Unsupervised Explanation Generation via Correct Instantiations

Sijie Cheng[1,2]*, Zhiyong Wu[1]†, Jiangjie Chen[2], Zhixing Li[3], Yang Liu[5,6], Lingpeng Kong[1,4]

[1]Shanghai Artificial Intelligence Laboratory
[2]Fudan University
[3]Full Truck Alliance
[4]The University of Hong Kong
[5]Institute for AI Industry Research, Tsinghua University
[6]Department of Computer Science and Technology, Tsinghua University

Abstract

While large pre-trained language models (PLM) have shown their great skills at solving discriminative tasks, a significant gap remains when compared with humans for explanation-related tasks. Among them, explaining the reason why a statement is wrong (e.g., against commonsense) is incredibly challenging. The major difficulty is finding the conflict point, where the statement contradicts our real world. This paper proposes NEON, a two-phrase, unsupervised explanation generation framework. NEON first generates corrected instantiations of the statement (*phase I*), then uses them to prompt large PLMs to find the conflict point and complete the explanation (*phase II*). We conduct extensive experiments on two standard explanation benchmarks, i.e., ComVE and e-SNLI. According to both automatic and human evaluations, NEON outperforms baselines, even for those with human-annotated instantiations. In addition to explaining a negative prediction, we further demonstrate that NEON remains effective when generalizing to different scenarios. The resources of NEON are available at: https://github.com/Shark-NLP/Neon.

1 Introduction

Nowadays, Explainable Natural Language Processing (ExNLP) (Danilevsky et al. 2020) has received increasing attention toward trustworthy NLP models. A valid explanation can not only ensure that a model solves a problem using the corresponding knowledge rather than exploiting superficial cues or short-cuts (Niven and Kao 2019; Geva, Goldberg, and Berant 2019; Cui et al. 2021), but they can also be used to improve model performance on downstream tasks (Wei et al. 2022; Wang et al. 2022; Creswell, Shanahan, and Higgins 2022).

In general, there are two main types of explanations in the field of ExNLP: highlights and free-text explanations (Wiegreffe and Marasovic 2021). Highlights (Lei, Barzilay, and Jaakkola 2016) methods use subsets of the input to support model prediction, thus can not solve the majority of tasks where input does not contain rationales. In this paper, we focus on free-text explanation, which justifies model predictions using natural language. Despite being more flexible and human-readable, free-text explanations pose great challenges (Lin et al. 2020; Rajani et al. 2019), as they require models' ability to accurately understand the logic behind the problem and express them in natural language.

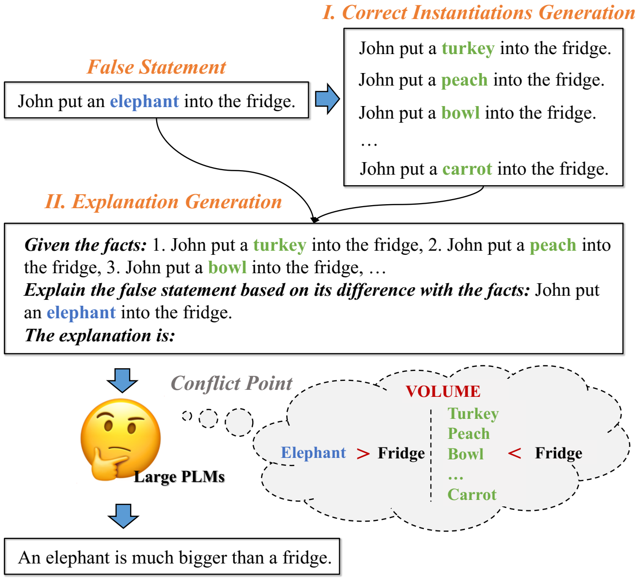
I. Correct Instantiations Generation
False Statement
John put an elephant into the fridge.
John put a turkey into the fridge.
John put a peach into the fridge.
John put a bowl into the fridge.
…
John put a carrot into the fridge.
II. Explanation Generation
Given the facts: 1. John put a turkey into the fridge, 2. John put a peach into the fridge, 3. John put a bowl into the fridge, …
Explain the false statement based on its difference with the facts: John put an elephant into the fridge.
The explanation is:
Conflict Point
Large PLMs
VOLUME
Elephant > Fridge
Turkey
Peach
Bowl
…
Carrot
< Fridge
An elephant is much bigger than a fridge.

Figure 1: Our proposed two-phase framework NEON (*correct instantiations generation* and *explanation generation*) explains a false statement in ComVE (Wang et al. 2020) task. The *conflict point* module is implicitly induced inside the large pre-trained language models.

In this paper, we propose a model agnostic framework to generate free-text explanations for *false statements*. Given a false statement which is against commonsense, "*John put an elephant into the fridge*", the model is expected to generate a convincing explanation "*an elephant is much bigger than a fridge*" to state the reason why the former statement is incorrect (ComVE; Wang et al. 2020). Explaining a false state-

[*] This work was done during her internship at Shanghai AI Laboratory. E-mail: sjcheng20@fudan.edu.cn.

