Abductive Commonsense Reasoning Exploiting Mutually Exclusive Explanations

Wenting Zhao and **Justin T. Chiu** and **Claire Cardie** and **Alexander M. Rush**
Department of Computer Science
Cornell University
{wz346,jtc257,ctc9,arush}@cornell.edu

Abstract

Abductive reasoning aims to find *plausible* explanations for an event. This style of reasoning is critical for commonsense tasks where there are often multiple plausible explanations. Existing approaches for abductive reasoning in natural language processing (NLP) often rely on manually generated annotations for supervision; however, such annotations can be subjective and biased. Instead of using direct supervision, this work proposes an approach for abductive commonsense reasoning that exploits the fact that only a subset of explanations is correct for a given context. The method uses posterior regularization to enforce a mutual exclusion constraint, encouraging the model to learn the distinction between fluent explanations and plausible ones. We evaluate our approach on a diverse set of abductive reasoning datasets; experimental results show that our approach outperforms or is comparable to directly applying pretrained language models in a zero-shot manner and other knowledge-augmented zero-shot methods.

1 Introduction

Abductive reasoning aims to find *plausible* explanations for an event (Paul, 1993). Unlike deduction, which draws a firm conclusion from a set of premises, abduction requires reasoning from an outcome to plausible explanations. Fig. 1 (top) demonstrates the distinction: given only the context x, both the blue and the red sentences describe possible subsequent events; however, upon seeing the outcome y only one of the two is a plausible explanation (although there may be others). Humans apply abduction in everyday situations (Andersen, 1973) such as reading-between-the-lines (Charniak and Shimony, 1990) and analyzing causes and effects (Thagard and Shelley, 1997; Pearl and Mackenzie, 2018). Learning to perform abduction is thus an important step towards building human-like machines with commonsense knowledge.

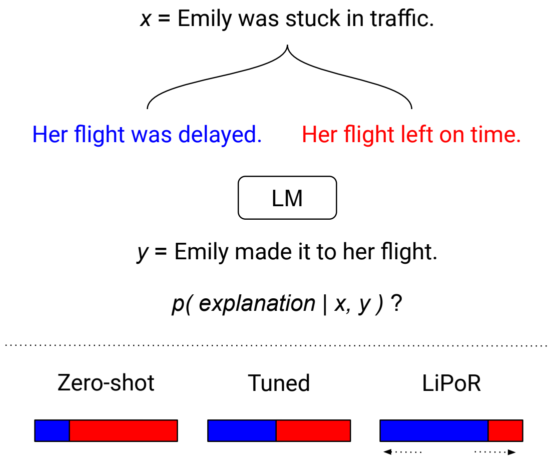
x = Emily was stuck in traffic.
Her flight was delayed.
Her flight left on time.
LM
y = Emily made it to her flight.
p(explanation | x, y) ?
Zero-shot
Tuned
LiPoR

Figure 1: **Top:** An abductive reasoning example consisting of a context x, an outcome y, and two candidate explanations. The goal is to identify the plausible explanation given x and y. To predict an explanation, one can apply a pretrained language model (shown as LM) to score y given x and an explanation, and then compute the posterior probability for the explanation. **Bottom:** Using a LM without fine-tuning (Zero-shot) leads to poor performance, whereas a LM fine-tuned via max-marginal likelihood (Tuned) fails to distinguish the two explanations. LiPoR is trained to partition the explanations in a mutually exclusive manner.

Abductive reasoning has been extensively studied in the setting where annotations are available (Storks et al., 2019). However, because determining whether an explanation is plausible is a subjective and noisy process, annotating plausibility of explanations can be problematic for commonsense reasoning problems. Zhang et al. (2020) show that, in a dataset verification step where five annotators are asked to determine whether a handwritten explanation is plausible, they disagree with each other on 62.34% of 1365 explanations. This subjectivity thus introduces annotator-specific bias as has been seen in related tasks (Elazar et al., 2021;

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 14883–14896
July 9-14, 2023 ©2023 Association for Computational Linguistics

Geva et al., 2019). The potential bias in plausibility annotation motivates the study of learning to perform abductive reasoning without plausibility annotations. Thus, we consider the setting where the context x and outcome y are observed, and models must learn to identify plausible explanations out of a given set of candidate explanations, without direct supervision over plausibility.

Rule-based methods use formal logic to reason about explanations (Paul, 1993); however, their limited coverage prevents them from scaling to the full complexity of natural language. Recently, pretrained language models, which have achieved remarkable performance on a range of NLP tasks (Li et al., 2020; Wei et al., 2022a), hold the potential for zero-shot abductive reasoning. Specifically, Bhagavatula et al. (2019) directly estimate the probability of an explanation for an outcome through Bayes' Rule (*Zero-shot* in Fig. 1). In practice, however, this direct approach can often lead to performance that is only slightly better than random guessing (Zhang et al., 2020; Zhou et al., 2021b).

