ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240815

DINEX: Interpretable NLP via Diversified Natural Language Explanations

Xugang Zhou[a,1], **JinDian Su**[a,*] **and WeiZheng Gu**[a,1]

[a]South China University Of Technology, GuangDong, China

Abstract. Natural Language Explanations (NLE) are becoming increasingly important in Interpretable Natural Language Processing, which can clarify the reasoning process and improve performance. It is essential to utilize human-authored gold explanations to improve the quality of explanations produced by the generator. However, the gold standard explanations within the same dataset are produced by a fixed group of annotators, leading to a more homogeneous perspective and style. This has inspired us to improve the quality of generated explanations by enhancing the diversity of the training set. Based on this, we introduce DINEX, a two-stage framework comprising a diversified explanation generator and an explanation-aware predictor, suitable for any task related to NLE. The first stage of DINEX augments the generator's training set through two approaches: Semantic Similarity Sampling (SSS) and Structural Variety Generation (SVG). This enables the generator to learn how to produce NLE from diverse perspectives and styles. In the second stage, DINEX improves the predictor's ability to capture the complementary semantics between contexts and explanations. It also reduces the impact of noise on predictions through dynamic perturbations. We conduct experiments on four datasets in the domains of Question Answering and Reasoning. The results show that DINEX achieves an average performance improvement of 5.49%, establishing a new state-of-the-art on the ComVE dataset. Evaluations by human evaluators and Large Language Models (LLMs) demonstrate that DINEX-generated explanations surpass the baseline in quality across 62.4% of the test samples.

1 Introduction

Neural language models achieve remarkable performance across a variety of NLP tasks [20], including Sentiment Analysis [39], Question Answering (QA) [14, 30], and Natural Language Inference (NLI) [31]. However, language models are often considered "black boxes" as their reasoning processes remain largely inaccessible. This lack of transparency has sparked interest among researchers in developing methods to provide evidence or explanations for predictions, leading to the emergence of the field called Interpretable NLP [6].

In Interpretable NLP [17, 25], explanations can be categorized into two forms: 1) Extractive Rationales (ER), which focuses on selecting a subset of the input as a crucial evidence for predictions [10]; 2) Natural Language Explanations (NLE), which generates human-readable explanations for predictions [8]. NLE can refer to external information beyond the task inputs, offering greater flexibility in terms of content, style, and length. Moreover, NLE can provide more comprehensive knowledge than ER, increasing interest in leveraging it to enhance performance [11].

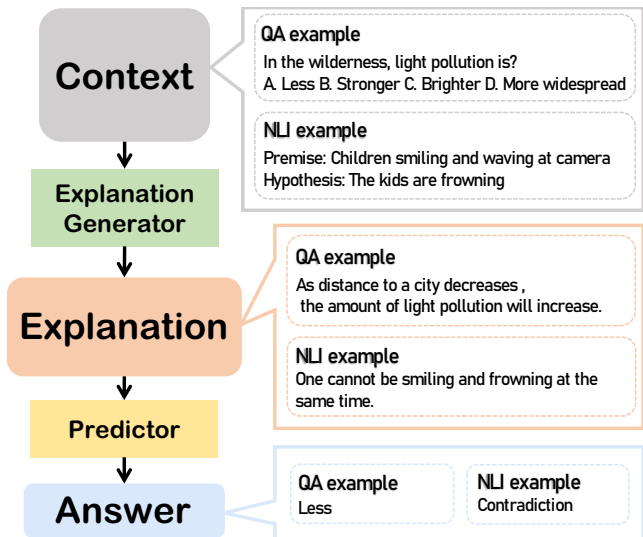


Figure 1: **The typical explain-then-predict framework.** The left part shows the data processing flow. Firstly, a piece of context is input into the generator, which then produces an explanation. The explanation and the context are then input into the predictor, which outputs the prediction. The right part shows specific examples of this framework in QA (Question Answering) and NLI (Natural Language Inference) tasks.

The Explain-then-Predict structure [25] is commonly used to generate NLE, as shown in Figure 1. It first employs an explanation generator to create an explanation for the input, which is then fed into the predictor for the final decision. This structure ensures that the predictor is inherently faithful by construction and leverages knowledge from explanations to improve performance. Generating high-quality explanations and fully utilizing knowledge from them are central tasks in this research field. NILE [22] and LIREX [45] are developed for the NLI task, where they fine-tune explanation generators through the human-authored gold explanations. However, finetuning with one gold explanation per question is hard to ensure the quality of the generated explanations. Prompt-based methods such as PINTO [37] rely on LLMs to generate explanations, but this approach may be impacted by hallucinations. During the prediction, NILE inputs the explanation for each label into the predictor, whereas LIREX only inputs the explanation with the highest confidence. PINTO employs

* Corresponding Author. Email: sujd@scut.edu.cn
[1] Equal contribution.

counterfactual regularization to prevent the spurious relation issue.

We observe that the gold explanations in the same dataset are produced by a fix group of annotators, resulting in a more homogeneous perspective and style. Based on this observation, we explore ways to improve the quality of generated explanations by increasing the diversity of gold explanations. In this work, we propose a Interpretable NLP framework via **DI**versified **N**atural Language **EX**planations (**DINEX**). As shown in Figure 2, DINEX, a two-stage framework, contains a diversified explanation generator and an explanation-aware predictor. It focuses on improvement from two perspectives: enhancing the quality of generated explanations and fully utilizing the explanations. In the first stage, DINEX introduces diverse training data through two data augmentation methods: Semantic Similarity Sampling (SSS) and Structure Variety Generation (SVG). SSS introduces explanations from other datasets with similar semantic. We estimate the semantic distribution of different datasets via *Kernel Density Estimation* [34] to filter out semantically similar data. SVG utilizes a small-scale (7B parameters) LLM for data augmentation through paraphrase and back-translation instructions. This method produces explanations with consistent semantics but varying structures. In the second stage, Explanation-Aware Predictor allows contexts and explanations to capture complementary information, and applies Dynamic Perturbations to against noise.