[†] Corresponding author. E-mail: wuzhiyong@pjlab.org.cn.

False Statement	**Explanation**	**Conflict Point**
John put an elephant into the fridge.	An elephant is much bigger than a fridge.	Volume
He drinks apple.	Apple can not be drunk.	Function
Jeff ran 100,000 miles today.	No way can someone run 100,000 miles in a day.	Speed
A giraffe is a person.	A giraffe is an animal, not human.	Property
Europe is in France.	Europe is a continent but france is a country.	Geography

Table 1: Examples and their exact conflict points to explain in ComVE task.

ment is generally considered to be more challenging (Wang et al. 2019), but it is the key to preventing models from making mistakes and improving their performances (Hoffmann and Magazzeni 2019; Lipton 1990).[1]

Recent studies (Jon et al. 2020; Konar et al. 2020) generally adopt sequence-to-sequence or language model generation approaches. The sequence-to-sequence (seq2seq) methods (Jon et al. 2020; Wan and Huang 2020) use the false statement as the source sequence to generate reason as the target sequence. As for the LM approaches (Konar et al. 2020; Fadel, Al-Ayyoub, and Cambria 2020), they manually design prompts for auto-regressive language models to generate explanations. We argue that both methods neglect that the key to solving such problems is to find the **conflict point** as shown in Table 1, where the false statement contradicts the commonsense knowledge. For instance, the conflict point of "*John put an elephant into the fridge*" is the relative volume between "*elephant*" and "*fridge*".

Finding the exact conflict point can be rather difficult, even for large PLMs. On the one hand, manually constructing a dataset with conflict points for training is labor-intensive and difficult to scale (Wang et al. 2020). On the other hand, exact triples of conflict points are rare in the external knowledge graph due to their tacitness and diversity. (Wan and Huang 2020; Konar et al. 2020). Considering the limitations of these direct methods mentioned above, we try to provide guided hints as prompts to implicitly elicit PLMs to reason the conflict point, inspired by the line of work about the chain of thought (Wei et al. 2022; Creswell, Shanahan, and Higgins 2022; Wang et al. 2022). To produce guided hints, we automatically generate a bunch of correct instantiations based on the false statement. Then, the conflict points can be implicitly induced from the difference between the commonality of our generated instantiations and the false statement. For example, given the false statement "*John put an elephant into the fridge*", we firstly generate a set of correct instantiations {"*John put a* ***turkey*** *into the fridge*", "*John put a* ***peach*** *into the fridge*", $\cdots$} and their underlying commonality is that their volumes are all smaller than the fridge. Combining these instantiations and the false statement, their difference can help PLMs better implicitly reason that the conflict point is the relative volume where an elephant is much bigger than a fridge.

In this paper, we propose NEON, a two-phase framework for unsupervised explanation generation via correct instantiations as shown in Figure 1. In the first phase, given the false statement, we attempt both in-context and unsupervised learning to generate correct instantiations automatically. In the second phase, combining both generated instantiations and the false statement, the PLMs can implicitly induce the conflict point better to generate explanations. To purely detect the ability of implicit induction in off-the-shelf PLMs, we explore the model performance in the unsupervised setting rather than the traditional supervised setup. We conduct extensive experiments on two standard explanation benchmarks, ComVE (Wang et al. 2020) and e-SNLI (Camburu et al. 2018). Experimental results prove the effectiveness of our method on both automatic and manual evaluations. Furthermore, we also conduct analysis experiments to demonstrate that the main idea of NEON can generally be extended to accommodate other explanation tasks.

The contributions of our work are as follows:

- We propose a novel method based on the importance of conflict points to solve the false statement explanation problem. To the best of our knowledge, we are the first to introduce the concept of the conflict point in the task.
- We propose a two-phase framework named NEON to elicit the large PLMs to induce through instantiations to unsupervised explanation generation.
- We present analyses of our generated instantiations and demonstrate the generality of NEON.

2 Methodology

2.1 Problem Formulation

Our target problem is to generate a reason to explain why the false statement does not make sense or is inconsistent. Given the original false statement with n tokens $\boldsymbol{x} = \{x^1, x^2, \cdots, x^n\}$, we automatically generate a set of correct instantiations $\mathbb{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_l\}$ with a commonality. Each instantiation $\boldsymbol{h} = \{h^1, h^2, \cdots, h^k\}$ with k tokens is a constrained text generating conditioned on the original false statement $\boldsymbol{x}$. According to both the false statement $\boldsymbol{x}$ and our generated correct instantiations $\mathbb{H}$, the pretrained language model G needs to implicitly reason the conflict point and give a rational explanation with m tokens $\boldsymbol{y} = \{y^1, y^2, \cdots, y^m\}$.

[1] It is worth noting that we also explore explaining correct statements in Section 4.5 to demonstrate the generality of our method.