To avoid these issues, we reduce abductive reasoning down to a single constraint — an explanation must be *plausible* or *implausible*. This restriction, argued by Gordon and Hobbs (2017), enforces that explanations are mutually exclusive; that is, one explanation being plausible automatically rules out some other explanations. We introduce **Li**kelihood learning with **Po**sterior **R**egularization (LiPoR), an approach to perform abductive reasoning that only leverages mutual exclusivity of explanations and does not rely on plausibility annotations. Specifically, we maximize the marginal likelihood of the outcome given the context and a set of explanations (*Tuned* in Fig 1), then use posterior regularization to enforce mutual exclusion between plausible and implausible explanations (*LiPoR* in Fig 1). We show how to impose this relation with a simple distributional constraint on the posterior of the model.

We empirically evaluate LiPoR on a diverse set of abductive reasoning datasets. Specifically, we consider four datasets under the abductive reasoning framework: αNLI (Bhagavatula et al., 2019), Sen-Making (Wang et al., 2019), δ-NLI (Rudinger et al., 2020), and WinoWhy (Zhang et al., 2020). Results show that LiPoR consistently outperforms pretrained language models directly applied in a zero-shot manner and is comparable to different variants of a state-of-the-art knowledge-augmented zero-shot method (Ma et al., 2021). As human-written explanation candidates are not always available during fine-tuning, we further evaluate LiPoR on the explanation candidates generated via prompting (Brown et al., 2020). We show that, even though automatically generated explanations are noisy, LiPoR can still leverage them and outperform strong zero-shot models including GPT3.

2 Related Work

Zero-shot commonsense reasoning. We categorize zero-shot approaches for commonsense reasoning into two groups. The first group uses pretrained language models as a source of world knowledge. Shwartz et al. (2020); Zhou et al. (2021a) query the language models with information seeking questions to identify background knowledge relevant to specific examples, and the answers returned by the models are later used as additional information for producing the final outputs. Dou and Peng (2022) convert multiple-choice QA to cloze-style sentences and have the language models score different answers. Qin et al. (2020) proposed a decoding algorithm that generates free-form explanations by considering the future contexts through backpropagation. Our approach also uses pretrained language models as a source of knowledge, but we perform additional maximum likelihood fine-tuning to fit the abductive task data.

The second group leverages external knowledge bases (KBs). Bosselut et al. (2021) leverage COMET (Bosselut et al., 2019), a dynamic knowledge graph, to generate a chain of commonsense inferences based on contexts of QA examples, which can be treated as explanations. Banerjee and Baral (2020); Ma et al. (2021) pretrain language models on artificial question answering (QA) datasets, created from knowledge graphs; a system trained on such datasets can directly perform zero-shot QA. Huang et al. (2021) formulate multiple-choice QA as natural language inference (NLI) and leverage both existing NLI datasets and KBs to identify answer choices in a zero-shot manner.

Relation to deductive reasoning. Both abduction and deduction have intermediate explanations. Abductive reasoning infers the most likely explanation from outcomes. In contrast, deductive reasoning infers a conclusion given a complete set of premises. However, outcomes are often not a direct result of premises but come from a chain of reasoning over intermediate explanations. Identifying and providing the correct chain of reasoning is crucial to building trustworthy systems.

Within the realm of deduction there are several different approaches that utilize neural models. Bostrom et al. (2021) develop a pipeline to automatically construct training examples from Wikipedia, so that a system trained on such data is able to generate deductive inferences from natural language inputs without direct human supervision. Arabshahi et al. (2021) present a neuro-symbolic theorem prover that extracts intermediate reasoning steps for understanding conversations. Rajani et al. (2019); Tafjord et al. (2021); Nye et al. (2022); Wei et al. (2022b) collect human annotated explanations for training interpretable systems which first generate intermediate explanations and then produce the final task outputs.

Explanations as latent variables. Modeling intermediate explanations as latent variables is a common approach, although training and inference details differ. Here we consider representative works in NLP. Zhou et al. (2020) apply a latent variable model to language understanding and train the model with variational expectation maximization. Their method can generate free-form explanations but requires a small set of labeled examples for supervision. Zhou et al. (2021b) apply such a model to probe dialogue generation in a zero-shot manner. Vig et al. (2020) apply a latent variable model to analyze gender bias in large pretrained language models by viewing the behaviors of neurons as unobserved explanations. Lei et al. (2016); Vafa et al. (2021) apply such a model to identify rationales for sequence classification/generation, where rationales are a minimal subset of inputs or previous words that can lead to the same predictions. LiPoR is a training scheme developed for learning such latent-variable models for abductive reasoning, which has a unique challenge of identifying multiple plausible explanations.

3 Abductive Reasoning

We consider four datasets that test abductive reasoning skills. While abduction can be difficult to pinpoint, we select datasets that obey the following criteria: there is a need for differentiating plausible explanations from implausible explanations, there is an observed outcome, and the outcome depends on intermediate explanations. Based on these criteria, we use αNLI (Bhagavatula et al., 2019), Sen-Making (Wang et al., 2019), δ-NLI (Rudinger et al., 2020), and WinoWhy (Zhang et al., 2020) as our target datasets.