We evaluate DINEX on four different datasets: ECQA [4], OBQA [28], e-SNLI [9] and ComVE [36]. The first two are designed for the QA task, while the latter two focus on the Reasoning task. Experimental results show that DINEX-SSS, which utilizes the Semantic Similarity Sampling data augmentation method, achieves an average performance improvement of 4.41%. Similarly, DINEX-SVG, employing the Structural Variety Generation method, records an average performance gain of 5.29%. We also assess the impact of explanations generated by different methods on LLMs performance. Evaluations conducted on two open-source models, LLaMA2-7B [35] and Mistral-7B [18], demonstrate that explanations produced by DINEX led to an 8.4% performance improvement. Furthermore, we evaluate the quality of explanations generated by DINEX under the same conditions. Through human evaluators and powerful LLMs (GPT-4 [2] and Claude 3 Sonnet [1]) assessments, the results indicate that explanations generated by DINEX perform better in 62.4% of the test samples. Our contributions can be summarized as follows:

1. We propose two distinct data augmentation methods: Semantic Similarity Sampling (SSS) and Structure Variety Generation (SVG). We are the first to show that introducing diverse data can improve the quality of generated explanations.
2. We propose an explain-then-predict framework for the Interpretable NLP, named DINEX. DINEX contains a diversified explanation generator and an explanation-aware predictor.
3. Our experimental results show that DINEX outperforms strong baselines on four datasets, and achieves SOTA on the ComVE dataset. In addition, compared to previous data augmentation methods, the performance improvement of SSS and SVG is significant.

2 Related Work

2.1 Interpretable NLP

The black-box nature of language models is raising concerns. For instance, it's unclear whether a model's good performance on datasets is due to an understanding of the task or just relying on shortcuts in the data. Moreover, enhancing the credibility of models in high-risk fields such as healthcare and law is critically important. Given these concerns, the Interpretable NLP has gained prominence. It demands that each prediction be accompanied by an explanation. Through such explanations, it becomes possible to quantify biases and fairness, understand the predictive behaviors of the models, and ensure robustness [26].

There are two types of explanations in Interpretable NLP: 1) Extractive Rationales (ER), which focuses on selecting a subset of the input as the crucial basis for predictions. ER is typically applied to knowledge-intensive tasks such as reading comprehension and intent recognition. Numerous regularization methods have been proposed to align the rationales extracted by language models with those manually annotated. 2) Natural Language Explanations (NLE), which generates human-readable explanations. Compared to ER, NLE offers greater flexibility in content, form, and length. Additionally, NLE is capable of incorporating knowledge beyond the task-specific inputs.

Recent researches in NLE focus on Reasoning and QA tasks. The Reasoning domain includes tasks such as NLI, Commonsense Validation, and Sentiment Classification. A notable characteristic of these tasks is their labels are fixed. For example, NLI task includes three labels (entail, contradict and neutral), while the labels of commonsense validation task are true or false. Many studies suggest that explanations could lead to label leakage [32, 16]. Specifically, if each question is tied to a single explanation, the predictor might choose answers by identifying overlaps between the explanation and candidates. To address this problem, NILE [22] and LIREX [45] fine-tune an explanation generator for each label, allowing the predictor to decide which explanation to rely on. However, in QA task, the candidates for each question are distinct, rendering it infeasible to construct a specific explanation generator for each label. Therefore, PINTO [37] leverages LLMs to generate a choice-specific explanation for each option through prompt. However, in most instances, the explanations for incorrect candidates are unrelated to the corresponding one, thereby failing to prevent label leakage.

2.2 Data Augmentation

Data augmentation is a technique that generates additional training data by applying heuristic transformations to existing training examples, aiming to obtain more robust and accurate models [19]. A key consideration for data augmentation is that the distribution of the augmented data should be neither too similar nor too divergent from the original data [13].

Training on examples that do not represent the given domain can lead to increased overfitting or poor performance. Data augmentation has proven beneficial for many NLP tasks, and researchers have proposed various data augmentation methods. For example, Zhang [44] and Wei [40] introduce a wordlist-based replacement approach, while Wang [38] and Kobayashi [21] adopt embedding-based word replacement techniques. Wei [40] and Xie [41] employ back-translation methods to generate data. In recent years, more work has shifted towards using generative models to create additional training examples [23, 5].

3 DINEX

DINEX is a two-stage framework, contains a diversified explanation generator and an explanation-aware predictor with regularization. Figure 2 shows the architecture of DINEX. In the first stage, given a context, the generator produces a variable number of explanations depending on the task type. For tasks with fixed labels, such as NLI

