xEffect*, COMET-ATOMIC's generation includes correct inferences, such as "to take a shower," "to cool down," "to drink some water," "to go outside," and incorrect inferences, such as "to turn on the heat" and "to drink a hot tea." COMET-FULL generates all of COMET-ATOMIC's correct inferences, but none of the incorrect inferences, demonstrating that training COMET jointly on ATOMIC and ANION can help avoid incorrect inferences involving commonsense mismatch in more compositional situations.

In summary, the ability to generate richer, contextual inferences for COMET-FULL is beneficial when handling complex events, but may not be necessary for many of the simple events in ATOMIC, and may backfire when subtler inferences are made.

Which variety of negated events are most crucial to include in training sets? As ablations, we train additional models using different subsets of ANION: logical negations (ATOMIC + ANION-L), semi-logical negations (ATOMIC + ANION-S), and commonsense contradictions (ATOMIC + ANION-C). These ablations evaluate whether knowledge models can adapt to certain types of negation more efficiently with additional data.

In Table 5, we show that training with examples of each negation type improves performance on the evaluation set related to that negation type. Interestingly, though, training on certain types of negation examples can also yield benefits downstream on other negation types. For example, training on commonsense contradictions (ANION-C) provides a clear benefit when evaluating on semi-logically negated events (ANION-S) as opposed to merely training on ATOMIC. Notably, the knowledge model trained with logically negated examples (ATOMIC + ANION-L) outperforms the model trained only on ATOMIC on all test sets.

Eval Set	Train Set	PPL ↓	BL2 ↑	P@10 ↑
ATOMIC	ATOMIC	9.30	14.18	55.18
	+ ANION-L	**9.27**	**14.20**	****58.11**
	+ ANION-S	9.30	14.09	55.74
	+ ANION-C	9.29	14.10	**52.22
ANION-L	ATOMIC	10.87	10.86	35.84
	+ ANION-L	**9.28**	**11.94**	****44.94**
	+ ANION-S	9.93	11.29	**44.01
	+ ANION-C	10.34	11.04	**42.33
ANION-S	ATOMIC	11.69	12.07	36.89
	+ ANION-L	10.69	12.69	**42.38
	+ ANION-S	**10.23**	**12.79**	****45.50**
	+ ANION-C	10.95	12.35	**41.76
ANION-C	ATOMIC	12.02	14.32	46.70
	+ ANION-L	11.72	14.43	47.78
	+ ANION-S	11.67	14.34	46.09
	+ ANION-C	**11.50**	**14.58**	****48.79**

Table 5: Ablation results of models trained and evaluated on different portions of ANION. The best result on each subset of ANION comes from training on similar examples. The model trained on negated events from ANION-L performs the best at generating inferences for the original ATOMIC events. Double asterisks (**) indicate significance at p<0.01.

5 Discriminating Inconsistent Inferences

While training on examples of negated events helps knowledge models generate commonsense inferences for these event types, there is still a large gap compared to their performance on affirmative events. To address this discrepancy, we introduce a discriminator-based approach for distinguishing inconsistent inferences of negated events. Our inference discriminator learns to identify plausible and invalid inferences of events by learning from contrastive samples from ATOMIC and ANION.

5.1 Experimental Setup

We fine-tune the RoBERTa-base model (Liu et al., 2019) as a binary classifier to identify whether a given knowledge tuple $\{h, r, t\}$ is logically valid. The model is trained on paired original and negated

events as described below. Such training examples inject implicit commonsense nuances that differ between oppositional events to teach the discriminator to identify logical pitfalls. Training details for discriminators can be found in Appendix A.3.

Data The paired events used to train the negation discriminator are automatically constructed from the ATOMIC and ANION knowledge graphs. Positive examples can be constructed by sampling tuples from each knowledge graph. To construct negative training samples, we introduce the concept of *common* and *contrast* sets among inferences of events and their oppositions.

Common and contrast sets distinguish how commonsense inferences are not necessarily negated in the same manner as their corresponding events. While certain inferences of events are also in opposition to a negated event, some may be common. For the events "X eats a cheeseburger" and "X eats a salad," an inference such as "X is hungry" might be common to both events while inferences such as "X is unhealthy" or "X is healthy" would be viewed as contrastive.

Specifically, we assume two head events in ATOMIC and ANION, and their respective set of tail inferences regarding a common relation type. We define the common set of these inferences as the intersection of the two sets of tail inferences connected to each head event by applying the exact match of string forms. The contrast set is formed by distinct tail inferences connected to the two head events. Logically valid (*i.e.*, positive) training examples consist of knowledge tuples from ATOMIC and ANION. Logically invalid (*i.e.*, negative) training examples are formed by swapping the set of contrast set inferences between paired original and negated events.[3]

To balance the training set, we sample the same number of positive and negative tuples for original and negation events. Statistics of the resulting training sets are in Table 6.

5.2 Experiments

Using different portions of ANION for training yields four unique discriminators (*i.e.*, **L**, **S**, **C** and **LSC**) that we apply to commonsense inferences generated by COMET. The discriminators classify each option as either logically *valid* or *invalid*, partitioning the candidates into two sets, which we evaluate with human judgements. As a baseline, we also record the precision of not using a discriminator, which assumes all generated inferences are valid candidates (*i.e.*, the *all* set).

[3]We note that annotations in ATOMIC and ANION are finite (*i.e.*, not covering the full space of possible commonsense inferences about events). As a result, it is possible that in a more expansive annotation, elements of the contrast sets would in fact be part of the common set of an event and its negation. For the purpose of this work, however, contrast sets were an efficient way of acquiring high-quality semantically negative examples for training discriminators.

Discriminator	Train Set	Size
Logical Negation (**L**)	ANION-L	324,843
Semi-logical Negation (**S**)	ANION-S	194,732
Commonsense Contradiction (**C**)	ANION-C	276,272
All Oppositional Data (**LSC**)	ANION	795,845

Table 6: Statistics of data used to train negation discriminators.