αNLI	x:	it was a very hot summer day
	z:	{he decided to run in the heat, **he drank a glass of ice cold water**}
	y:	he felt much better
Sen-Making	z:	{**a restaurant does not have doctors or medical treatment**, a restaurant is usually too noisy for a patient, there are different types of restaurants in the city}
	y:	it is not true that he was sent to a restaurant for treatment
δ-NLI	x:	four people and a child walking in the street
	z:	{**people from all over the world are gathered in the area**, **the people buy cotton candy from a booth**, the family is the only humans in the area, the family is walking their dog}
	y:	the family is enjoying the world's fair
WinoWhy	x:	the fish ate the worm, it was hungry
	z:	{**hungry staff tend to eat**, worm is one being eaten, the worm is a common name for a variety of fish
	y:	therefore, it refers to the fish

Table 1: Examples conversions from different datasets. Every dataset comes with candidate explanations (shown in the pink cells), and only a subset of them are plausible explanations (shown in boldface). We set x in Sen-Making dataset to empty.

To convert each to the abduction format, we first identify a context x, which sets a scope for candidate explanations $\mathcal{Z}$, as well as an outcome y. The outcome could either be an event caused by z or a conclusion reached by z. Importantly, we differentiate explanation candidates $\mathcal{Z}$ as ones that are consistent with x, from plausible explanations that are consistent with both x and y. A central assumption is that training abductive reasoning systems with the candidate set introduces less noise and subjectivity than directly supervising the systems with plausibility annotations.

Example conversions of each dataset are shown in Table 1. Because αNLI is designed as an abduction task, the conversion is straightforward. Sen-Making is a benchmark that tests if a system can identify the reason why a statement is against common sense. In this case, a context is not required. We turn the nonsensical statement into a negative sentence, which becomes y. Then the original answer choices become z. δ-NLI is a defeasible inference task, which requires deciding whether new evidence has strengthened or weakened the original hypothesis. δ-NLI is made of extensions to three existing inference datasets: SNLI (Bowman et al., 2015), ATOMIC (Sap et al., 2019), and SOCIAL-CHEM-101 (Forbes et al., 2020); each of them will be referred to as δ-N for brevity, where N can be replaced by a dataset name. We map premises and hypotheses to contexts and outcomes, respectively. We then turn updates that strengthen a hypothesis into a plausible explanation and updates that weaken a hypothesis into an implausible explanation. WinoWhy is a follow-up task for Winograd Schema Challenge (WSC) (Levesque et al., 2012): Given the pronoun coreference resolution question and the answer from a WSC example, WinoWhy seeks to select all plausible reasons for why the pronoun is resolved to the answer. We thus turn the question of the WSC example into a context x and the answer into a declarative sentence y.

Notably these datasets differ in the number of plausible explanations, which we denote by a value $m \geq 1$. In αNLI and Sen-Making, m is fixed to 1 for all examples. However, in δ-NLI and WinoWhy, m is variable, and we assume that half of explanations are plausible. However these explanations are discrete; an explanation is either plausible or implausible. A successful unsupervised system should assign high probabilities to plausible explanations and low probabilities to implausible explanations. This discreteness is encoded into some of the tasks directly. For example, Bhagavatula et al. (2019); Zhang et al. (2020) instruct the annotators to make minimal possible changes to plausible explanations to produce implausible explanations, so that a system would fail if it predicts explanations based on superficial lexical features.

4 LiPoR

We now describe LiPoR, a method to adapt pretrained language models to incorporate mutual exclusivity between explanations. As we have seen, an abductive reasoning example consists of a context x, an observed outcome y, and an unobserved explanation $z \in \mathcal{Z}$, which, together with x, has led to y. Importantly, the candidate set of explanations $\mathcal{Z}$ is given during training but the plausibility of each explanation is not.[1] The goal of abductive reasoning is to produce a distribution over explanations z, defined by $p(z|x, y)$. We are interested in modeling the joint distribution $p(y, z|x)$, which is factored as follows:

$$p(y, z|x) = p(y|x, z)p(z|x) \tag{1}$$

Given Eq 1, the posterior distribution can be obtained via the Bayes' rule,

$$p(z|x, y) = \frac{p(y|z, x)p(z|x)}{p(y|x)}. \tag{2}$$

Because x itself does not provide further information for z, we set $p(z|x)$ to be a uniform distribution. Therefore, we only parameterize $p(y|x, z)$.