Phase I: Correct Instantiations Generation
Task: Based on the incorrect statement, generate the correct statement. /* Example 1 */ Incorrect statement: *He drinks apple*. Correct statement: ***He drinks milk.*** /* Test data */ Incorrect statement: *John put an elephant into the fridge.* Correct statement:
Phase II: Unsupervised Explanation Generation
Given the facts: ***1. John put a turkey into the fridge, 2. John put a peach into the fridge, 3. John put a bowl into the fridge,*** Explain the following statement based on its difference with the facts: *John put an elephant into the fridge*. The explanation is:

Table 2: The prompt instances of in-context learning in our two phases: presented are the *incorrect statements* and the *correct statements* (bold). We use 16 examples per prompt in the first phase.

2.2 Correct Instantiations Generation

In the first phase, we attempt two different means to generate correct instantiations $\mathbb{H}$ conditioned on the false statements $\boldsymbol{x}$ to prove the flexibility of our framework NEON. One adopts in-context learning with larger language models under the few-shot setting, the other one is based on traditional constrained text generation in an unsupervised way.

In-context Learning Considering the large number of parameters in PLMs, in-context learning uses a series of demonstrations of a specific task as prompts to induce model generation, while the parameters of large PLMs are fixed (Brown et al. 2020; Radford et al. 2019). Besides the advantage of no extra need to train or finetune, in-context learning can also reduce the reliance on large amounts of annotated data. Therefore, due to the great performance of in-context learning with large PLMs in the recent studies (Wiegreffe et al. 2021), we attempt to use in-context learning to generate correct instantiations $\mathbb{H}$ automatically given the original false statement $\boldsymbol{x}_{\text{ori}}$.

Following Wiegreffe et al. (2021), we apply the in-context learning under few-shot setups to generate our correct instantiations. We specifically design a prompt followed by the false statement that the model needs to correct. To construct the prompt, we first randomly sample 200 instances ([correct statement, incorrect statement] in the ComVE task and [entailment statement, contradiction statement] in the e-SNLI task) from the training dataset. Then we randomly select K instances and concatenate them to construct our prompt as shown at the top of Table 2. Finally, we feed the model with both constructed prompt and our test data to infer the completion which can be regarded as our generated correct instantiations.

Constrained Text Generation Despite its simplicity, in-context learning often requires human annotations, which is not always available. In this section, we explore the challenging setting where instantiations are generated in a fully unsupervised manner. As a preliminary study, we apply the widely used constrained text generation framework CGMH (Miao et al. 2019).

Firstly, we adopt perplexity (PPL) computed by the masked language models to detect the conflicting position. Given the false statement $\boldsymbol{x} = \{x^1, x^2, \cdots, x^n\}$, we compute the relative perplexity score of the original statement to its masking sentence which replaces x^i with `[MASK]`. Then we normalize scores and sample the edited position based on this distribution. If the i-th token is unlikely to exist in the position, the perplexity score S^i_{PPL} is larger, which indicates the token should be edited with a higher priority.

$$S^i_{\text{PPL}} = \frac{\text{PPL}(\boldsymbol{x})}{\text{PPL}(\boldsymbol{x} \backslash \{x^i\})} \tag{1}$$

Given the sampled positions, we need to determine each position's action. Our token-level actions mainly include three types following Chen et al. (2022): *insert*, *delete* and *replace*. As for the acceptance rate to each generated sentence $\boldsymbol{x}'$ with edited action, our considering property is **fluency** which is important to guarantee in generative tasks. We measure this fluency score via computing likelihood based on the auto-regression language models, e.g., GPT-2 (Radford et al. 2019).

$$S_{\text{Fluency}} = \prod_{i=1}^{n} P_{\text{LM}}(h^i | h^{<i}) \tag{2}$$

2.3 Unsupervised Explanation Generation

As demonstrated in recent studies (Zhong et al. 2022), PLMs can capture subtle yet critical differences between different groups of sentences. This inspires us that capturing the differences between the false statement $\boldsymbol{x}$ and our generated instantiations $\mathbb{H}$ can help PLMs induce conflict points. Therefore, in the second phase, given both the false statement $\boldsymbol{x}$ and our generated instantiations $\mathbb{H}$, we implicitly induce the large PLMs to generate the free-text explanation $\boldsymbol{y}$ in the zero-shot setting.

Zero-shot Learning We adopt a similar prompt construction strategy as discussed in the correct instantiations generation phase. However, unlike the template of few-shot learning in instruction style, our template of zero-shot learning is more fluency like a complete sentence, following previous studies (Sanh et al. 2021). In particular, we directly design a natural language description according to different tasks instead of sampled exemplars from training datasets as shown at the bottom of Table 2. Considering the variance due to different descriptions, more analysis on the design of prompting can be found in Section 4.2 and 4.4. Finally, based on our constructed prompt, the PLMs generate the completion as our generated explanation.