Eval Set		#	BL2↑	P@*k*↑
ATOMIC	all	10.0	14.18	55.18
	valid	6.3	**14.24**	**59.07**
	invalid	3.7	13.93	44.10
ANION-L	all	10.0	10.86	35.84
	valid	5.6	**11.33**	**45.59**
	invalid	4.4	10.13	25.96
ANION-S	all	10.0	12.07	36.89
	valid	6.3	**12.63**	**44.93**
	invalid	3.7	11.32	27.83
ANION-C	all	10.0	14.32	46.70
	valid	5.9	**14.78**	**51.45**
	invalid	4.1	13.56	37.33

Table 7: The evaluation of the *all*, *valid* and *invalid* sets of inferences generated by COMET-ATOMIC as partitioned by the **LSC** discriminator. **P@*k*** corresponds to the human-rated precision of a set. k is the number of elements in *all, valid,* or *invalid* set. For the *valid* set, higher **P@*k*** is better (*i.e.*, more valid inferences are being partitioned). For the *invalid* set, lower **P@*k*** is better (*i.e.*, fewer valid inferences are being included).

Metrics We evaluate and compare the quality of the *all*, *valid* and *invalid* sets using BLEU-2 and the same human evaluation as in §4. The *all* set contains the full set of 10 candidates, while the *valid* and *invalid* sets have varying number of elements depending on how discriminators classify them, summing to 10. To compute statistical significance between valid and all sets, we use a permutation test with 100K permutations. Details are provided in Appendix A.4.

Do discriminators effectively distinguish inconsistent inferences? The results in Table 7 demonstrate that the discriminator trained on all subsets of ANION (**LSC**) can select subsets of inferences (*i.e.*,

Event + Rel	Generation	V	P
X does not skate around ***xAttr***	athletic	✗	✗
	careless	✗	✗
	lazy	✓	✓
	uncoordinated	✓	✓
	unskilled	✓	✓
X does not sit behind Y ***xIntent***	to be alone	✓	✓
	to be left alone	✓	✓
	to avoid Y	✓	✓
	to sit	✗	✗
	to wait	✓	✗
X does not look angry ***xNeed***	to calm down	✗	✓
	to watch a movie	✓	✗
	to have been provoked	✗	✗
	to not be angry	✓	✓
	to be calm	✓	✓
X refuses to hear a scary noise ***xWant***	to run away	✗	✗
	to go to sleep	✓	✓
	to be safe	✓	✓
	to keep quiet	✓	✓
	to avoid the noise	✓	✓
X never brings Y into conflicts ***oWant***	to avoid X	✗	✗
	to be left alone	✗	✓
	to thank X	✓	✓
	to fight back	✗	✗
	to avoid conflict	✗	✓
X scarcely gets sunburned ***xReact***	burned	✗	✗
	hurt	✗	✗
	sick	✗	✗
	sad	✗	✗
	satisfied	✓	✓
X under no circumstances forgets Y's wallet ***oReact***	upset	✗	✗
	sad	✗	✗
	angry	✗	✗
	thankful	✓	✓
	grateful	✓	✓
X has trouble with advertising X's business ***xEffect***	loses money	✓	✓
	loses clients	✓	✓
	gets fired	✓	✓
	gets sued	✗	✗
	cries	✓	✓
X puts Y out of mind ***oEffect***	has a better day	✗	✗
	becomes sad	✓	✓
	cries	✓	✓
	is grateful towards X	✗	✗
	feels better	✗	✗

Table 8: Inferences of randomly selected ANION events by COMET-ATOMIC. The top 5 options are classified as *valid* or *invalid* by the **LSC** discriminator. **V** indicates whether an option is classified as *valid* by the **LSC** discriminator. **P** indicates whether an option is plausible judging by humans.

the *valid* set) that are more logically consistent with their seed event. This observation holds across all evaluation subsets of ANION, as well as the original ATOMIC evaluation set. Table 8 shows examples of *valid* and *invalid* candidates for negated and contradicted events from ANION as specified by the

Eval \ Disc		L	S	C	LSC
ATOMIC	all	**55.69**	55.93	56.94	58.30
	valid	55.65	**56.18**	**57.26**	**59.07**
	%iprv	-0.07	0.44	0.57	1.32
ANION-L	all	39.46	37.85	36.43	39.45
	valid	****46.39**	****41.93**	**37.54**	****45.59**
	%iprv	17.55	10.78	3.03	15.57
ANION-S	all	37.13	39.29	37.72	38.55
	valid	**37.48**	****44.58**	**39.03**	****44.93**
	%iprv	0.96	13.47	3.45	16.56
ANION-C	all	**46.92**	47.32	48.26	48.81
	valid	46.83	**47.68**	**48.79**	***51.45**
	%iprv	-0.20	0.75	1.09	5.40

Table 9: P@{# valid} scores of the *all* and *valid* sets determined by the **L**, **S**, **C** and **LSC** discriminators. Generations are from COMET-ATOMIC. Asterisks (**) indicate significance at p<0.01. iprv% is the improvement of the *valid* over the *all* set. Underlines show the highest iprv% across discriminators.

LSC discriminator. The discriminator is notably good at identifying invalid inferences wrongly associated to corresponding affirmative events (*e.g.*, "athletic" and "careless" for the event "X does not skate around" under the relation, *xAttr*).

However, this analysis leaves open the possibility that we are generating too many inferences for each event, but that the decoder could rank correct inferences higher among the full set of generated candidates. To evaluate this possibility, we count the number of elements in the *valid* sets for each example and only keep the same number of the top-scoring elements from the *all* set (scored using generation perplexity). In Table 9, we see the average precision score for the pruned *all* sets (P@{# valid}) still underperforms the precision of their corresponding *valid* sets.