4.1 Baseline: Fine-tuning via Max-marginal Likelihood

We note that any off-the-shelf pretrained language model can be applied to evaluate $p(z|x, y)$ for an abductive reasoning task in a zero-shot fashion. To adapt the pretrained model to a specific task distribution without plausibility annotations, we maximize the following marginal likelihood function $\mathcal{L}(\cdot)$ with respect to parameters θ for all examples:

$$\mathcal{L}(\theta) = \log \sum_{z \in \mathcal{Z}} p_\theta(y|x, z)p(z|x). \tag{3}$$

Maximizing the marginal likelihood encourages the model to prefer explanations that assign the outcome high probability. Mechanically, the marginal likelihood requires computing the probability of the outcome given every explanation in the set $\mathcal{Z}$. Training then gives credit (gradient) to explanations that assign high probability to the outcome, encouraging the model to prefer explanations that explain the outcome. We parameterize $p(y|x, z)$ by θ, a language model, that takes "x [SEP] z" as input and returns a probability distribution over y. By optimizing this objective, we find θ under which $p(y|x)$ has a high likelihood, thus shifting the pretrained model to the new task-specific distribution. Furthermore, this objective does not require plausibility annotations for explanations.

4.2 Incorporating Mutual Exclusivity

The goal of abductive reasoning is to separate out plausible and implausible explanations. However, we note that $\mathcal{L}(\theta)$ itself only maximizes $p(y|x)$. In practice, this does not require the model to learn any distinctions between explanations, and we observe that in practice the approach learns

[1] While manually collecting $\mathcal{Z}$ can be expensive, we also show that $\mathcal{Z}$ can be also be obtained cheaply via language model prompting in Sec. 7.

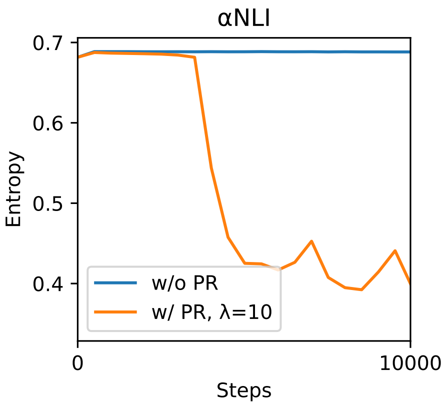

Figure 2: Entropy of $p(z|x, y)$ on αNLI at different training steps. The orange line and the blue line represent with and without PR, respectively. Without PR the model never learns to distinguish between explanations.

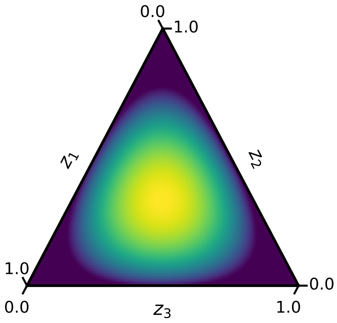

Figure 3: Visualization of $\Omega(\cdot)$ for $|\mathcal{Z}| = 3$ and $m = 2$. The lighter colors correspond to larger values. This constraint penalizes models that select too many plausible explanations.

to treat them all as plausible. The blue line in Fig 2 shows the entropy of $p(z|x, y)$ on the αNLI dataset when fine-tuning a model with $\mathcal{L}(\theta)$. We note that a uniform distribution of two categories has approximately an entropy of 0.6931, the upper bound on the entropy of $p(z|x, y)$ for the αNLI examples. Fine-tuning via max-marginal likelihood alone yields an entropy close to the upper bound, meaning the model believes that different z explain y equally well.

To impose the mutual exclusivity among explanations, we apply posterior regularization (PR), which places soft constraints on posterior distributions (Ganchev et al., 2010). The posterior regularized likelihood shows as follows:

$$\mathcal{L}_{PR}(\theta) = \mathcal{L}(\theta) - \lambda\Omega(p_\theta(z|x, y)). \quad (4)$$

To enforce a model to prefer specific explanations over the others, we choose $\Omega : \mathcal{R}^{|\mathcal{Z}|} \to \mathcal{R}$ to be the following function, proposed in Chen et al. (2020):

$$\Omega(p(z|x, y)) = \max(H(p_\theta(z|x, y)), \ln(m)) \quad (5)$$

$H(\cdot)$ is the entropy function. In Fig. 3, we plot $\Omega(\cdot)$ when $|\mathcal{Z}| = 3$ and $m = 2$, which shows that distributions with a non-zero probability for the third explanation have larger Ω values. $\Omega(\cdot)$ thus penalizes a posterior distribution that has an entropy higher than $\ln(m)$, which sets an upper bound at the entropy of a distribution whose probability mass collapses to m categories. When $m = 1$, $\Omega(\cdot)$ reduces to

$$\Omega(p(z|x, y)) = H(p_\theta(z|x, y)). \quad (6)$$

The orange line in Fig. 2 shows that incorporating $\Omega(\cdot)$ enables the model to differentiate between different explanations. Notice that, except for $m = 1$, there is no guarantee that $\Omega(\cdot)$ penalizes all distributions that have probability mass in more than m categories, but we will empirically justify that $\Omega(\cdot)$ eliminates undesired posterior distributions.

5 Experimental Setup

Metrics. Accuracy is used to evaluate a system's predictive power. For datasets with $m = 1$, accuracy is computed with regards to each example (i.e., whether the plausible explanation has been identified for each example). Otherwise, to stay consistent with evaluation in prior works, we compute accuracy with regards to each explanation (i.e., whether the plausibility of each explanation is correctly predicted). Therefore, more weight will be given to the instances that have larger $|\mathcal{Z}|$ (within a single dataset, the variance of $|\mathcal{Z}|$ for different examples is very small).