Row	Method	ComVE				e-SNLI			
		BLEU	**ROUGE**	**BERTScore**	**S-BERT**	**BLEU**	**ROUGE**	**BERTScore**	**S-BERT**
1	**Random**	1.47	17.81	46.21	42.54	4.94	24.23	50.73	43.05
2	**Retrieval-BM25**	1.51	17.23	45.18	38.68	4.29	23.31	49.80	42.09
3	**Retrieval-SBERT**	1.69	18.55	46.64	45.47	4.64	24.45	51.16	48.22
4	**Original**	1.88	20.21	48.68	51.82	4.71	25.38	50.92	46.39
5	**Human-annotated**	2.48	21.25	49.66	**55.21**	5.57	25.62	51.96	49.19
6	**Top-1**	2.42	21.42	49.86	55.03	6.03	25.87	51.97	48.51
7	**NEON w/ CGMH**	3.37	20.10	48.92	49.50	4.67	26.04	51.04	48.42
8	**NEON w/ In-context**	**3.39**	**22.50**	**51.50**	54.52	**6.20**	**27.28**	**53.87**	**51.69**

Table 3: The automatic evaluation results of ComVE and e-SNLI tasks.

3 Experiments

3.1 Experimental Setups

Datasets Our experiments are conducted on the two important explanation benchmarks, ComVE (Wang et al. 2020) and e-SNLI (Camburu et al. 2018). The ComVE task asks annotators to create 11,997 instances in the format of $\{c_n, s_n, r_{n_1}, r_{n_2}, r_{n_3}\}$, where c_n and s_n are the correct and incorrect statement, respectively. $\{r_1, r_2, r_3\}$ are three reference reasons to explain the incorrect statement. Then they divide all these annotated instances into train/dev/test datasets with 10,000/997/1,000 instances. As for the e-SNLI task, the c_n and s_n can be seen as entailment and contradiction statements, respectively. Filtering the odd instances with only entailment or contradiction statement, our obtained train/dev/test is 5,189/3,280/2,640.

Models In our main experiments, We all adopt the large pre-trained language model OPT-175B (Zhang et al. 2022). To ensure the generalization of our framework, we also conduct experiments on other PLMs varying from small model scale to large. More details can be found in Section 4.3.

Implementation Details In the first phase, to fix the max-length of the context window ($n_{ctx} = 2048$), we set the number of examples as $K = 16$. Moreover, the max length of generated instantiations is 25 for ComVE and 40 for e-SNLI. As for constrained text generation, we adopt GPT-2-large and RoBERTa-large. In the second phase, the max length of generated explanations is 30 for both tasks. The hyper-parameter of Top-p is 0.9, and the temperature is 0 for all generation models. We repeat the same experiment three times and report the average accuracy for all experiments. Our experiments are conducted with 8 A100 GPUs.

Baselines We compare our framework with the following baselines. (1) **Original**: the model only feeds with the false (contradiction) statement to generate its rational explanation. (2) **Random**: We feed a randomly sampled human-annotated correct (entailment) statement and the false statement into the model. (3) **Retrieval**: we adopt both BM25 (Robertson, Zaragoza et al. 2009) and Sentence-BERT (SBERT) (Reimers and Gurevych 2019) to retrieve the five nearest statements from the Open Mind Common Sense (OMCS) corpus (Singh et al. 2002), then give them and the false statement into the model. (4) **Human-annotated**: we offer both the false statement and its corresponding human-annotated correct statement to the model. It is worth noting that the human-annotated statement can be regarded as the upper bound of our generated single instantiation. (5) **Top-1**: the model generates explanations based on the false statement and our Top-1 generated correct instantiation. To ensure fairness, we keep the templates of all these baselines (except for the original baseline) the same as ours.

3.2 Evaluation Metrics

Automatic Evaluation Metrics Considering that the official automatic evaluation metric BLEU (Papineni et al. 2002) is too harsh to evaluate the quality of explanations (Zhao et al. 2019; Konar et al. 2020; Fadel, Al-Ayyoub, and Cambria 2020), we further involve a set of common evaluation metrics as supplementary following Becker, Liang, and Frank (2021). In detail, we measure diverse aspects, including token overlap using BLEU and ROUGE (Lin 2004), semantic similarity using both BERTScore (Zhang et al. 2019) and S-BERT (Reimers and Gurevych 2019).

Manual Evaluation Metrics Due to the limitation of existing automatic metrics in the open-ended text generation community (Zhang et al. 2019; Novikova et al. 2017), we further conduct manual evaluations to compensate following Wiegreffe et al. (2021). Firstly, we randomly select 100 samples from the test set. We then evaluate generated explanations through head-to-head comparisons. In order to directly reflect the impact of our instantiations, three annotators are asked to choose the better explanation between the original baseline and NEON. To ensure fairness, we shuffle all the generated explanations to be unordered. We specifically design two aspects: one is the *preferred explanation* from the comprehensive consideration, and the other one needs to express the *conflict points* explicitly.