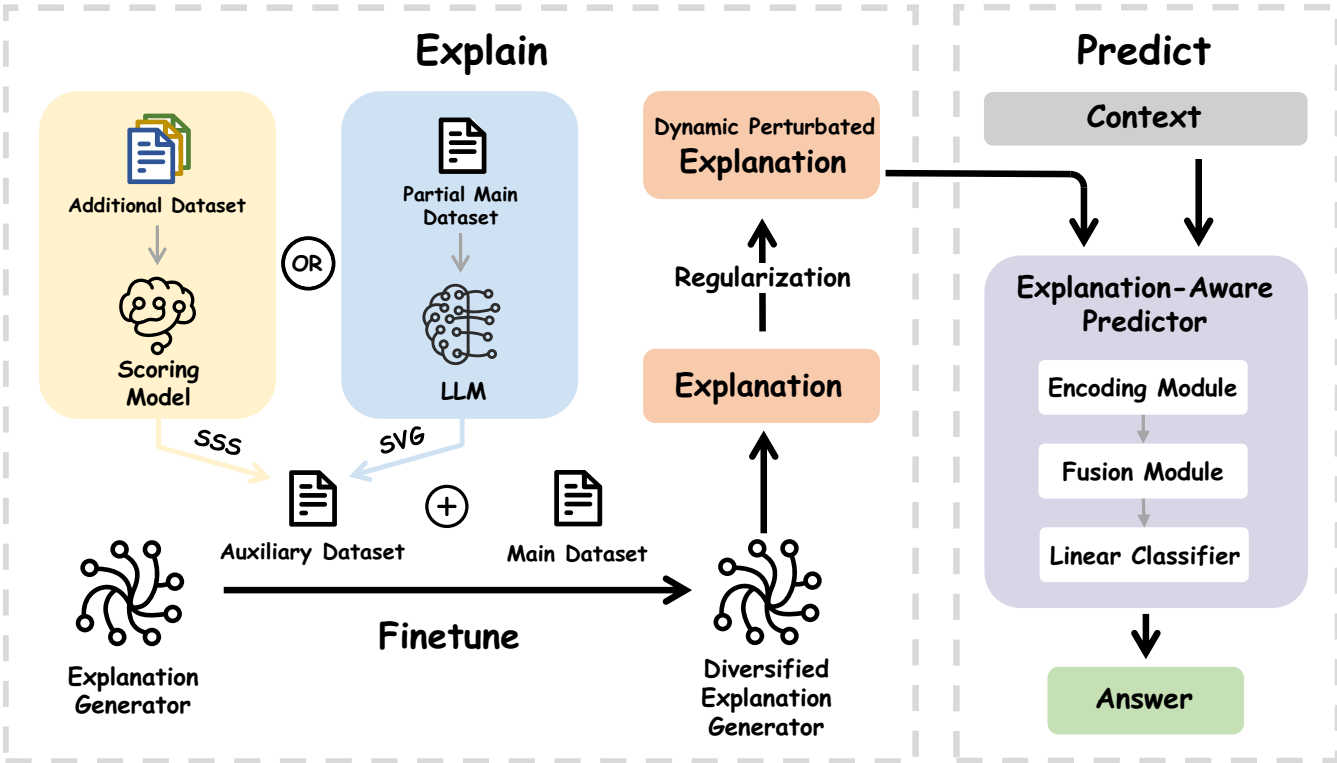


Figure 2: **An overview of DINEX framework.** The left part describes the fine-tuning and inference process of the explanation generator. The training set is expanded by SSS or SVG methods, and the generator is fine-tuned through diverse explanations. The right part shows the predict process of the explanation-aware predictor.

where each sample's candidate labels are "entail," "contradict" and "neutral," multiple generators are trained, each dedicated to producing explanations for a specific label. The context of NLI consists of a premise and a hypothesis. For tasks with non-fixed labels, such as QA where each question has different candidates, only one generator is trained to produce explanations. Its context is formed by concatenating the question and candidates. In the second stage, the predictor makes the final decision based on the context and the generated explanations.

3.1 Diversified Explanation Generator

Since the explanations in the dataset are annotated by the same group, the explanation style is monotonous. Previous works primarily rely on explanations from corresponding datasets to fine-tune the generator [11, 27]. This approach limits the generator's ability to learn diverse representational structures, consequently reducing the quality of the explanations. Therefore, it is necessary to incorporate data with varied structures into the training set. Feng [13] suggests that the distribution of augmented data should neither be too similar nor too divergent from the original data. We further refine this idea, suggesting that the augmented data should satisfy two specific conditions:

Semantic Similarity: The introduced samples should match the semantic distribution of the main dataset. Excessive semantic changes in the dataset can degrade the expressiveness of the model [42].

Structural Variety: The introduced explanations should adopt a distinct style from the main dataset to augment the expressive diversity.

Across different datasets, the representation structure of explanations varies significantly, satisfying the need for *Structural Variety*. We need to identify data from extra datasets that exhibit *semantic similarity* to the main dataset. By conducting initial verification experiments, we find that there are overlapping parts in the semantic distributions of different datasets. Therefore, we first train a regression scoring model M_{scorer} to quantify the similarity. And then we implement a filtering algorithm to choose the semantically similar data.

Training M_{scorer} requires a specially designed dataset. To construct this training set D_{scorer}, we randomly select m samples from each of the main dataset D_{main} and the extra dataset D_{extra} , totaling $2m$ samples:

$$D_{\text{scorer}} = \{(x_i, y_i) \mid 1 \leq i \leq 2m\} \tag{1}$$

x_i is the explanation of the i-th data, and y_i depends on the source of this data:

$$y_i = \begin{cases} 1, & x_i \in D_{\text{main}} \\ 0, & x_i \in D_{\text{extra}} \end{cases} \tag{2}$$

Then we train the regression model M_{scorer} with D_{scorer}. After that, all samples from D_{main} and D_{extra} are rated by the M_{scorer} to obtain the sets S_{main} and S_{extra}:

$$\begin{aligned} S_{\text{main}} &= \{(x_j, s_j) \mid 1 \leq j \leq m\} \\ S_{\text{extra}} &= \{(x_k, s_k) \mid 1 \leq k \leq m\} \end{aligned} \tag{3}$$

$s_j, s_k \in [0, 1]$ is the *semantic similarity* score.