Baselines. We consider three groups of baselines: (1) methods that do not rely on plausibility annotations (shown as w/o annotations), (2) pretrained language models fine-tuned with plausibility annotations (shown as w/ annotations), and (3) methods that incorporate external knowledge bases (shown as "w/ KBs"). For (1), we first consider previous best published results achieved by a RoBERTa-large model for αNLI (Ma et al., 2021), by a BERT model for Sen-Making (Wang et al., 2019), and a GPT-small model for WinoWhy (Zhang et al., 2020) (all abbreviated as Prev. Best). Additionally, we use GPT-Neo (Black et al., 2021), GPT3 (text-davinci-002) (Brown et al., 2020), and the

		αNLI	Sen-Making	δ-ATOMIC	δ-SNLI	δ-SOCIAL	WinoWhy
w/o annotations	Previous Best	65.50	45.60	-	-	-	56.37
	ZS GPT-NEO	57.47	29.80	47.53	45.38	51.69	59.13
	ZS GPT3	67.54	43.00	50.73	49.69	49.22	50.99
	ZS BART	50.96	47.80	59.05	55.12	52.58	45.69
	Tuned BART	57.40	63.50	67.49	64.76	53.88	55.32
	LiPoR	**71.56**	**65.50**	**76.82**	**65.26**	**57.19**	**69.88**
w/ annotations	RoBERTa	85.60	93.10	78.30	81.60	86.20	75.04
w/ KB	KDDC-ATOMIC (N)	70.80	51.00	75.90	69.83	64.49	42.44
	KDDC-CWWV (N)	70.00	45.70	62.48	63.24	62.90	40.45
	KDDC-CSKG (N)	70.50	49.60	72.20	69.93	63.80	44.05
	QNLI-ATOMIC (N)	-	-	-	-	-	73.47
	Previous Best (Y)	87.30	95.00	-	-	-	87.55

Table 2: Accuracy for identifying plausible explanations using methods with and without plausibility annotations. On each dataset, we boldface the best result within the methods without annotations. Suffix (Y) / (N) denotes whether a knowledge-augmented method use (Y) or not use (N) annotations, respectively.

BART-large model (Lewis et al., 2020) to directly score x [SEP] z [SEP] y for each z in a zero-shot (ZS) manner. We threshold the outputs of these models in the same way as done in our method to choose the plausible explanations. Finally, we consider BART fine-tuned with Eq. 3 (Tuned BART) as a baseline to better understand the role of posterior regularization. For (2), a RoBERTa-large model (Liu et al., 2019) is fine-tuned with plausibility annotations (abbreviated as RoBERTa). For this baseline, we refer to the best result in the literature: Ma et al. (2021) for αNLI, Wang et al. (2020) for Sen-Making, Rudinger et al. (2020) for δ-NLI, and Zhang et al. (2020) for WinoWhy. For (3), we run different variants of Knowledge-driven Data Construction (abbreviated as KDDC) (Ma et al., 2021), a method that leverages external knolwedge but not plausibility annotations. We note that KDDC is designed to predict a single correct answer with argmax. To handle the datasets that have more than one correct answers, we modify KDDC to choose the answers that have scores higher than the median. We also include Knowledge-Enabled Natural Language Inference (Huang et al., 2021) that is first supervised on QNLI (Wang et al., 2018) and then incorporate ATOMIC at inference time for WinoWhy (abbreviated as QNLI-ATOMIC). For models that use both external knowledge and plausibility annotations, we take RAINBOW (Raina et al., 2005) for αNLI, ECNU-SenseMaker (Zhao et al., 2020) for Sen-Making, and RoBERTa-Grande (Zhang et al., 2020) for WinoWhy.

Prompt for plausible explanations: Provide a brief explanation for why it is not sensible that y
Prompt for implausible explanations: Provide a brief explanation for why y

y: He poured orange juice on his cereal.

In: *Provide a brief explanation for why it is not sensible that he poured orange juice on his cereal.*
Out: It is not sensible because orange juice does not go well with cereal.

In: *Provide a brief explanation for why he poured orange juice on his cereal*
Out: He wanted to eat a healthy breakfast.

Figure 4: Prompts for producing competing explanations, followed by an example generation.

Implementation & Hyperparameters. We choose a BART-large model (Lewis et al., 2020) to be θ. We train the model with the Hugging Face Transformers framework (Wolf et al., 2020). We perform grid search with learning rates {1e-6, 3e-6, 5e-6, 1e-5}, batch sizes {2,4,8,16}, and λ {1e-2,1e-1,1,10}. We train 50 epochs for WinoWhy and 10 epochs for all other datasets. We perform evaluation on dev sets every 500 steps. We choose the checkpoint whose posterior distributions have the lowest average entropy on dev sets to run tests if the entropy starts to diverge during training. If the entropy converges, we choose the checkpoint at the end of training.