Which negation categories are most important to provide a discriminator for? To examine the generalization effects of each negation type, we also train discriminators on a single negation subset of ANION examples (*i.e.*, **L**, **S**, **C**) and compare the P@{# valid} score of the *all* and *valid* sets. Results in Table 9 indicate that each discriminator is best for identifying valid inferences for the types of events on which it was trained. The **L**, **S**, and **C** discriminators all achieve improvements when partitioning events similar to their training. However, the **LSC** discriminator trained on all negation forms shows the largest *valid* set improvement across all discriminators on ATOMIC, ANION-S,

Eval \ Disc		L	S	C	LSC
ATOMIC	all	54.16	54.49	55.03	55.68
	valid	**54.20**	**54.64**	**55.71**	****57.58**
	%iprv	0.08	0.28	1.23	3.41
ANION-L	all	46.54	46.26	46.15	46.39
	valid	****50.71**	****48.36**	**46.16**	****49.85**
	%iprv	8.98	4.54	0.03	7.45
ANION-S	all	46.90	47.73	47.47	47.53
	valid	**47.14**	****50.42**	**48.20**	****50.62**
	%iprv	0.51	5.65	1.55	6.50
ANION-C	all	50.80	51.29	51.28	51.83
	valid	**50.94**	**51.52**	**52.65**	***53.91**
	%iprv	0.28	0.45	2.67	4.02

Table 10: P@{# valid} scores of the *all* and *valid* sets determined by the **L**, **S**, **C** and **LSC** discriminators. Generations are from COMET-FULL . Single (*) and double asterisks (**) indicate significance at p<0.05 and p<0.01, respectively. iprv% is the improvement of the *valid* over the *all* set. Underlines indicate the highest iprv% across discriminators.

Beam Size	Set	#✓	#total	P@#total
10	all	3.6	10.0	35.84
	valid	2.1	4.4	45.59
25	all	8.1	25.0	32.29
	valid	**4.3**	10.5	**38.18**

Table 11: Number of correct generations from applying the **LSC** discriminator to generations of COMET-ATOMIC for beam size of 10 and 25 for logical negation events. #✓ is the number of correct options. *#total* is the number of options in each set.

and ANION-C. On ANION-L, the **LSC** discriminator still yields a significantly improved *valid* set.

6 Discussion

Are learning-based and discriminator-based approaches complementary? We apply our discriminators to the generations of the COMET model trained on ANION. In Table 10, we see that the **LSC** discriminator, when applied to generations of COMET trained on ANION, achieves significant improvements over all evaluation sets, including the original events. The full evaluation of the P@{# valid} and P@3 scores of applying different discriminators to generations of COMET trained on different data over all evaluation sets are shown in Table 13 and 14 in Appendix A.

Can discriminators be used to more aggressively generate inferences? While applying discriminators to generated inferences yields a *valid* subset with higher accuracy, we are left with fewer correct inferences in total. Thus, we investigate the efficiency of using discriminators to expand the number of inferences generated. We decode inferences from COMET with beam size 25, and then apply the discriminator to this larger candidate set.

Table 11 shows that for logical negation, the *valid* set of beam 25 has higher accuracy and more correct options than the *all* set of beam 10. Thus, when we have a larger and potentially more noisy set of candidates, applying the negation discriminator yields a set of options that have higher quality than using all the candidates from a smaller set of initial generations.

7 Conclusion

We present the first comprehensive study on commonsense implications of negations and contradictions. To expand commonsense resources for the challenge of negation modeling, we introduce ANION, a large scale commonsense knowledge graph for negated and contradicted events. We use ANION to train commonsense knowledge models and demonstrate that it effectively enriches machine commonsense inference capabilities around negation. Lastly, we propose a negation discriminator capable of identifying logical flaws in commonsense inferences. By combining the model trained on ANION with the negation discriminator, we achieve a further performance boost.

Ethical Considerations

ANION Language Choice and Implications

We select English as the base language of ANION so that our resource may be directly linked with the original ATOMIC knowledge graph. We acknowledge, however, that resources in English are more likely to reflect the mindsets and behaviors of English speakers. Furthermore, and in our case specifically, our annotators were primarily from the US. Consequently, this language choice biases the content of the knowledge graph toward North American perspectives, which affects what models trained on these resources would learn about social norms (Acharya et al., 2021). Future works may also include other languages and cultures to make the ANION resource more culturally and ideologically inclusive.

Crowdworker Recruitment, Quality Control and Remuneration

We recruit crowdworkers from MTurk who are located within the US with HIT approval rates higher than 98%. To ensure high quality task completions, we post pilot batches and manually examine tens of thousands of responses to identify users who provide high quality annotations. We select 834 qualified users for the formal data collection and human evaluation tasks. Since the entire study spans multiple months, we regularly sample responses to re-examine their quality during the formal study, and remove HITs from crowdworkers who provide decreased-quality responses over time. We are particularly cautious about the human evaluation tasks, so even with qualified users, we still comprehensively examine tens of thousands of human evaluation tasks by grouping HITs per users, and look at their responses together to identify potential spamming behaviors and inconsistencies.

For the data collection and human evaluation tasks, we aimed to compensate crowdworkers with an average of $15 per hour. To ensure a fair payment, we first post a pilot task to evaluate average time cost of a specific task, and pay users at a high rate in this round to avoid underpayment during the pilot study. We then calculate new payment from the pilot task such that approximately 75% of the HITs would have been paid with more than $15 per hour at the adjusted rate in the pilot round. We then adopt this new rate for the formal study. We repeat the above procedure of determining payment periodically during the study to ensure the crowdworkers are consistently well-paid.

Acknowledgements

The authors thank Elisa Kreiss for helpful discussions. We also thank the anonymous reviewers and meta-reviewers for their helpful feedback. This research was supported in part by DARPA under the MCS program through NIWC Pacific (N66001-19-2-4031), and the Allen Institute for AI (AI2).

References

Anurag Acharya, Kartik Talamadupula, and Mark A Finlayson. 2021. An atlas of cultural commonsense for machine reasoning. In *AAAI*.

JL Ackrill. 1963. *Aristotle's Categories and De Interpretatione*. Clarendon Press.

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

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *ICLR*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2020. Dynamic Neuro-Symbolic Knowledge Graph Construction for Zero-shot Commonsense Question Answering. *arXiv: 1911.03876*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *ACL*.

Isaac Councill, Ryan McDonald, and Leonid Velikovich. 2010. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pages 51–59, Uppsala, Sweden. University of Antwerp.

Federico Fancellu and Bonnie Webber. 2015. Translating negation: A manual error analysis. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 2–11, Denver, Colorado. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*.

H Paul Grice, Peter Cole, Jerry Morgan, et al. 1975. Logic and conversation. *1975*, pages 41–58.

Winston Haynes. 2013. *Bonferroni Correction*, pages 154–154. Springer New York, New York, NY.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus). *arXiv: 1606.08415*.

Laurence R. Horn and Heinrich Wansing. 2020. Negation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, spring 2020 edition. Metaphysics Research Lab, Stanford University.