After obtaining S_{main} and S_{extra} , we apply a filtering algorithm to construct the final auxiliary dataset D_{SSS}, with the details as follows:

1. Apply *Gaussian Kernel Density Estimation* to s_j and s_k respectively, resulting in probability density functions $\hat{f}_{\text{main}}(s)$ and $\hat{f}_{\text{extra}}(s)$. Formulas 4, 5, and 6 describe only the fitting process for

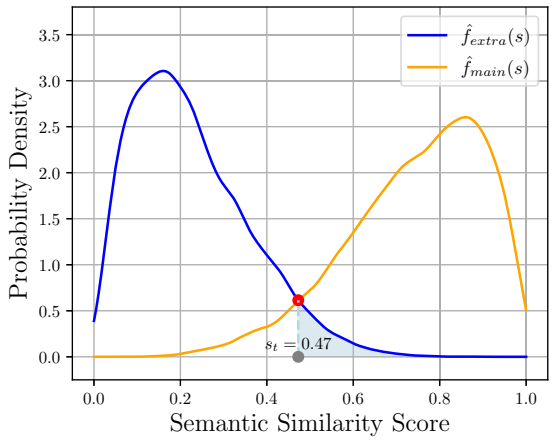


Figure 3: **An example of $\hat{f}_{\text{main}}(s)$ and $\hat{f}_{\text{extra}}(s)$.** The red dot is the intersection point of $\hat{f}_{\text{main}}(s)$ and $\hat{f}_{\text{extra}}(s)$. The light blue area to the lower right of it represents the part of D_{extra} whose semantics can be accepted by D_{main}.

$\hat{f}_{\text{main}}(s)$. The calculation process for $\hat{f}_{\text{extra}}(s)$ is identical, therefore, it does not require separate formulas.
a) Set the kernel function for D_{main}:

$$K_{\text{main}}(s - s_j) = \frac{\exp\left(-\frac{(s-s_j)^2}{2h_{\text{main}}^2}\right)}{\sqrt{2\pi}h_{\text{main}}} \tag{4}$$

h_{main} is the bandwidths chosen by Silverman's rule:

$$h_{\text{main}} = \left(\frac{4\hat{\sigma}_{\text{main}}^5}{3m}\right)^{1/5} \tag{5}$$

$\hat{\sigma}_{\text{main}}$ is the standard deviations of D_{main}.
b) Perform *Kernel Density Estimation* and calculate the fitted functions:

$$\hat{f}_{\text{main}}(s) = \frac{1}{m}\sum_{j=1}^{m} K_{\text{main}}\,(s - s_j) \tag{6}$$

2. Calculate the intersection point of $\hat{f}_{\text{main}}(s)$ and $\hat{f}_{\text{extra}}(s)$, and note the corresponding score as s_{t}. An example is illustrated in the Figure 3.
3. Construct the auxiliary dataset D_{SSS}:

$$D_{sss} = \{(c_l, x_l) \mid (x_l, s_l) \in S_{\text{extra}}, s_l \geq s_t\} \tag{7}$$

where x_l is the explanation, and c_l is the context for this data.

In addition, we introduced a hyperparameter λ_s to control the size of D_{SSS} for preventing excessive influence on the semantics of D_{main}.

$$|D_{SSS}| \leq \lambda_s |D_{\text{main}}| \tag{8}$$

3.1.1 Structure Variety Generation

Structure Variety Generation (SVG) aims to rewrite explanations within the main dataset, providing multiple stylistic explanations for a question. The process of rewriting alters the form while preserving the semantic. Through a small-scale LLM, SVG employs two different instructions for rewriting: paraphrase and back-translation. Specifically, the paraphrase instructs the LLM to rephrase explanations as much as possible without changing the meaning. The back-translation instruction involves an "English → French → English" translation process.

SVG creates an auxiliary dataset D_{SVG}, and introduces a hyperparameter λ_v to limit its size:

$$|D_{SVG}| \leq \lambda_v |D_{\text{main}}| \tag{9}$$

3.2 Explanation-Aware Predictor

In the second stage, the predictor receives a piece of context along with explanations. For tasks with fixed labels, the predictor employs contexts and explanations corresponding to different labels for the decision. For tasks with non-fixed labels, the predictor utilizes contexts and a single explanation to predict the final outcome. It is important to acknowledge that the presence of a single explanation in QA tasks can lead to potential label leakage issues. To prevent shortcut problems that may arise from "overlap", the explanation fed into the predictor is randomly inserted among various candidates. For more detailed experimental procedures, refer to Section 4.1.

3.2.1 Explanation-Awareness

In previous studies, the predictor's input for each question is $s = [context, explanation]$. However, direct concatenating may fail to accurately represent the information between the question and explanation. To address this, we divide the input into s_1 and s_2:

$$s_1 = [context, explanation]\ s_2 \quad = [explanation] \tag{10}$$

The pretrained model encodes s_1 and s_2, respectively. Then the encoded vectors are fed to the fusion module, which contains a cross-attention and GRUs. The cross-attention captures the associated semantics of them, which are then integrated by GRUs. Finally, the ouput is calculated via a linear layer.

3.2.2 Dynamically Perturbed Explanation

In the early epochs of training, the predictor may fit to the noise in the explanations such as spurious correlations [7, 15]. To ensure the predicting follows the explanation semantics faithfully, we implement a regularization algorithm to dynamically perturb the explanations in s_1 and s_2 during training.

We randomly choose an explanation from s_1 or s_2, and each token has a probability p of being masked. The value of p decreases gradually in each epoch:

$$p = \frac{p_{\text{origin}}}{1 + \gamma \cdot epoch} \tag{11}$$

where p_{origin} is the initial probability, γ is the decay coefficient, and $epoch$ is the current number of epochs. This approach avoids the model from fitting noise in the early epochs.

4 Experiment Setup

We conducted experiments on two Interpretable NLP tasks to demonstrate the effectiveness of DINEX.
Question Answering: This task requires the model to answer specific questions. Importantly, the candidates differ for each question. We utilize two datasets:
ECQA[3]: A dataset constructed from the CommonsenseQA benchmark [33], which contains explanations supporting correct answers.

OBQA[28]: A four-choice Scientific QA dataset based on open books, where each question contains one fact that supports the correct answer.

Reasoning: This task requires the model to reason answers based on varying contexts, such as the premises and hypotheses in NLI tasks, while all samples have the same set of candidate labels. We use two datasets:

ComVE[36]: A two-choice commonsense validation dataset that contains two statements, and one of them violates commonsense. The model needs to reason which statement is against commonsense.

e-SNLI[9]: A classic NLI dataset consists of premises paired with hypotheses. The model is required to reason about their relationship, determining whether it is neutral, contradict, or entail.