Because there are not train/dev/test sets for WinoWhy, to perform a direct comparison with other methods, we do not split the dataset ourselves and simply train models on all of the data and choose the checkpoint based on loss values.

Automatic Candidate Generation LiPoR assumes access to a candidate explanation set $\mathcal{Z}$ during training with human-written explanations. However, we may also want to use the model in domains without a candidate set. We consider a variant that uses a noisy automatic candidate generation process. In this setting, set $\tilde{\mathcal{Z}}$ will contain a set of explanations with no guarantee that any are plausible.

To generate $\tilde{\mathcal{Z}}$ we utilize language model prompting with GPT3 (text-davinci-002) (Brown et al., 2020). Using prompt templates inspired by the instructions given to human annotators, we have the model generate explanations. We show example prompts for the Sen-Making dataset in Fig. 4. For datasets with fewer than 1000 unique contexts x (i.e., δ-NLI and Winowhy), we generate one plausible explanation and one implausible explanation for every x. For the other datasets, we randomly sample 1000 unique contexts and otherwise stay the same. We release the prompts as well as the generated explanations for every dataset in the supplementary materials.

In this setting, LiPoR uses a lower PR penalty $\lambda = 0.1$. We additionally consider two more baselines. First, we score the most plausible explanation with the prompt as a prefix (denoted as Prompted GPT3). Secondly, we supervise RoBERTa-large with the generated explanations.

6 Results

We summarize the results in Table 2. First of all, LiPoR produces the best results compared to all other methods without plausibility annotations, including GPT3 which has many more parameters and is pretrained on more data. We note that LiPoR consistently outperforms Tuned BART, suggesting that posterior regularization plays a positive role in selecting plausible explanations. Compared to knowledge-augmented methods without plausibility annotations, LiPoR is able to produce better results on αNLI, Sen-Making, and δ-ATOMIC. We note that δ-NLI is in part created from knowledge bases, and therefore KDDC-* is particularly good at δ-ATOMIC, δ-SNLI, and δ-SOCIAL, but fail on WinoWhy and Sen-Making. Additionally, QNLI-ATOMIC outperforms LiPoR by 4 points on Winowhy, but this improvement is expected given how much related task data it was pretrained on. Finally, LiPoR still cannot match the performance of RoBERTa trained with plausibility annotations.

In Table 4, we show the confusion matrices for comparing among ZS BART, Tuned BART, and LiPoR on the αNLI test set. Tuned BART and LiPoR make the same predictions on a majority of examples, and on the instances they disagree, LiPoR is able to correctly identify plausible explanations on twice as many examples. We also observe a similar trend for ZS BART and Tuned BART.

Fine-tuning with Generated Explanations Table 3 compares LiPoR fine-tuned with generated explanation candidates to the best performing methods without plausibility annotations. Even with noisy candidate sets, LiPoR is still able to leverage such data. It outperforms zero-shot GPT3 methods and improves over Prompted GPT3. Additionally, LiPoR is more robust than RoBERTa trained with plausibility annotations when such annotations are noisy. Therefore, even though the generated explanations by themselves correlate weakly with plausibility, they can be used in LiPoR.

7 Analysis

Preserving Plausible Candidates Models trained to prefer single plausible explanations can become overconfident in their predictions. A major benefit of LiPoR is that it considers multiple plausible candidates. While LiPoR is fine-tuned to favor mutual exclusivity, we find that at test time it remains able to score multiple plausible explanations highly. Table 5 presents two examples in which both explanations are plausible. The RoBERTa model trained with plausibility annotations produces posterior distributions that collapse to one explanation. However, LiPoR can assign significant probability to both explanations.

Qualitative Comparison Table 6 presents a number of examples accompanied with the predictions made by fine-tuning via max-marginal likelihood (-PR) and LiPoR (+PR) side by side. The two examples on the top are among the more difficult abduction examples: the first example requires a model to draw a connection between abstract concepts and concrete objects ("what you love" → "taking long warm showers"); the second example requires a model to figure out an inclusion relation (Nepal is a country in Asia). We italicize the words that co-occur across x, z and y, and we speculate that fine-tuning chooses the wrong explanations because of lexical overlap shortcuts. LiPoR, however, was able to correctly flip these predictions with

	αNLI	Sen-Making	δ-ATOMIC	δ-SNLI	δ-SOCIAL	Winowhy
ZS GPT3	**67.54**	43.00	50.73	49.69	49.22	50.99
Prompted GPT3	49.19	53.80	48.23	51.26	50.86	58.10
LiPoR	57.50	**61.50**	**67.60**	**64.40**	**55.40**	**58.67**
RoBERTa (Y)	53.71	61.30	62.74	57.81	51.78	42.13

Table 3: Comparing LiPoR to several baselines on automatically generated explanation candidate sets. (Y) indicates that a method uses plausibility annotations.