Md Mosharaf Hossain, Antonios Anastasopoulos, Eduardo Blanco, and Alexis Palmer. 2020. It's not a non-issue: Negation as a source of error in machine translation. *arXiv: 2010.05432*.

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: 2010.05953*.

Benjamin Jowett et al. 1892. *The Dialogues of Plato: Parmenides. Theaetetus. Sophist. Statesman. Philebus*, volume 4. Oxford University Press, American branch.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *arXiv: 1412.6980*.

Douglas B Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

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, Berlin, Germany. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP*.

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: 1907.11692*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Ian Pratt. 1994. *Defeasible Inference*, pages 85–107. Macmillan Education UK, London.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. 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 5043–5053, Hong Kong, China. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation frames: A data-driven investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–321, Berlin, Germany. Association for Computational Linguistics.

Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108, Lisboa, Portugal. Association for Computational Linguistics.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *NeurIPS*.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2020. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *AAAI*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. In *EMNLP*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Ming Xiang, Julian Grove, and Anastasia Giannakidou. 2016. Semantic and pragmatic processes in the comprehension of negation: An event related potential study of negative polarity sensitivity. *Journal of Neurolinguistics*, 38:71–88.

A Appendices

A.1 ANION Data Collection Details

Heuristic of Creating Logical and Semi-logical Negation Events For logical negation, with the majority of the original events being simple sentences with one predicate, our general rule of thumb is to negate the original event at the sentence level. Specifically, with respect to each original event, we first identify each tokens' part of speech (POS) tags via the NLTK toolkit[4]. Then, we insert the negation cue *not* after the subject of each sentence, with majority of the case the entity "PersonX," with few exceptions of "PersonX's" and "PersonX and PersonY."

To ensure the grammar correctness of the heuristically generated logical negation events, we add appropriate auxiliary verbs (*e.g.*, do, does, did, is, was, can, could, would, should, may, might) in accordance with the tenses (*e.g.*, present, past, future) of the original events. Since NLTK's POS parser fails to recognize some of the verbs that have both noun and verb usage (*e.g.*, "waters" the plant, "supports" her argument), we curate a list of dual-used words and map them manually. Also, while converting the original events to their logical negation counterparts, we revise grammar mistakes from ATOMIC and exclude awkward expressions as much as possible. In addition, to make the negation forms sound more natural, we replace the modifier "some" by "any" during conversion (*e.g.*, "PersonX buys some shoes" is converted to "PersonX doesn't buy any shoes"). For the minority of compound events with clauses or complex sentence structures, we disregard them for the purpose of ensuring the data quality.

For semi-logical negation events, we curate a list of semi-logical negation cues besides *not* from various sources[5] (Councill et al., 2010; Hossain et al., 2020; Kim et al., 2019) and categorize them into four types including affixes, single-word cues, multi-word cues and negative verbs (Table 1). We identify appropriate rules to insert each semi-logical negation cue in simple base events from ATOMIC consisting of a subject and a predicate. We apply the rules to original events from ATOMIC and randomly select at least 200 automatically generated semi-logical negation events per each negation cue for manual screening by the first author to avoid misplacement of negation cues and awkward expressions. In the end, we were able to identify 5,019 high quality semi-logical negation events originating from ATOMIC.

As a final quality control step of the constructed logical and semi-logical events, after obtaining the crowdsourced inferences for each event, we remove all events that annotators comment as "unclear," "doesn't make sense" or "grammatically wrong."

Crowdsourcing of Commonsense Contradiction Events For collecting commonsense contradiction events, we present an original ATOMIC event to the annotators and ask them to formulate corresponding opposite events. We exclude ATOMIC events with placeholders representing generic objects) to capture semantic and pragmatic subtlety. In the MTurk task, we present annotators detailed instructions of formulating the opposite events (*e.g.*, avoid using negative words as much as possible, use complete sentences, follow grammar rules) and concrete examples as references. Figure 2 shows details of the MTurk task. Although we explicitly instruct annotators to avoid using negation cues, there are still some exceptions. Therefore, after the compilation of all commonsense contradiction events, we remove ones that contain any explicit negation cues to make sure the categorization is clean.

Crowdsourcing of ANION Event Inferences For the collection of ANION event inferences, we adopt the MTurk templates used by the original ATOMIC data collection[6]. Similarly to logical and semi-logical events, we remove all inferences of events that annotators comment as "unclear," "doesn't make sense" or "grammatically wrong."

A.2 Training Details of COMET Models

Input A knowledge tuple $\{h, r, t\}$ is represented as a concatenated sequence with tokens of each element in the tuple: $X = \{X^h, X^r, X^t\}$ where $X^h = \{x_0^h, ..., x_{|h|}^h\}$ are the tokens comprising the event, $X^r = \{x_0^r, ..., x_{|r|}^r\}$ as tokens comprising the relation, and $X^t = \{x_0^t, ..., x_{|t|}^t\}$ are the tokens comprising the commonsense inference.

Initialization Similar to Bosselut et al. (2020), we initialize the trained parameters of COMET to the 345M parameter GPT2 model (GPT2-M) from Radford et al. (2019). Special tokens that

[4] https://www.nltk.org

[5] https://dictionary.cambridge.org/us/grammar/british-grammar/negation_2

[6] https://homes.cs.washington.edu/~msap/atomic/mTurkFiles/

represent relation types (*e.g.*, *xIntent*) are added to the vocabulary and initialized via sampling from the normal distribution.

Hyperparameters Following Bosselut et al. (2019), we use a dropout rate of 0.1 and GeLU (Hendrycks and Gimpel, 2020) units as activation functions. During training, we use the Adam optimizer (Kingma and Ba, 2017) with a batch size of 64. For COMET models trained on different subsets of the ATOMIC and ANION datasets, we adopt a maximum learning rate of 6.25e-5 with a warmup period of 0.002 times of the total number of minibatches customized for each model, which decays linearly until finishing training.

We train different COMET models for different subsets of the full on original data (ATOMIC), original and logical negation data (ATOMIC + ANION-L), original and semi-logical negation data (ATOMIC + ANION-S), original and commonsense contradiction data (ATOMIC + ANION-C), and the overall dataset (ATOMIC + ANION), for 21k, 25K, 24K, 24K and 29K minibatches respectively, and apply early stopping for all models. The rest of the hyperparameters are the same as those of GPT2-M in Radford et al. (2019) implemented via the publicly available HuggingFace API[7] .