4.1 Implementation Details

For the *Question Answering* task, we train one generator and one predictor. The generator receives the question along with all candidates, and generates an explanation. Typically, the explanation have lexical overlap with the correct answer. The predictor may exploit the overlap to make a prediction, if the explanation and candidates are both provided as input to the predictor. This phenomenon is referred to as "overlap shortcut". To prevent the predictor from it, we randomly insert candidates from the question into the the explanation, ensuring that each explanation contains multiple candidates. Therefore, the predictor cannot rely on the "overlap shortcut" to determine the correct answer. For the *Reasoning* task, we train a generator for each label, which only generates explanations relevant to it, avoiding the issue of label leakage [22]. Similar to the QA task, we train only one predictor.

In terms of the specific backbone selection, we choose Flan-T5 [12] as the explanation generator, which is an encoder-decoder model pretrained with instructions. We select the large version with 780M parameters for a balance between generative ability and fine-tuning cost. When applying the SSS approach, each dataset selects semantically similar data from other datasets. Considering the SVG approach, Mixtral-7B [18], a small-scale LLM, is utilized to rewrite 50% of the samples in each dataset. For the explanation-aware predictor, we use a pre-trained model – RoBERTa-Large [24] with 340M parameters. The replacement rate of dynamic perturbations ranges between 10% and 30%.

4.2 Baselines

Our experiments consider a representative range of strong baselines.

w/o NLE: Fine-tune a RoBERTa model to select the answer without any explanation.

w/ gold NLE: Fine-tune a RoBERTa model using the gold explanations provided in the training set to predict the answer.

We also compared recent related work:

EASE[43]: An LLM-based in-context learning framework integrating explanations into the ensemble procedure through soft probabilities.

FLamE[46]: A two-stage few-shot learning framework utilizing GPT-3 and RoBERTa.

PINTO[37]: A prompted pipeline framework employing a GPT-NeoX to generate explanations.

KNIFE[11]: It distills NLE knowledge from a teacher LM to a student LM.

REFER[26]: An end-to-end rationale extraction framework that optimizes for faithfulness, plausibility, and performance.

NILE[22]: It is an explan-than-predict framework, a classical approach for using explanations to assist in NLP tasks. It employs GPT-2 as the generator and RoBERTa as the predictor.

4.3 Data Augmentation Methods

In DINEX, we enable the generator to learn the diversity of natural language through data augmentation methods (SSS and SVG). Therefore, we compare three other data augmentation methods, the details of which are as follows:

EDA[40]: Randomly insert, delete, and swap 30% of the tokens in an explanation to obtain new data.

REP[44]: Replace words in the explanation with synonyms. Specifically, this method uses the Wordnet implementation from NLTK. We replace 30% of the words in the explanation with synonyms.

C&R[29]: Mask (Corrupt) 30% of the tokens in the explanation and a BERT model is used to make predictions (Reconstruct), thereby generating a new explanation.

4.4 DINEX Variants

We aim to explore the impact of different modules on the performance of the DINEX framework. Therefore, we modify and replace some modules, resulting in several DINEX variants. The first variant is Standard. It is a basic explain-then-predict framework that only utilizes a Flan-T5 as the explanation generator and a RoBERTa as the predictor. This variant does not use any data augmentation methods or Explanation-Awareness.

The first phase of DINEX requires a data augmentation method.We explore multiple variants employing different data augmentation methods. Alongside the two methods we proposed (SSS and SVG), these variants also incorporated other data augmentation approaches (EDA, REP, and C&R) for comparative purposes. For fairness, each auxiliary dataset produced by the data augmentation methods is at most 25% the size of the main dataset.

In the second phase of DINEX, we use an Explanation-Aware Predictor to make choice. To assess its impact, we replace the Explanation-Aware Predictor (EAP) with a Basic Predictor (BP) as a new variant. BP directly leverages a RoBERTa model for prediction. In addition, we conduct experiments on a special variant: In the first phase, it does not generate explanations using a generator, but directly use the Gold explanations from the dataset. In the second phase, it use EAP for prediction. This special variant can be used to observe the effectiveness of Explanation-Awareness on Gold explanations.

5 Results

5.1 Main Results

In Table 1, we illustrate the performance comparison between DINEX-Best and previously representative work across four datasets. Table 2 shows the performance of different DINEX variants introduced in Section 4.4. We summarize the following conclusions from the results.

Firstly, incorporating explanations as additional signals can significantly improve performance. On four datasets, the accuracy with gold NLE (*w/ gold NLE*) far exceeds that without NLE (*w/o NLE*), with the improvement in the ECQA dataset reaching an impressive 41.07%.

Secondly, increasing the diversity of explanations within the training set significantly enhances the task's performance. As shown in Table 2, when using the Basic Predictor (BP), SSS and SVG methods exhibit average improvement of 2.67% and 4.20% respectively, compared to the Standard variant. Among other data augmentation

Method	QA(Accuracy)		Reasoning(Accuracy)	
	ECQA	OBQA	ComVE	e-SNLI
w/o NLE	$55.93_{\pm 2.14}$	$56.31_{\pm 2.54}$	$88.60_{\pm 0.97}$	$85.21_{\pm 0.84}$
w/ Gold NLE	$97.00_{\pm 0.64}$	$65.34_{\pm 1.56}$	$92.51_{\pm 0.53}$	$94.21_{\pm 0.61}$
FLamE	45.11	46.23	80.24	84.98
EASE	60.45	64.43	86.84	86.79
PINTO	73.50	58.85	90.23	90.91
KNIFE	90.10	61.53	92.87	91.25
REFER	-	-	-	90.48
NILE	88.31	59.72	91.71	91.86
Standard	$89.06_{\pm 0.61}$	$59.03_{\pm 0.82}$	$90.31_{\pm 0.40}$	$89.92_{\pm 0.74}$
DINEX-Best	$\mathbf{93.75_{\pm 0.51}}$	$\mathbf{67.88_{\pm 0.41}}$	$\mathbf{97.50_{\pm 0.54}}$	$\mathbf{91.15_{\pm 0.62}}$