3.3 Results

Automatic Evaluation Table 3 shows the automatic evaluation results of NEON and baselines. To illustrate the effectiveness of introducing instantiations, we first compare the Original baseline against others (Row 5-8). As we can see, incorporating instantiations in explanation generation consistently improves model performance over the baseline without instantiations. Given the necessity of instantiations,

Dataset	Preferred Explanation (%)			κ
	Original	Tie	NEON	
ComVE	20.33	42.67	37.00	0.47
e-SNLI	18.67	41.67	39.67	0.39
	Conflict Point (%)			
ComVE	19.33	46.00	34.67	0.45
e-SNLI	15.67	53.67	30.67	0.36

Table 4: Head-to-head human evaluation for 100 explanations generated by the original baseline and NEON. Results are shown as % preferences with Fleiss Kappa κ.

we further investigate how the quality of instantiations affects performance. We observe significant performance deterioration when equipping the model with instantiations that come from random knowledge (Row 1) or strong retrieval models (Row 2-3). This indicates that introducing irrelevant or inaccurate information would hurt model performance. The above comparison also verifies the effectiveness of our instantiation generation method, which is further supported by the comparable performance between the Top-1 and Human-annotated baseline. Finally, by comparing NEON with Top-1 and Human-annotated, we find that ensemble multiple instantiations that share a commonality significantly outperforms baselines on almost all metrics in both tasks. We hypothesize that ensembling similar instantiations can help the model better locate the conflict points. This hypothesis is later supported by manual evaluation and nuanced analysis of instantiations (Section 4.1). Despite the excellent performance of NEON under the in-context setting, we find it barely outperforms the Original baseline when editing is performed in a fully unsupervised manner (Row 7). The reason could be that we use relatively small PLMs in CGMH due to computation constraints, whereas we use OPT-175B for in-context editing. We leave it as future work to investigate how to apply CGMH on huge PLMs for instantiation generation. Given the performance gap, all later analyses will be based on NEON w/ In-context.

Manual Evaluation As shown in Table 4, for both ComVE and e-SNLI tasks, NEON outperforms the original baseline with respect to *preferred explanation* and *conflict point*. The proportional distribution of the preferred explanation is similar to the conflict point, which supports our claim that it is important to find the conflict point to generate good explanations. In the conflict point aspect, the fact that NEON beat the original baseline reflects the contribution of our generated instantiations. It is worth noticing that there still remains a significant proportion of ties (40%). We believe a better method of finding conflict points can contribute to closing this gap.

4 Analysis

4.1 Quality of Generated Instantiations

Automatic Evaluation To check the correctness of generated instantiations, we fine-tune RoBERTa-Large (Liu et al. 2019) on both training datasets as binary classifiers. It achieves the accuracy of 88.97 and 84.25 on the ComVE and e-SNLI, respectively. We use these fine-tuned RoBERTa models to evaluate the quality of our generated instantiations. Because the performance of in-context learning is much better than CGMH in our first phase. We conduct experiments mainly on in-context learning in our analyses.

Dataset	Acc.	Gram.	Fact.	Diver.	Common.
ComVE	72.80	2.97	2.66	2.63	2.56
e-SNLI	81.67	2.88	2.72	2.89	2.66

Table 5: The manual evaluation results of our generated instantiations. (i. Acceptability; ii. Grammaticality; iii. Factuality; iv. Diversity; v. Commonality)

Method	BLEU	ROUGE	BERTScore	S-BERT
Top-1	**2.47**	20.77	49.13	54.25
Top-1*	2.20	**21.39**	**49.63**	**54.98**
NEON	3.39	21.65	49.09	53.11
NEON*	**3.51**	**22.32**	**49.54**	**54.53**

Table 6: The comparison of our methods before (marked *) and after filtering the low-quality instantiations.

Manual Evaluation Following previous studies (Wiegreffe et al. 2021), we assume that the desired instantiations need to meet the requirements at least in terms of both surface and explanation levels. Therefore, we further conduct manual evaluations in five primary criteria: i. *Acceptability* - Generated instantiations are acceptable in overall judgment; ii. *Grammaticality* - Generated instantiations should be at least fluent with no grammatical mistakes; iii. *Factuality* - Generated instantiations should be factually correct; iv. *Diversity* - We expect to generate more diverse instantiations; v. *Commonality* - Generated instantiations are expected to have a commonality to help large PLMs infer the conflict point. We randomly select 100 samples from the test set and their corresponding generated instantiations. Then, after shuffling all selected samples, three annotators are asked to choose acceptable/unacceptable for the acceptability metric and use a 3-point Likert-scale rating to evaluate sampled data for the other four aspects.

Results We evaluate the quality of the automatically generated and human-generated instantiations, they reached the accuracy of 70.28/89.60 and 92.30/97.84, respectively. Note that in-context learning only uses a few exemplars in the prompts. As shown in Table 5, the human acceptance of the generated instantiations is 72.80/81.67, consistent with the results of the automatic evaluation discussed above. As for the surface-level criteria, the score of grammaticality is pretty high, while the score of factuality is relatively worse. The results of the diversity and commonality metrics are over 2.5 points, indicating that the instantiations have a high diversity while sharing a common underlying property well.