All models are fine-tuned and evaluated on a single NVIDIA QUADRO RTX 8000 GPU for six to twelve hours depending on the complexity of the experimental setup.

A.3 Training Details of Negation Discriminator

Input As input to the discriminator model, we design sentence patterns that express relation types in natural language and fill out the patterned sentences with events and conditions before encoding them (*e.g.*, "PersonX addresses a talk. As a result, PersonX wants to convince others."). Relations and their corresponding patterned sentences are listed in Table 12. Adopting patterned sentences is found to be a more effective approach than concatenating components in knowledge tuples from the pilot study.

Loss Function The negation discriminator is trained to minimized the binary cross-entropy loss:

$$\mathcal{L}_D = y \cdot \log P(y) + (1-y) \cdot \log\left(1 - P(y)\right) \quad (2)$$

[7]https://huggingface.co/transformers/

Relation	Patterned sentences
xIntent	$\{h\}$. Because PersonX wanted $\{t\}$.
xNeed	$\{h\}$. Before, PersonX needed $\{t\}$.
xAttr	$\{h\}$. PersonX is seen as $\{t\}$.
xWant	$\{h\}$. As a result, PersonX wants $\{t\}$.
oWant	$\{h\}$. As a result, others want $\{t\}$.
xEffect	$\{h\}$. As a result, PersonX then $\{t\}$.
oEffect	$\{h\}$. As a result, others then $\{t\}$.
xReact	$\{h\}$. As a result, PersonX feels $\{t\}$.
oReact	$\{h\}$. As a result, others feel $\{t\}$.

Table 12: Patterned sentences representing relation types in ATOMIC, used to construct inputs for training negation discriminators.

where y is the label for an input (*i.e.*, logically valid or invalid).

Hyperparameters Parameters are initialized with the trained weights of the RoBERTa-base model in Liu et al. (2019). During training, we use the Adam optimizer (Kingma and Ba, 2017) and train the model with a batch size of 64. We adopt a maximum learning rate of 4.5e-5 with a warmup period of 10 minibatches. We trained **L**, **S**, **C**, **LSC** discriminators, for 25K, 14K, 21K and 6K minibatches respectively, and apply early stopping for all models. We use a probability threshold of 0.7 to determine whether an input knowledge tuples to the discriminator is plausible based on pilot study on the development sets. The rest of the hyperparameters are the same as those of RoBERTa-base (Liu et al., 2019) implemented via the publicly available HuggingFace API[8] .

All models are fine-tuned and evaluated on a single NVIDIA QUADRO RTX 8000 GPU for four to six hours depending on the different experimental setups.

A.4 Statistical Significance Testing

To compare P@{# valid} for the *all* and *valid* sets, we use a Permutation Test[9] with 1,000 permutations to test for statistical significance. For multiple comparisons, we use the Bonferroni method (Haynes, 2013) to correct significance thresholds.

A.5 Quality Check for the Human Evaluation

We conduct comprehensive pre- and post-evaluation screening on the users and the tasks being completed to ensure the objectivity and high quality of the evaluations. Besides qualifying users

[8]https://huggingface.co/transformers/

[9]http://rasbt.github.io/mlxtend/

Eval		ATOMIC			ANION-L			ANION-S			ANION-C		
Trn	**Dis**	all	valid	ipv%	all	valid	ipv%	all	valid	ipv%	all	valid	ipv%
ATOMIC	L	**55.69**	55.65	-0.07	39.46	****46.39**	17.55	37.13	**37.48**	0.96	**46.92**	46.83	-0.20
	S	55.93	**56.18**	0.44	37.85	****41.93**	10.78	39.29	****44.58**	13.47	47.32	**47.68**	0.75
	C	56.94	**57.26**	0.57	36.43	**37.54**	3.03	37.72	**39.03**	3.45	48.26	**48.79**	1.09
	LSC	58.30	**59.07**	1.32	39.45	****45.59**	15.57	38.55	****44.93**	16.56	48.81	***51.44**	5.40
ATOMIC+ ANION-L	L	58.62	**58.72**	0.16	46.05	****51.19**	11.16	42.42	**42.89**	1.11	**47.98**	47.97	-0.02
	S	58.93	**59.31**	0.64	45.90	****49.00**	6.77	44.22	****47.59**	7.64	48.10	**48.69**	1.24
	C	59.63	**60.07**	0.73	45.88	**46.23**	0.76	43.40	**43.74**	0.79	48.81	**49.84**	2.12
	LSC	60.83	**62.49**	2.74	45.96	****50.19**	9.20	44.61	****48.30**	8.27	49.73	***51.97**	4.51
ATOMIC+ ANION-S	L	**56.37**	56.35	-0.05	44.77	****51.76**	15.60	45.58	**45.87**	0.63	46.24	**46.29**	0.11
	S	56.60	**56.66**	0.11	44.39	****47.42**	6.83	46.07	****48.32**	4.89	46.62	**47.17**	1.19
	C	57.46	**57.60**	0.23	44.46	**45.39**	2.07	45.81	**47.15**	2.93	47.38	**48.83**	3.06
	LSC	58.74	***60.39**	2.81	44.94	****49.88**	10.98	46.08	****48.67**	5.62	48.56	****51.22**	5.46
ATOMIC+ ANION-C	L	52.72	**52.73**	0.02	43.45	****49.62**	14.20	41.83	**41.88**	0.12	48.93	**48.97**	0.07
	S	52.93	**53.33**	0.76	42.66	****46.40**	8.75	42.57	****46.40**	8.98	49.18	**49.49**	0.62
	C	53.70	**54.07**	0.69	42.83	**43.26**	1.00	42.25	**42.70**	1.07	49.30	***50.97**	3.38
	LSC	54.38	**55.74**	2.49	44.17	****48.84**	10.58	42.37	****46.22**	9.10	50.07	****52.80**	5.46
ATOMIC+ ANION	L	54.16	**54.20**	0.08	46.54	****50.71**	8.98	46.90	**47.14**	0.51	50.80	**50.94**	0.28
	S	54.49	**54.64**	0.28	46.26	****48.36**	4.54	47.73	****50.42**	5.65	51.29	**51.52**	0.45
	C	55.03	**55.71**	1.23	46.15	**46.16**	0.03	47.47	**48.20**	1.55	51.28	**52.65**	2.67
	LSC	55.68	****57.58**	3.41	46.39	****49.85**	7.45	47.53	****50.62**	6.50	51.83	***53.91**	4.02

Table 13: For generations of COMET models trained on different subsets of ATOMIC and ANION, the Precision @ {# valid} scores of the *all* and *valid* sets determined by **L**, **S**, **C** and **LSC** discriminators with respect to the original and negation evaluation sets. The single (*) and double asterisks (**) indicate significance at p<0.05 and p<0.01 respectively. iprv% is the percentage improvement of the *valid* set over the *all* set.