	Tuned ✓	Tuned ✗
ZS ✓	1140	419
ZS ✗	618	882

	LiPoR ✓	LiPoR ✗
Tuned ✓	1449	309
Tuned ✗	767	534

Table 4: **Left:** Comparison between ZS BART and Tuned BART on αNLI. **Right:** Comparison between Tuned BART and LiPoR. {*} ✓ and {*} ✗ denote the number of instances for which plausible explanations are correctly / incorrectly identified by {*}, respectively.

	Example	Y	N
x:	Sally went to Italy in the spring.		
z:	Sally took a lot of pictures when she went sightseeing.	71.7	50.0
	Sally took pictures at every place she visited.	28.3	50.0
y:	When she got home, Sally showed her pictures to all her friends.		
x:	Mike didn't study for a test.		
z:	Mike was normally a good student.	100	50.0
	Everyone in class failed the test except for Mike.	0	50.0
y:	The teacher was very disappointed.		
?	LiPoR assigns close probabilities to the indistinguishably likely explanations, while the supervised model collapses to one of the explanations.		

Table 5: Comparison between posterior probabilities for each explanation produced by a RoBERTa model trained with plausibility annotations (Y) and LiPoR (N) on individual test examples, respectively.

	Example	-PR	+PR
x:	I love taking long warm *showers*.		
z:	*Showers* make me sleepy.	50.3	6.0
	Doing what you love is important.	49.7	94.0
y:	That's why I take two of them every day.		
x:	Neil wanted to see the *mountains* of Asia.		
z:	**Neil booked a tripped online.**	47.5	64.0
	Neil took a trip to see the Rocky *mountains* instead.	52.5	36.0
y:	Neil loved being so close to the *mountains* in Nepal!		
👍	Fine-tuning (-PR) looks at superficial word co-occurrences, but LiPoR (+PR) tries to understand the true context.		

	Example	-PR	+PR
x:	We went to the park today.		
z:	We were rained on!	53.5	3.9
	We met a golden retriever puppy and he played with us.	46.5	96.1
y:	I love going to the park!		
x:	Before my lunch time I got a phone call.		
z:	My best friend wanted to go on a trip.	50.5	40.9
	My best friend wanted to try a new restaurant for lunch.	49.5	59.1
y:	We had an amazing time!		
👍	LiPoR (+PR) is able to correct the bias towards shorter explanations.		

Table 6: Comparison between posterior probabilities for each explanation produced by fine-tuning (-PR) and LiPoR (+PR) on individual test examples, respectively. The two tables consist of examples where LiPoR successfully corrects the mistakes made by fine-tuning. The plausible explanation labeled by human annotators are in boldface.

high confidence.

The two examples on the bottom are those for which Tuned BART fails to identify the plausible explanation because one explanation is short and the other is long. Again, LiPoR is able to correct these mistakes. Furthermore, the probability produced by LiPoR for each explanation also reflects the model's confidence to a certain degree. In the first example, "we met a golden retriever puppy and he played with us" is a much better explanation than "we were rained on," because one does not need to go to a park to experience rain. As a result, the difference between probabilities for the two explanations is 92.2%. For the second example, "we had an amazing time" could refer to both trying out a new restaurant and going on a trip. The phone call was received before lunch time makes the second explanation more likely, but the first explanation can still be what actually happened. As a result, LiPoR assigns 40.9% to the "trip" explanation and 59.1% to the "restaurant" explanation, leading to a smaller gap than that of the first example.

8 Conclusion

We introduce LiPoR, which fine-tunes pretrained language models on abductive reasoning tasks without plausibility annotations. Results shows that LiPoR achieves comparable performance to that of knowledge-augmented zero-shot methods.

Ethical Statement

LiPoR shares similar concerns with other contemporary approaches for performing commonsense reasoning. Specifically, because LiPoR exploits the knowledge already present in pretrained language models, it can potentially reinforce existing harmful biases in such models.

Acknowledgement

AR and JC are supported by a Sloan Fellowship, NSF CAREER #2037519, and NSF #1901030. CC and WZ are supported by NSF #1815455.

References

Henning Andersen. 1973. Abductive and deductive change. *Language*, pages 765–793.

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2021. Conversational neuro-symbolic commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4902–4911.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162.

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*.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6266–6278.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Eugene Charniak and Solomon E Shimony. 1990. Probabilistic semantics for cost based abduction. In *Proceedings of the eighth National conference on Artificial intelligence-Volume 1*, pages 106–111.

Di Chen, Yiwei Bai, Wenting Zhao, Sebastian Ament, John Gregoire, and Carla Gomes. 2020. Deep reasoning networks for unsupervised pattern de-mixing with constraint reasoning. In *International Conference on Machine Learning*, pages 1500–1509. PMLR.

Zi-Yi Dou and Nanyun Peng. 2022. Zero-shot commonsense question answering with cloze translation and consistency optimization. In *The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the winograd schema. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10486–10500.

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.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. 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 1161–1166.

Andrew S. Gordon and Jerry R. Hobbs. 2017. *Explanation*, page 299–305. Cambridge University Press.

Canming Huang, Weinan He, and Yongmei Liu. 2021. Improving unsupervised commonsense reasoning using knowledge-enabled natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4875–4885.