Table 1: **Main Results.** This table reports the performance of each strong baseline and DINEX best results on the four datasets. We report the mean and standard deviation (std) of accuracy over random seeds for the results, using the format "mean ± std". We bold the best results for each dataset for methods that used generated explanations.

methods, the rule-based Easy Data Augmentation (EDA) has a negative impact on performance, by disrupting the semantic information of sentences. Conversely, the synonym replacement method (REP) and the corrupt-and-reconstruct method (C&R) both enhance the diversity of explanations in terms of structure and style, resulting in improved performance to a certain extent.

Thirdly, table 2 reveals that SVG performs better on average than SSS. This may be due to the data generated by the SVG method remains semantically consistent with the original data. In contrast, the data introduced by SSS inevitably differs from the main dataset. However, SSS leverages existing datasets at a lower cost, whereas SVG requires a LLM to generate data, resulting in more time and computational overhead.

Finally, Explanation-Awareness has a positive impact on performance. Table 2 shows that variants using the Explanation-Aware Predictor (EAP) achieve an average performance improvement of 1.62% compared to the Basic Predictor (BP). However, it is noteworthy that Explanation-Awareness does not perform well on the ECQA dataset. This may be due to the high quality of explanations in ECQA, which allows the BP to perform well. The high quality of explanations in ECQA is evident from Table 1, where *w/ gold NLE* outperform *w/o NLE* by 41.07%.

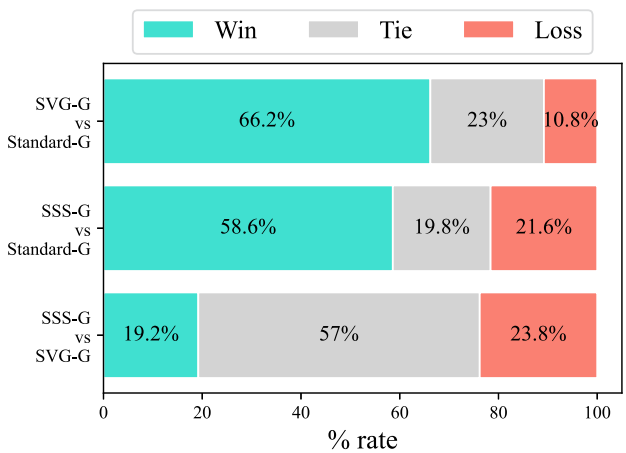


Figure 4: **Comparison results of explanation quality between SVG-G, SSS-G, and Standard-G.**

Variants	QA(Accuracy)		Reasoning(Accuracy)	
	ECQA	OBQA	ComVE	e-SNLI
Standard	$89.06_{\pm 0.61}$	$59.03_{\pm 0.82}$	$90.31_{\pm 1.12}$	$89.92_{\pm 0.34}$
Gold+EAP	$97.13_{\pm 0.13}$	$68.42_{\pm 3.08}$	$95.27_{\pm 1.76}$	$94.56_{\pm 0.30}$
Gold+BP	$97.00_{\pm 0.64}$	$65.34_{\pm 1.56}$	$92.51_{\pm 1.53}$	$94.21_{\pm 0.61}$
EDA+EAP	$85.43_{\pm 0.86}$	$57.78_{\pm 0.93}$	$88.09_{\pm 1.06}$	$87.34_{\pm 0.29}$
EDA+BP	$83.40_{\pm 0.66}$	$56.82_{\pm 1.02}$	$87.28_{\pm 1.54}$	$87.01_{\pm 0.54}$
REP+EAP	$89.92_{\pm 1.24}$	$62.11_{\pm 0.89}$	$92.33_{\pm 0.92}$	$89.34_{\pm 0.22}$
REP+BP	$88.44_{\pm 0.99}$	$62.01_{\pm 0.77}$	$91.87_{\pm 1.39}$	$89.07_{\pm 0.19}$
C&R+EAP	$90.93_{\pm 0.75}$	$63.23_{\pm 2.30}$	$91.67_{\pm 0.73}$	$89.96_{\pm 0.45}$
C&R+BP	$90.62_{\pm 0.79}$	$62.08_{\pm 1.44}$	$91.03_{\pm 0.85}$	$89.63_{\pm 0.32}$
SSS+EAP	$92.17_{\pm 0.70}$	$65.42_{\pm 1.09}$	**97.50±0.54**	$90.86_{\pm 0.31}$
SSS+BP	$91.44_{\pm 0.84}$	$64.23_{\pm 0.63}$	$93.26_{\pm 1.10}$	$89.24_{\pm 0.44}$
SVG+EAP	$93.20_{\pm 0.42}$	**67.88±0.41**	$97.23_{\pm 0.42}$	**91.15±0.49**
SVG+BP	**93.75±0.51**	$65.82_{\pm 0.42}$	$93.60_{\pm 1.51}$	$91.11_{\pm 0.32}$

Table 2: **Results of DINEX variants.** We name the DINEX variants using the following format: "Data augmentation method used in the first stage + predictor used in the second stage". The predictors are categorized into Explanation-Aware Predictor (EAP) and Basic Predictor (BP).

5.2 *Analysis Results*

5.2.1 *Quality of DINEX-Generated Explantions*

The Main Results (5.1) have demonstrated that both the SSS and SVG methods enhance the performance of downstream tasks. In this section, we further investigate the quality of the explanations generated by these methods through a more direct way. We evaluate the quality of explanations generated by three distinct generators:

SSS-G: The generator fine-tuned using a dataset created with the SSS method.