Eval		ATOMIC			ANION-L			ANION-S			ANION-C		
Trn	**Dis**	all	valid	iprv%	all	valid	iprv%	all	valid	iprv%	all	valid	iprv%
ATOMIC	L	59.41	**59.65**	0.40	44.92	****49.95**	11.20	39.47	**39.94**	1.21	50.77	**50.91**	0.27
	S	59.48	**60.14**	1.12	42.88	****46.24**	7.83	45.27	****49.25**	8.81	51.22	**51.84**	1.21
	C	59.89	**60.89**	1.66	39.20	**40.28**	2.75	40.07	**41.40**	3.32	51.77	**52.88**	2.15
	LSC	61.37	**63.12**	2.85	46.00	****50.34**	9.44	46.15	****50.23**	8.85	53.29	**55.24**	3.65
ATOMIC+ ANION-L	L	61.33	**61.57**	0.39	51.47	****56.16**	9.11	45.73	**46.04**	0.68	51.40	**51.57**	0.33
	S	61.13	**62.05**	1.51	50.12	****53.40**	6.54	50.84	****54.09**	6.40	51.89	**52.91**	1.96
	C	61.48	**62.96**	2.40	48.23	**49.06**	1.72	46.42	**46.99**	1.22	52.31	**53.67**	2.61
	LSC	63.66	***65.85**	3.44	51.90	****56.26**	8.40	51.12	****54.59**	6.78	53.97	**56.15**	4.04
ATOMIC+ ANION-S	L	60.25	**60.65**	0.67	48.11	****54.45**	13.18	45.97	**46.35**	0.81	50.82	**50.89**	0.15
	S	60.23	**60.85**	1.03	46.48	****49.14**	5.72	47.58	****50.29**	5.70	51.11	**52.00**	1.74
	C	60.43	**61.28**	1.40	44.61	**46.31**	3.80	46.21	****48.78**	5.58	51.72	**53.25**	2.95
	LSC	62.22	***64.44**	3.58	47.63	****51.12**	7.32	48.24	***50.70**	5.11	53.51	***56.04**	4.74
ATOMIC+ ANION-C	L	54.36	**54.80**	0.81	46.25	****51.57**	11.51	42.81	**43.13**	0.76	50.71	**50.81**	0.20
	S	54.50	**55.75**	2.29	45.59	***48.11**	5.53	45.40	****48.50**	6.83	51.00	**51.78**	1.52
	C	54.43	**55.50**	1.97	42.61	**43.26**	1.53	43.11	**44.13**	2.38	51.44	***53.46**	3.93
	LSC	55.68	***57.91**	4.01	47.11	****51.25**	8.80	45.75	****49.03**	7.18	52.44	****55.68**	6.18
ATOMIC+ ANION	L	56.63	**57.11**	0.85	50.39	****54.52**	8.20	47.92	**48.27**	0.73	53.11	**53.41**	0.56
	S	56.53	**57.42**	1.57	48.92	****52.07**	6.44	49.21	****52.51**	6.72	53.10	**53.67**	1.09
	C	56.40	**57.64**	2.21	47.96	**48.30**	0.70	48.16	**50.00**	3.82	53.90	**55.48**	2.94
	LSC	58.27	**60.53**	3.87	50.25	****54.27**	8.02	49.85	****53.09**	6.50	54.50	****57.66**	5.79

Table 14: For generations of COMET models trained on different subsets of ATOMIC and ANION, the Precision @ 3 scores of the *all* and *valid* sets determined by **L**, **S**, **C** and **LSC** discriminators with respect to the original and negation evaluation sets. The single (*) and double asterisks (**) indicate significance at p<0.05 and p<0.01 respectively. iprv% is the percentage improvement of the *valid* set over the *all* set.

Event + Rel	Generation	V	P
X does not skate around ***xAttr***	athletic	✗	✗
	careless	✗	✗
	lazy	✓	✓
	uncoordinated	✓	✓
	unskilled	✓	✓
X does not sit behind Y ***xIntent***	to be alone	✓	✓
	to be left alone	✓	✓
	to avoid Y	✓	✓
	to sit	✗	✗
	to wait	✓	✗
X does not look angry ***xNeed***	to calm down	✗	✓
	to watch a movie	✓	✗
	to have been provoked	✗	✗
	to not be angry	✓	✓
	to be calm	✓	✓
X does not rent an apartment ***xWant***	to save money	✓	✓
	to get a job	✓	✗
	to pay rent	✗	✗
	to move in	✗	✗
	to get a new apartment	✓	✓
X is not offered the job ***oWant***	to hire X	✗	✗
	to fire X	✓	✓
	to hire someone else	✓	✓
	to accept the job	✗	✗
	to hire them	✗	✗
X does not buy a snack ***xReact***	satisfied	✗	✗
	hungry	✓	✓
	satiated	✗	✗
	full	✗	✗
	guilty	✓	✗
X does not bring Y into conflict ***oReact***	relieved	✓	✓
	sad	✗	✗
	satisfied	✓	✓
	grateful	✓	✓
	angry	✗	✗
X does not learn new things ***xEffect***	gains knowledge	✗	✗
	becomes lazy	✓	✓
	gets bored	✓	✓
	becomes ignorant	✗	✓
	cries	✓	✓
X does not put Y in mind ***oEffect***	becomes confused	✗	✗
	does not think about X	✓	✓
	Y thinks about X	✗	✗
	Y is not remembered	✓	✓
	cries	✗	✓

Table 15: Randomly selected generations of the original COMET model regarding logical negation events in ANION-L. The top 5 options are classified as either *valid* or *invalid* by the **LSC** discriminator. **V** indicates whether an option is classified as *valid* by the **LSC** discriminator. **P** indicates whether an option is plausible judging by humans.