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.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. 2020. Unsupervised text generation by learning from search. *Advances in Neural Information Processing Systems*, 33:10820–10831.

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*.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In *35th AAAI Conference on Artificial Intelligence*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

Gabriele Paul. 1993. Approaches to abductive reasoning: an overview. *Artificial intelligence review*, 7(2):109–152.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 794–805.

Rajat Raina, Andrew Y Ng, and Christopher D Manning. 2005. Robust textual inference via learning and abductive reasoning. In *AAAI*, pages 1099–1105.

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.

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.

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

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4615–4629.

Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634.

Paul Thagard and Cameron Shelley. 1997. Abductive reasoning: Logic, visual thinking, and coherence. In *Logic and scientific methods*, pages 413–427. Springer.

Keyon Vafa, Yuntian Deng, David Blei, and Alexander M Rush. 2021. Rationales for sequential predictions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10314–10332.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

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.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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.

Qian Zhao, Siyu Tao, Jie Zhou, Linlin Wang, Xin Lin, and Liang He. 2020. ECNU-SenseMaker at SemEval-2020 task 4: Leveraging heterogeneous knowledge resources for commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 401–410, Barcelona (online). International Committee for Computational Linguistics.

Pei Zhou, Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Jay Pujara, Xiang Ren, Yang Liu, and Dilek Hakkani-Tur. 2021a. Think before you speak: Learning to generate implicit knowledge for response generation by self-talk. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 251–253, Online. Association for Computational Linguistics.

Pei Zhou, Pegah Jandaghi, Hyundong Cho, Bill Yuchen Lin, Jay Pujara, and Xiang Ren. 2021b. Probing commonsense explanation in dialogue response generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4132–4146, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. *Advances in Neural Information Processing Systems*, 33:6803–6814.

A Additional Experiments

How do models with different architectures and sizes perform at abductive reasoning? Table 7 summarizes the results on the αNLI dataset with different model architectures and model sizes, which are obtained from the same grid search described in Sec. 5. Within the same architecture, models with more parameters are better at abductive reasoning. When comparing between BART and T5, BART can produce consistent better results at each size.

Does a learnable $p(z|x)$ model lead to better performance? Here we test if a learnable $p(z|x)$ model instead of a uniform $p(z|x)$ model leads to better performance. We should note that a learnable $p(z|x)$ model may result in reasoning shortcuts: because if the signal from $p(z|x)$ is too strong, then this term will dominate Eq. 2; thus, $p(z|x, y)$ computed in this way is no longer a result of thinking backwards. We parametrize the learnable $p(z|x)$ model by a BART-large model, which takes x as an input and returns a probability distribution over all sequences. Table 8 shows the comparison between the two $p(z|x)$ models on the αNLI dataset. Although the uniform $p(z|x)$ model outperforms the learnable $p(z|x)$ model, the difference between them is not significant.

How do methods without plausibility annotations perform in presence of distractors? In order to test the robustness of different methods without plausibility annotations, we evaluate them on two types of distractors added to the αNLI test set. The first type of distractor randomly samples a third explanation from another example, and the second type of distractor constructs a third explanation with randomly sampled words from the vocabulary of the αNLI dataset with a length that falls in-between the lengths of the two original explanations. Table 8 compares the results with and without the distractors. Notice that after adding a third option, the chance of getting the plausible explanation with a random guess is $\frac{1}{3}$. LiPoR's accuracy drops significantly with the presence of distrators, while the relative decrease for GPT NEO is smaller. Furthermore, the zero-shot results (i.e., ZS and GPT NEO) suggest that it is more difficult to identify the first type of distractor than the second one. Our interpretation for a worse performing LiPoR's on distractors is that the distrators break our assumption: $p(z|x)$ is no longer uniform, and the probability of a distracting explanation is independent of the probability of x. Therefore, the original factorization in Eq. 1 no longer applies. To build an unsupervised system that is robust to distractors requires incorporating the new assumptions in the data generating process.

	BART	T5
small	-	54.14
base	60.08	57.31
large	71.56	65.48

Table 7: Comparison between different model architectures and model sizes on the αNLI dataset.

	Original	+Rand. E's	+Rand. W's
GPT NEO	57.47	51.12	57.37
ZS	50.96	34.39	38.22
LL	57.40	53.48	53.52
LiPoR w/ unif. $p(z|x)$	**71.56**	58.58	57.40
LiPoR w/ learned $p(z|x)$	69.92	**59.14**	**59.24**

Table 8: Comparison between different unsupervised approaches on the αNLI test set. +Rand. E's is adding a random explanation taken from another example. +Rand. W's is adding random words from the vocabulary of αNLI whose length is between the lengths of two original explanations. Best results for each setting is in boldface.

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

□ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- □ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
  Left blank.

- □ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
  Left blank.

- □ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
  Left blank.

D □ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**

Left blank.

- □ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
  Left blank.

- □ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
  Left blank.

- □ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
  Left blank.

- □ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
  Left blank.

- □ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
  Left blank.