SVG-G: The generator fine-tuned using a dataset created with the SVG method.

Standard-G: The generator fine-tuned with the original training set.

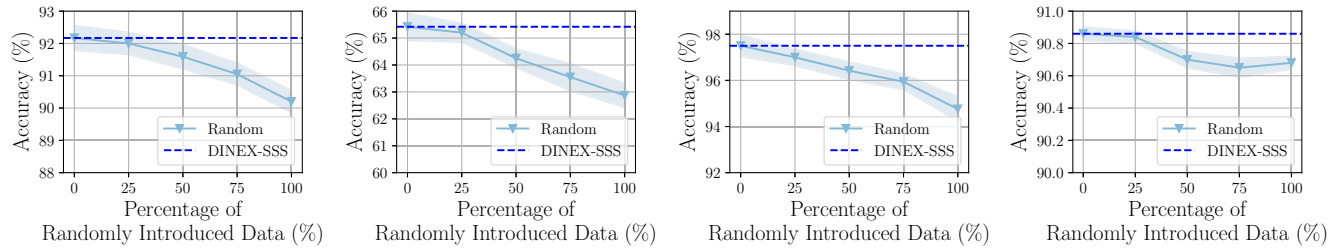
At first, we randomly select 50 samples from the test set and use these generators to produce corresponding explanations for the each sample. Then, we organize these explanations into three pairwise comparative sets: SSS-G vs Standard-G, SVG-G vs Standard-G, and SSS-G vs SVG-G. Each comparative set is evaluated by human evaluators[2] and two powerful LLMs (GPT 4 and Claude 3 Sonnet) to determine which explanation is better. Specifically, the outcomes of each comparison are labeled as "win," "tie," or "lose." We observe that the agreement rate between humans and LLMs reaches 86%.

Consequently, we evaluate 500 samples. To save costs, this time we conduct the evaluations only through LLMs. As shown in Figure 4, the results indicate that both SSS-G and SVG-G outperform the Standard-G, with an average win rate of 62.4%, a tie rate of 21.4%, and a loss rate of 16.2%. The results substantiate that the SSS and SVG data augmentation methods significantly improve the quality of generated explanations. Moreover, SVG slightly outperforms SSS, reinforcing the observations from Section 5.1 that SVG contributes to more enhancements in performance.

5.2.2 *Necessity of Semantic Similarity*

In Section 3.1, we propose that diverse explanations should meet the requirement of *Semantic Similarity*. In this section, we conduct an

[2] The Human evaluators were hired from https://www.upwork.com/hire/data-annotation-specialists/.



(a) ECQA (b) OBQA (c) ComVE (d) e-SNLI

Figure 5: **The performance curves for each dataset after the introduction of random data.** The dashed lines represent the performance with semantically similar data introduced, while the solid lines show the performance after introducing random data at different proportions.

analysis experiment to validate its necessity. When introducing new data from other datasets, we no longer consider whether it satisfies *Semantic Similarity*, but instead choose randomly. Specifically, we replace a certain proportion (0%, 25%, 50%, 75%, and 100%. 0% represents using the original D_{SSS}) of the data in D_{SSS} with random data from other datasets. Figure 5 illustrates the performance curves on four datasets. The dashed lines represent the performance of introducing semantically similar data, while the solid lines represent the performance of randomly introduced data. The reaults show that the method considering *Semantic Similarity* (SSS) is significantly better than the method of random introduction. It is noteworthy that we observe a very limited impact of random data on e-SNLI. This is because the large size of the its training set ensures that the introduction of a small amount of random data does not disrupt the semantic integrity.

5.2.3 *Impact of DINEX-Generated Explanations on LLMs*

In this section, we investigate whether explanations generated by DINEX can influence LLMs. We continue to use the three generators introduced in Section 5.2.1 (SSS-G, SVG-G, Standard-G). Firstly, we concatenate explanations generated by these generators with the context of the task to form prompts. Then, we input these prompts into LLMs (Mixtral-7B and Llama2-7B) to observe their performance on downstream tasks. On the ECQA dataset, we test 500 samples, with each LLM making three rounds of predictions per sample. The experimental results are shown in the table 3. We find that the average performance of SSS and SVG improved by 7.3% over the standard method, demonstrating that high-quality explanations can enhance the predictive accuracy of LLMs.

LLM	SSS-G	SVG-G	Standard-G
LlaMa2 - 7B	91.3	89.5	82.6
Mixtral - 7B	93.2	94.7	87.1

Table 3: **Impact of DINEX-Generated Explanations on LLMs.** We evaluate two small-scale LLMs on the ECQA dataset.

5.2.4 *Ablation Study of EAP*

In Section 3.2.2, We mention that during the initial epochs of training predictors, the model may fit to the noise in the explanations. We introduce Dynamic Perturbation (DP) to counteract this negative impact. To test whether DP can make the model more robust when facing noise, we construct an erroneous dataset, D_{noise}, to train the Basic Predictor. In D_{noise}, each data point is paired with an explanation from another data point instead of its own explanation. We compare the performance of Basic Predictors with and without DP across four datasets. As shown in Table 3, the Predictor with DP outperforms the Predictor without DP. These results confirm that DP substantially enhances the robustness of the predictor.

We expect that the cross-attention mechanism will facilitate a more effective interaction between the query and the explanation, hence we have designed an ablation study for it. Table 5 presents the performance across four datasets. The results indicate a pronounced decrement in performance in the absence of cross-attention. This observation confirms the necessity of cross-attention, which integrates query and explanatory information in a manner that significantly augments the predictive proficiency.

Predictor	Accuracy			
	ECQA	OBQA	ComVE	eSNLI
Basic Predictor w/ DP	**42.26**	**46.02**	**77.35**	**72.17**
Basic Predictor w/o DP	37.98	41.2	71.12	65.64

Table 4: **Perturbation Results.** This table shows the performance of Basic Predictor with or without dynamic perturbation (DP) in the face of incorrect explanations.