Furthermore, we filter the low-quality instantiations de-

#	BLEU	ROUGE	BERTScore	S-BERT
1	2.42	21.03	49.22	52.70
2	2.61	21.14	49.22	52.56
3	3.32	21.32	49.46	51.79
4	3.29	22.26	50.97	**54.74**
5	3.39	**22.50**	**51.50**	54.52
6	3.01	21.49	49.11	49.06
7	**3.48**	21.57	49.45	49.66
8	3.28	21.27	49.66	49.94
9	3.16	21.70	49.91	48.73
10	3.39	21.21	49.94	49.47

Table 7: Model performance with the different number of ensemble instantiations in the ComVE task.

termined by the automatic metric to probe the correlation between the quality and model performance. Taking the ComVE task as an example, we first generate 10 instantiations per data for 1,000 test data and filter low-quality instantiations. We then obtain 987 and 773 samples with Top-1 and ensemble (five) high-quality instantiations, respectively. Finally, we compare the model performance before and after filtering. The results are shown in Table 6. As for the Top-1 instantiation, the results before and after filtering are comparable which supports the similar situation between our Human-annotated and Top-1 baselines reported in Table 3. Our ensemble method shows a general improvement in model performance after filtering. Combining the above phenomena, we believe that the ensemble method is more stringent about the quality of generated instantiations due to asking for commonality among them.

4.2 Effects on Instantiations Number

Despite the fact that the PLMs can implicitly induce the conflict points through instantiations, it still remains a question of whether more generated instantiations lead to better performance. Therefore, we detect the model performance with the different numbers of generated instantiations varying from 1 to 10. The results are shown in Table 7.[2] When the number of instantiations increases from 1 to 5, the model performance exhibits an upward trend. This phenomenon indicates that the increasing diversity of generated instantiations decreases the possibility of other misleading conflict points. However, as the number of instantiations increases from 6 to 10, the model performance plateaus. We conjecture there are two-fold reasons. One is that sufficient diversity and more noise will limit the improvement of model performances when the number reaches a certain level. The other one is that prompts containing overlong and unoriginal sequences will damage the performance.

4.3 Effects on Model Size

In this section, we detect the model performance to generate explanations with increasing model scales. As shown in Figure 2, the experimental results are similar in most of these models, except for the smallest model GPT2-M. This phenomenon indicates that only offering an extra instantiation will be regarded as noise to hurt model performances when the model parameter is relatively small. However, NEON with ensemble instantiations obviously beat all baselines with different model scales, reflecting its robustness. Moreover, as the scale of model parameters increases, the performance gap between NEON and the baselines becomes larger. This trend shows that the implicit induction through instantiations of large PLMs is an emerging ability with increasing model scales, which is consistent with previous studies (Wei et al. 2022; Wang et al. 2022).

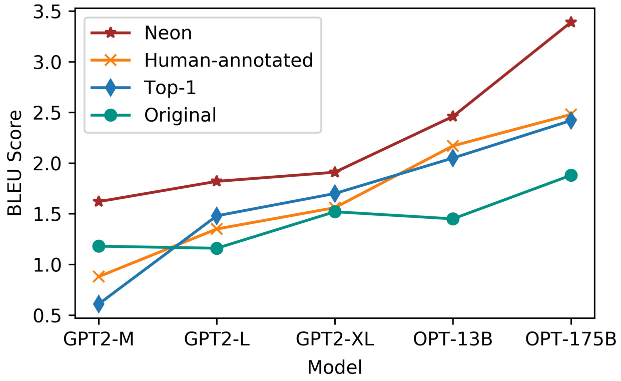

Figure 2: Model performance of increasing model scales in the ComVE task.

4.4 Robustness of Prompting

According to previous studies (Zhao et al. 2021; Lu et al. 2021), the model performances are sensitive to templates. Therefore, we further evaluate the robustness of NEON following Wei et al. (2022). Another two annotators B and C are asked to write the templates independently. Furthermore, we ask annotator A to write an instruction-style template that is more concise, following Cobbe et al. (2021). Results shown in Figure 3 indicate that though there exists a variance among different annotated templates, all our prompts still outperform the original baseline. However, the instruction style prompt is significantly worse than the natural language description style in the zero-shot setting, due to the lack of instruction style signals in the pre-training corpus.

4.5 Demonstration of Generality

In this section, we adapt NEON to generate explanations for correct statements. Taking the e-SNLI task as an example, given the entailment statement c_n, there are three ground-truth explanations $\{r'_1, r'_2, r'_3\}$. We directly use the generated correct instantiations in the first phase as guided hints. We find that their commonality with the entailment statement can help PLMs to explain as shown in Table 8. Notably, the Top-1 baseline is slightly worse than the original baseline. This phenomenon suggests that a single instantiation no longer provides valid information like the contrast

[2] To keep the templates consistent, we separate each instance by an ordinal number, including only one instantiation.

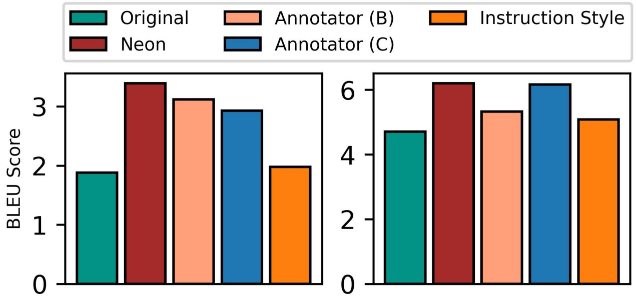

Figure 3: Independently-written for the robustness of NEON.