Event + Rel	Generation	V	P
X hardly ever increases X's knowledge ***xAttr***	intelligent	✗	✗
	determined	✗	✗
	studious	✗	✗
	lazy	✓	✓
	dedicated	✗	✗
X avoids skating around ***xIntent***	to have fun	✗	✗
	to be safe	✓	✓
	to stay home	✓	✓
	to stay in shape	✗	✗
	to get fit	✗	✗
X not at all wants to learn karate ***xNeed***	learn karate	✗	✗
	to not like it	✓	✓
	to avoid it	✓	✓
	to be lazy	✓	✓
	to find a teacher	✗	✗
X refuses to hear a scary noise ***xWant***	to run away	✗	✗
	to go to sleep	✓	✓
	to be safe	✓	✓
	to keep quiet	✓	✓
	to avoid the noise	✓	✓
X never brings Y into conflicts ***oWant***	to avoid X	✗	✗
	to be left alone	✗	✓
	to thank X	✓	✓
	to fight back	✗	✗
	to avoid conflict	✗	✓
X scarcely gets sunburned ***xReact***	burned	✗	✗
	hurt	✗	✗
	sick	✗	✗
	sad	✗	✗
	satisfied	✓	✓
X under no circumstances forgets Y's wallet ***oReact***	upset	✗	✗
	sad	✗	✗
	angry	✗	✗
	thankful	✓	✓
	grateful	✓	✓
X has trouble with advertising X's business ***xEffect***	loses money	✓	✓
	loses clients	✓	✓
	gets fired	✓	✓
	gets sued	✗	✗
	cries	✓	✓
X fails to make it through the day ***oEffect***	loses a friend	✓	✓
	worries about X	✓	✓
	worried	✓	✓
	want them to do better	✓	✓
	cries	✓	✗

Table 16: Randomly selected generations of the original COMET model regarding semi-logical negation events from ANION-S. The top 5 options are classified as either *valid* or *invalid* by the **LSC** discriminator. **V** indicates whether an option is classified as *valid* by the **LSC** discriminator. **P** indicates whether an option is plausible judging by humans.

during pilot batches, we double check to remove evaluation tasks that are not carefully conducted (*e.g.*, tasks done by users that select all/no options for all hundreds of tasks that they perform). Figure 3 shows a snippet of the human evaluation MTurk task.

Event + Rel	Generation	V	P
X keeps the old one ***xAttr***	determined	✗	✗
	careful	✗	✗
	loyal	✓	✓
	tired	✗	✗
	caring	✗	✗
X pays twice the retail price for a rare item ***xIntent***	to own something rare	✓	✓
	to buy something rare	✓	✓
	to purchase something rare	✓	✓
	to obtain something	✗	✗
	to acquire something	✗	✗
X is a complete loner ***xNeed***	to avoid people	✓	✓
	to be lonely	✓	✓
	to start a relationship	✗	✗
	to stay quiet	✓	✓
	to get into trouble	✗	✗
X is miserable playing ***xWant***	to get better	✓	✓
	to take a break	✗	✓
	to go home	✗	✗
	to go to sleep	✗	✗
	to cry	✓	✓
X is picky about everything ***oWant***	to ignore X	✓	✓
	to avoid X	✓	✓
	to talk to X	✗	✗
	to help X	✗	✗
	to make X feel better	✓	✗
X resigns himself ***xReact***	relieved	✓	✓
	relaxed	✓	✗
	satisfied	✗	✗
	accomplished	✗	✗
	sad	✓	✓
X gives away X's laptop ***oReact***	grateful	✓	✓
	thankful	✓	✓
	upset	✗	✗
	sad	✗	✗
	surprised	✓	✓
X goes home ***xEffect***	relaxes	✓	✓
	goes to sleep	✓	✓
	is greeted by family	✗	✓
	gets rest	✓	✓
	gets tired	✗	✗
X puts Y out of mind ***oEffect***	has a better day	✗	✗
	becomes sad	✓	✓
	cries	✓	✓
	becomes grateful towards X	✗	✗
	feels better	✗	✗

Table 17: Randomly selected generations of the original COMET model regarding commonsense contradiction events from ANION-C. The top 5 options are classified as either *valid* or *invalid* by the **LSC** discriminator. **V** indicates whether an option is classified as *valid* by the **LSC** discriminator. **P** indicates whether an option is plausible judging by humans.