Method	Accuracy			
	ECQA	OBQA	ComVE	e-SNLI
SSS + EAP	92.17	65.42	**97.50**	90.86
SSS + EAP w/o CA	91.58	64.51	93.18	89.42
SVG + EAP	**93.20**	**67.88**	97.23	**91.15**
SVG + EAP w/o CA	93.06	66.15	93.84	91.09

Table 5: **Ablation Study on Cross-Attention.** Performance comparison of different variants across various datasets, where "CA" represents Cross-Attention.

6 Conclusion

In this paper, we propose two data augmentation methods to increase the diversity of explanation structures: Semantic Similarity Sampling (SSS) and Structure Variety Generation (SVG). Furthermore, we propose a two-stage Interpretable NLP framework, named DINEX. Its explanation generator produces explanations that are more consistent with human expression via SSS and SVG approaches. And its predictor enables context and explanations to capture complementary information.

The experiment results show DINEX surpasses strong baselines on four datasets, and achieves SOTA performance on the ComVE dataset. Additionally, evaluations by humans and LLMs have demonstrated that DINEX can generate explanations of higher quality.

References

[1] The claude 3 model family: Opus, sonnet, haiku. URL https://api.semanticscholar.org/CorpusID:268232499.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] S. Aggarwal, D. Mandowara, V. Agrawal, D. Khandelwal, P. Singla, and D. Garg. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Jan 2021. doi: 10.18653/v1/2021.acl-long.238. URL http://dx.doi.org/10.18653/v1/2021.acl-long.238.

[4] S. Aggarwal, D. Mandowara, V. Agrawal, D. Khandelwal, P. Singla, and D. Garg. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, 2021.

[5] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper, and N. Zwerdling. Not enough data? deep learning to the rescue! arxiv 2019. *arXiv preprint arXiv:1911.03118*, 1911.

[6] Y. Belinkov, S. Gehrmann, and E. Pavlick. Interpretability and analysis in neural nlp. In *Proceedings of the 58th annual meeting of the association for computational linguistics: tutorial abstracts*, pages 1–5, 2020.

[7] R. Branco, A. Branco, J. Rodrigues, and J. Silva. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, 2021.

[8] E. Cambria, L. Malandri, F. Mercorio, M. Mezzanzanica, and N. Nobani. A survey on xai and natural language explanations. *Information Processing & Management*, 60(1):103111, 2023.

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

[10] A. Chan, M. Sanjabi, L. Mathias, L. Tan, S. Nie, X. Peng, X. Ren, and H. Firooz. Unirex: A unified learning framework for language model rationale extraction. In *International Conference on Machine Learning*, pages 2867–2889. PMLR, 2022.

[11] A. Chan, Z. Zeng, W. Lake, B. Joshi, H. Chen, and X. Ren. Knife: Distilling meta-reasoning knowledge with free-text rationales. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

[12] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[13] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

[14] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005.00646*, 2020.

[15] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[16] P. Hase, S. Zhang, H. Xie, and M. Bansal. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? *arXiv preprint arXiv:2010.04119*, 2020.

[17] A. Jacovi and Y. Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[18] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[19] O. Kashefi and R. Hwa. Quantifying the evaluation of heuristic methods for textual data augmentation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 200–208, 2020.

[20] J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2, 2019.

[21] S. Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

[22] S. Kumar and P. Talukdar. Nile: Natural language inference with faithful natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8730–8742, 2020.

[23] V. Kumar, A. Choudhary, and E. Cho. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*, 2020.

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

[25] Q. Lyu, M. Apidianaki, and C. Callison-Burch. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, pages 1–70, 2024.

[26] M. R. G. Madani and P. Minervini. Refer: An end-to-end rationale extraction framework for explanation regularization. *arXiv preprint arXiv:2310.14418*, 2023.

[27] B. P. Majumder, O.-M. Camburu, T. Lukasiewicz, and J. McAuley. Knowledge-grounded self-rationalization via extractive and natural language explanations. *arXiv preprint arXiv:2106.13876*, 2021.

[28] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[29] N. Ng, K. Cho, and M. Ghassemi. Ssmba: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 1268–1283. Association for Computational Linguistics (ACL), 2020.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21 (1):5485–5551, 2020.

[31] S. Storks, Q. Gao, and J. Y. Chai. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, 2019.

[32] J. Sun, S. Swayamdipta, J. May, and X. Ma. Investigating the benefits of free-form rationales. *arXiv preprint arXiv:2206.11083*, 2022.

[33] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv: Computation and Language,arXiv: Computation and Language*, Nov 2018.

[34] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[35] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[36] C. Wang, S. Liang, Y. Jin, Y. Wang, X. Zhu, and Y. Zhang. Semeval-2020 task 4: Commonsense validation and explanation. *arXiv preprint arXiv:2007.00236*, 2020.

[37] P. Wang, A. Chan, F. Ilievski, M. Chen, and X. Ren. Pinto: Faithful language reasoning using prompt-generated rationales. Nov 2022.

[38] W. Y. Wang and D. Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563, 2015.

[39] M. Wankhade, A. C. S. Rao, and C. Kulkarni. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780, 2022.

[40] J. Wei and K. Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[41] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

[42] X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett, and R. Pasunuru. Complementary explanations for effective in-context learning. Nov 2022.

[43] Y. Yu, J. Shen, T. Liu, Z. Qin, J. N. Yan, J. Liu, C. Zhang, and M. Bendersky. Explanation-aware soft ensemble empowers large language model in-context learning. *arXiv preprint arXiv:2311.07099*, 2023.

[44] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[45] X. Zhao and V. V. Vydiswaran. Lirex: Augmenting language inference with relevant explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14532–14539, 2021.

[46] Y. Zhou, Y. Zhang, and C. Tan. Flame: Few-shot learning from natural language explanations. Jun 2023.