Method	BLEU	ROUGE	BERTScore	S-BERT
Original	8.11	29.73	52.66	53.18
Top-1	9.22	28.64	52.64	50.81
NEON	**11.18**	**31.69**	**55.30**	**56.33**

Table 8: Model performance of generating explanations for correct statements in the e-SNLI task.

in false statements during the explanation of correct statements. However, NEON still significantly outperforms baselines, which demonstrates the effectiveness and generality.

4.6 Case Study

Finally, we present some generated examples of two phases as shown in Table 9. In the first phase, the quality of our generated instantiations is comparable to ground-truth instantiations. Especially, the keywords of “safety”, “shelter” and “peace” meet the needs of both diversity and commonality. As for the second phase, NEON and other baselines yield fluent explanations. However, the original baseline does not give a correct definition of *home*, and the Human-annotated baseline pays much attention to the keyword “security” so that its generated explanation is irrelevant. Overall, NEON products a better explanation thanks to the commonality induced from the generated instantiations in the first phase.

5 Related Work

Natural Language Explanations To improve the transparency and reliability of large deep learning language models, a promising approach is to ask models to generate natural language free-form explanations. This explanation technique is first introduced in computer vision applications (Park et al. 2018; Hendricks et al. 2016). Then it broadcasts into diverse domains of natural language processing. For instance, Camburu et al. (2018) annotate natural language explanations of the entailment relations in the Stanford Natural Language Inference dataset. Rajani et al. (2019); Wang et al. (2020); Aggarwal et al. (2021) extends this idea into commonsense explanations. To solve these explanation tasks, traditional methods (Jon et al. 2020; Konar et al. 2020) usually finetune generation models with explicit human supervision. Wan and Huang (2020); Konar et al. (2020) exploit external knowledge graph to fill in the needed background in the explanation. Most relevant to our study is the contrastive explanation (Paranjape et al. 2021; Ross, Marasović, and Peters 2020), which gives a contrastive explanation to answer “Why P rather than Q”. However, they often focus on only one fact. In contrast, we notice that ensemble instantiations with a commonality can help find the exact conflict point.

Phase I: Correct Instantiations Generation

False Statement: A home is a place for violence.
NEON: 1. A home is a place for safety. 2. A home is a place for love. 3. A home is where you live. 4. A home is a place for shelter. 5. A home is a place of peace.
Ground-truth: A home gives a person a sense of security.

Phase II: Unsupervised Explanation Generation

Original: That’s the wrong definition of home. A place isn’t a home, you are home.
Human-annotated: People commit suicide and violence when there is no security.
NEON: A home is a place for peace, then it is not a place for violence.
Ground-truth: Safety and Security place is our home.

Table 9: Case study of the ComVE task.

In-context Learning After the fine-tuning paradigm of large PLMs, in-context learning has been attractive due to its simple operation and strong interaction. More important, it does not have to update the model parameters anymore. Brown et al. (2020) propose that large PLMs can complete a generation given a few demonstrations as prompts. Recently, more studies have paid attention to generating rationales through in-context learning to help language model performance. Wei et al. (2022) adopt “chain-of-thought” reasoning prompt to induce large PLMs reason step-by-step. Similarly, Wang et al. (2022) explore that ensemble rationales can significantly improve the model performance. Zelikman, Wu, and Goodman (2022) propose a bootstrapping strategy to improve the quality of rationale examples.

6 Conclusion and Future Work

In this paper, we propose a two-phase framework NEON to help large PLMs generate explanations by implicitly identifying conflict points in the statement. In the first phase, we generate a bunch of correct instantiations with a commonality based on the false statement. In the second phase, given both generated correct instantiations and the false statement, we adopt prompts to generate explanations according to their differences. Experiments in the unsupervised setting show that our proposed framework significantly outperforms baselines in both automatic and human evaluations. Furthermore, our analysis shows the effectiveness, robustness, and generality of NEON. We regard NEON as a first attempt towards using methods based on conflict points, which we argue is an important factor in solving the explanation tasks. Future work could focus on incorporating conflict points into the textual representations, e.g. through contrastive learning.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful and constructive feedback. We appreciate Tao Yu for the resource of OPT-175B; Jiacheng Ye and Lei Li for their valuable discussions. This work is partially supported by the Shanghai Committee of Science and Technology (Grant No. 21DZ1100100), and National Natural Science Foundation of China (No. 61925601).

References

Aggarwal, S.; Mandowara, D.; Agrawal, V.; Khandelwal, D.; Singla, P.; and Garg, D. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Workshop on Commonsense Reasoning and Knowledge Bases*.