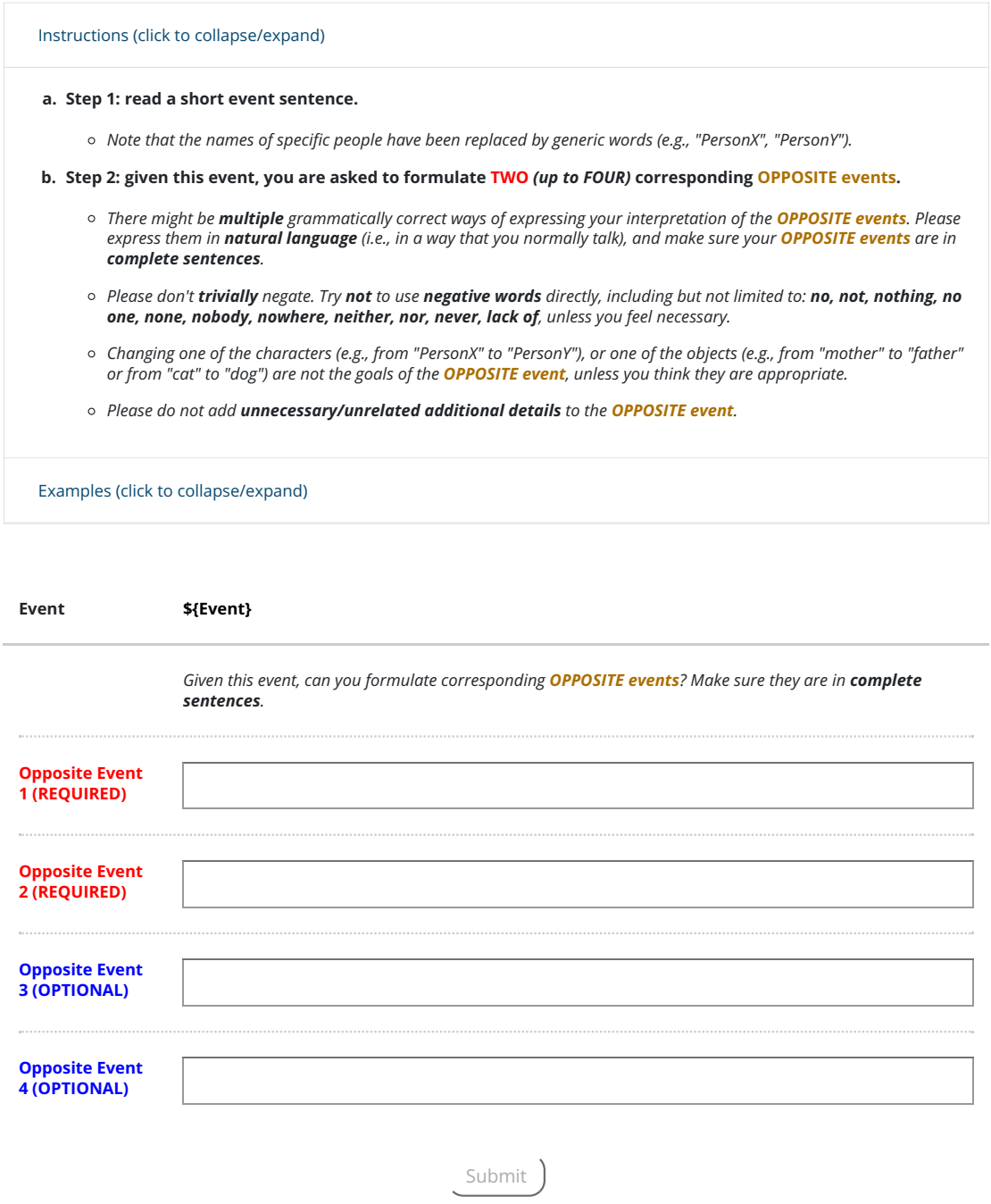

Figure 2: Snippet of the annotation task used to collect commonsense contradiction events.

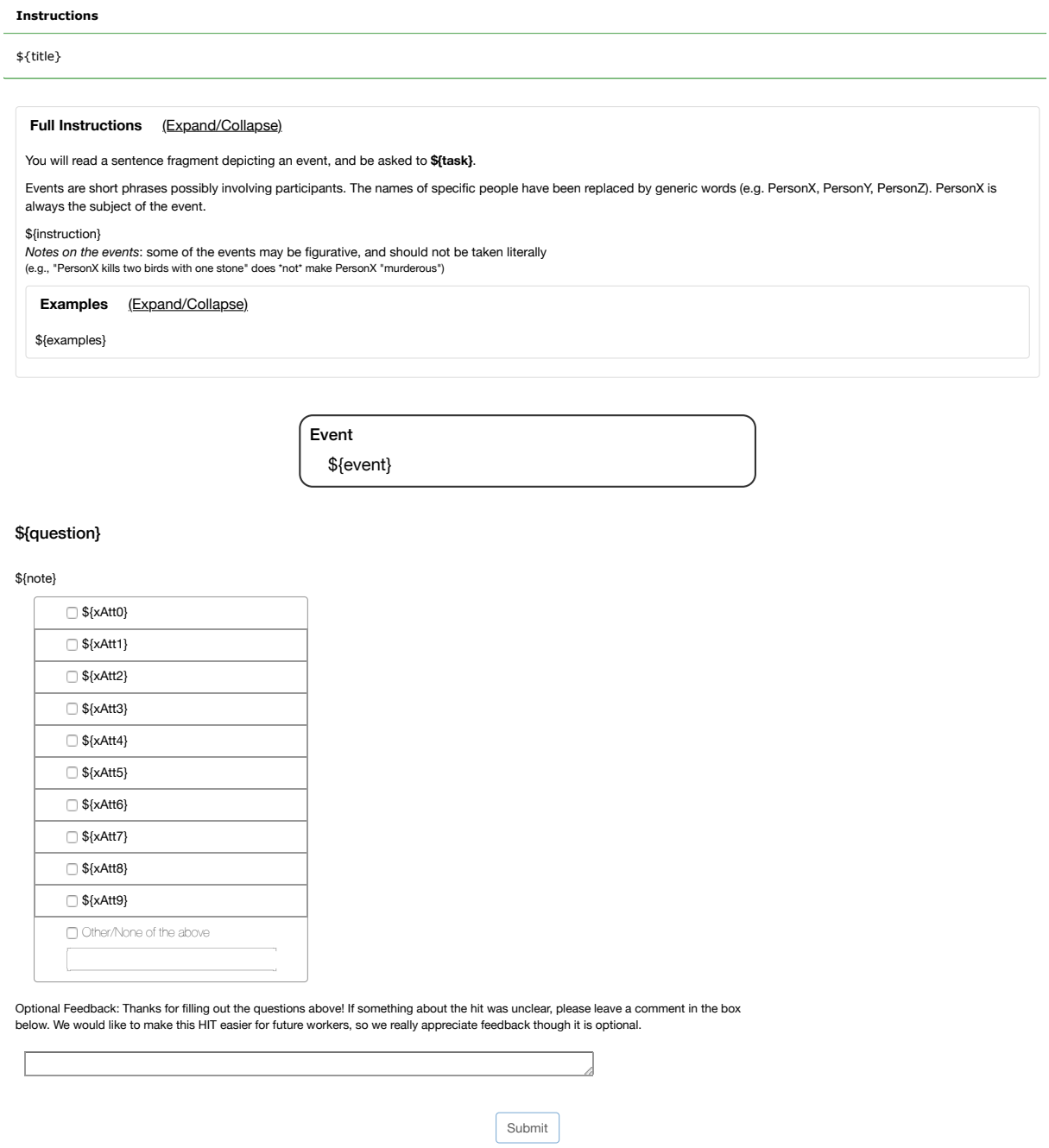

Figure 3: Snippet of the human evaluation task used to evaluate model generated tail inferences.