Becker, M.; Liang, S.; and Frank, A. 2021. Reconstructing implicit knowledge with language models. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 11–24.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Camburu, O.-M.; Rocktäschel, T.; Lukasiewicz, T.; and Blunsom, P. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Chen, J.; Gan, C.; Cheng, S.; Zhou, H.; Xiao, Y.; and Li, L. 2022. Unsupervised Editing for Counterfactual Stories. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10473–10481.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Creswell, A.; Shanahan, M.; and Higgins, I. 2022. Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning. *arXiv preprint arXiv:2205.09712*.

Cui, L.; Cheng, S.; Wu, Y.; and Zhang, Y. 2021. On Commonsense Cues in BERT for Solving Commonsense Tasks. In *FINDINGS*.

Danilevsky, M.; Qian, K.; Aharonov, R.; Katsis, Y.; Kawas, B.; and Sen, P. 2020. A survey of the state of explainable AI for natural language processing. *arXiv preprint arXiv:2010.00711*.

Fadel, A.; Al-Ayyoub, M.; and Cambria, E. 2020. JUSTers at SemEval-2020 Task 4: Evaluating Transformer Models against Commonsense Validation and Explanation. In *SEMEVAL*.

Geva, M.; Goldberg, Y.; and Berant, J. 2019. Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. *ArXiv*, abs/1908.07898.

Hendricks, L. A.; Akata, Z.; Rohrbach, M.; Donahue, J.; Schiele, B.; and Darrell, T. 2016. Generating visual explanations. In *European conference on computer vision*, 3–19. Springer.

Hoffmann, J.; and Magazzeni, D. 2019. Explainable AI planning (XAIP): overview and the case of contrastive explanation. *Reasoning Web. Explainable Artificial Intelligence*, 277–282.

Jon, J.; Fajcik, M.; Docekal, M.; and Smrz, P. 2020. BUT-FIT at SemEval-2020 Task 4: Multilingual Commonsense. *ArXiv*, abs/2008.07259.

Konar, A.; Huang, C.; Trabelsi, A.; and Zaiane, O. R. 2020. Ana at semeval-2020 task 4: Multi-task learning for commonsense reasoning (union). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 367–373.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Lin, B. Y.; Shen, M.; Zhou, W.; Zhou, P.; Bhagavatula, C.; Choi, Y.; and Ren, X. 2020. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. In *FINDINGS*.

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

Lipton, P. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27: 247–266.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, Y.; Bartolo, M.; Moore, A.; Riedel, S.; and Stenetorp, P. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Miao, N.; Zhou, H.; Mou, L.; Yan, R.; and Li, L. 2019. Cgmh: Constrained sentence generation by metropolis-hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6834–6842.

Niven, T.; and Kao, H.-Y. 2019. Probing Neural Network Comprehension of Natural Language Arguments. *ArXiv*, abs/1907.07355.

Novikova, J.; Dušek, O.; Curry, A. C.; and Rieser, V. 2017. Why we need new evaluation metrics for NLG. *arXiv preprint arXiv:1707.06875*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Paranjape, B.; Michael, J.; Ghazvininejad, M.; Zettlemoyer, L.; and Hajishirzi, H. 2021. Prompting contrastive explanations for commonsense reasoning tasks. *arXiv preprint arXiv:2106.06823*.

Park, D. H.; Hendricks, L. A.; Akata, Z.; Rohrbach, A.; Schiele, B.; Darrell, T.; and Rohrbach, M. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8779–8788.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Ross, A.; Marasović, A.; and Peters, M. E. 2020. Explaining NLP models via minimal contrastive editing (MiCE). *arXiv preprint arXiv:2012.13985*.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Scao, T. L.; Raja, A.; et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 1223–1237. Springer.

Wan, J.; and Huang, X. 2020. Kalm at semeval-2020 task 4: Knowledge-aware language models for comprehension and generation. *arXiv preprint arXiv:2005.11768*.

Wang, C.; Liang, S.; Jin, Y.; Wang, Y.; Zhu, X.; and Zhang, Y. 2020. SemEval-2020 Task 4: Commonsense Validation and Explanation. In *SEMEVAL*.

Wang, C.; Liang, S.; Zhang, Y.; Li, X.; and Gao, T. 2019. Does it make sense? and why? a pilot study for sense making and explanation. *arXiv preprint arXiv:1906.00363*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Chi, E.; Le, Q.; and Zhou, D. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv*, abs/2201.11903.

Wiegreffe, S.; Hessel, J.; Swayamdipta, S.; Riedl, M.; and Choi, Y. 2021. Reframing human-ai collaboration for generating free-text explanations. *arXiv preprint arXiv:2112.08674*.

Wiegreffe, S.; and Marasovic, A. 2021. Teach me to explain: A review of datasets for explainable natural language processing. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Zelikman, E.; Wu, Y.; and Goodman, N. D. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhao, W.; Peyrard, M.; Liu, F.; Gao, Y.; Meyer, C. M.; and Eger, S. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, 12697–12706. PMLR.

Zhong, R.; Snell, C.; Klein, D.; and Steinhardt, J. 2022. Describing Differences between Text Distributions with Natural Language. In *International Conference on Machine Learning*, 27099–27116. PMLR.