DISCERN: Decoding Systematic Errors in Natural Language for Text Classifiers

Rakesh R. Menon **Shashank Srivastava**
UNC Chapel Hill
{rrmenon, ssrivastava}@cs.unc.edu

Abstract

Despite their high predictive accuracies, current machine learning systems often exhibit systematic biases stemming from annotation artifacts or insufficient support for certain classes in the dataset. Recent work proposes automatic methods for identifying and explaining systematic biases using keywords. We introduce DISCERN, a framework for interpreting systematic biases in text classifiers using language explanations. DISCERN iteratively generates *precise* natural language descriptions of systematic errors by employing an interactive loop between two large language models. Finally, we use the descriptions to improve classifiers by augmenting classifier training sets with synthetically generated instances or annotated examples via active learning. On three text-classification datasets, we demonstrate that language explanations from our framework induce consistent performance improvements that go beyond what is achievable with exemplars of systematic bias. Finally, in human evaluations, we show that users can interpret systematic biases more effectively (by over 25% relative) and efficiently when described through language explanations as opposed to cluster exemplars.[1]

1 Introduction

A broader adoption and trust in machine learning systems would require a confluence of high predictive performance and human interpretability. Despite their high predictive accuracies, current machine learning systems often exhibit systematic biases (Robertson, 2024; Kayser-Bril, 2020; Stuart-Ulin, 2018) stemming from annotation artifacts (Gururangan et al., 2018; McCoy et al., 2019) or insufficient support for certain classes in the dataset (Sagawa* et al., 2020). Such biases impede the deployment of systems for real-world applications. Hence, identifying data *sub-populations* where systems underperform is crucial for a comprehensive understanding of its limitations, thereby guiding future refinement strategies.

In line with this objective, to identify *semantically meaningful* sub-populations whose examples have similar characteristics because of a shared underlying structure, previous work proposes to cluster examples and qualitatively examine clusters where the system performs poorly (d'Eon et al., 2022). In efforts to alleviate the necessity for manual analysis, recent works propose automatic methods to identify and explain underperforming clusters by associating keywords with underperforming clusters (Eyuboglu et al., 2022; Jain et al., 2023; Hua et al., 2023). However, identifying relevant keywords requires domain expertise and even then, they may not capture all error types.

Building on recent advancements in large language models (LLMs, Achiam et al., 2023; Touvron et al., 2023; Jiang et al., 2023), we aim to bridge this gap with open-ended natural language descriptions of error types. Such descriptions can offer two major advantages: (1) language descriptions can help structure the generations or acquisitions of new labeled examples, and (2) articulation can allow developers to audit and intervene in the debugging process. With this premise, we introduce DISCERN, an iterative approach to improve text-classfiers using *precise* natural language descriptions of their systematic errors (see Figure 1).

DISCERN utilizes off-the-shelf large language models for distinct roles. An **explainer LLM** is used to generate predicate-style descriptions[2] for underperforming clusters of training examples. To enhance precision, DISCERN refines the predicates identified by the explainer through an interaction loop. In this loop, an **evaluator LLM** assesses

[1] Code is available at: https://github.com/rrmenon10/DISCERN

[2] Predicate-style descriptions refer to concise statements in natural language that describe characteristics or patterns observed within a specific subset of data.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 19565–19583

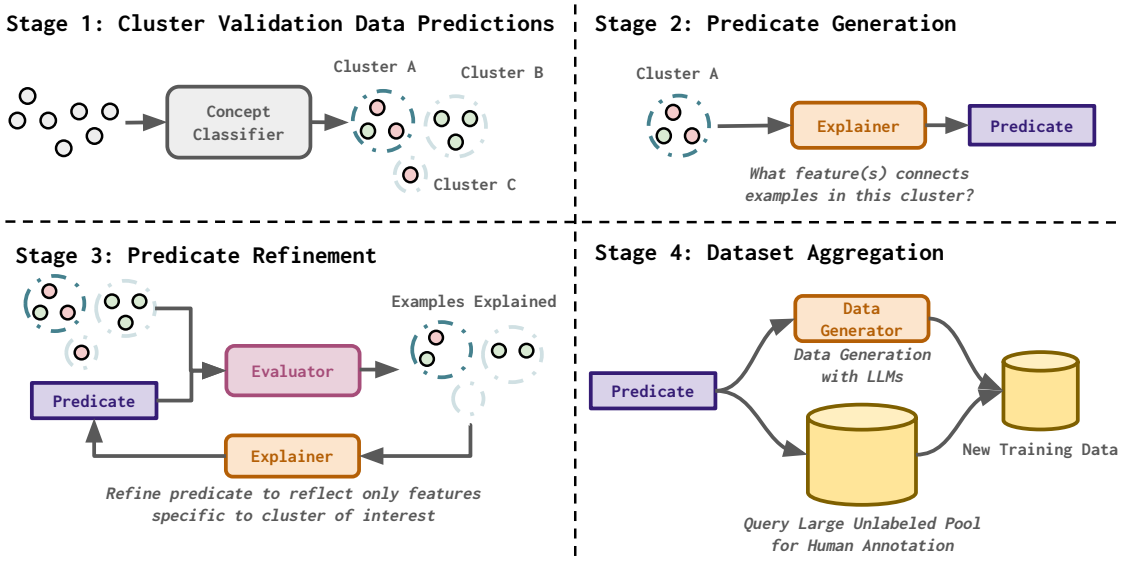

Figure 1: Overview of our classifier debugging framework, DISCERN. The framework comprises four stages: (1) clustering validation set examples to identify data sub-populations where the classifier makes most errors, (2) cluster description generation using an explainer LLM, (3) refining cluster descriptions through interaction between the explainer and evaluator for higher precision, and (4) model refinement through dataset aggregation.

whether the predicate applies exclusively to examples within a given cluster. Using this feedback on the examples successfully explained by the predicate as well as those that it struggles to explain, the explainer dynamically adjusts its prediction until a desired precision threshold is achieved. Finally, the generated descriptions are utilized to augment training sets, either through data augmentation using a **data-generator LLM** or active learning, to retrain and improve the classifier.

In experiments, on a set of three different text-classification tasks, we demonstrate the utility of descriptions generated by our framework in identifying meaningful systematic biases in classifiers. On the AGNews dataset, by augmenting the training set with synthetically generated instances, we are able to achieve statistically significant improvements over baseline approaches that generate instances from examples of biased instances alone (Section 5). In other data sets, DISCERN can reduce misclassification rates in biased clusters by at least 10%. Importantly, we show that language explanations of systematic biases are more helpful for users, and they are 25% more effective in identifying new biased instances (Table 5). Finally, we evaluate the multiple design choices that constitute our framework and ascertain the capacity of our framework to enhance its performance in conjunction with the integration of larger and more robust language models.

Our contributions are as follows:

- A framework for generating precise natural language explanations of systematic errors in models designed for text classification tasks. The precision of the explanations enables a deeper understanding of the underlying biases and aids in developing effective mitigation strategies.
- Quantitative evaluations demonstrating the value of DISCERN's explanations through improved classifier performance by synthetic data augmentation and active learning.[3]
- Qualitative evaluations of DISCERN explanations against other approaches emphasize the crucial role of explanations in an efficient and effective understanding of systematic biases.
- We analyze the role of different design choices that lead to the generalizability of our framework and outline opportunities for improvement.

[3]Code to reproduce experiments will be released on first publication.

2 Related Work

Automatic Failure Discovery. To identify failure modes in model predictions, early works employ manual inspection of model prediction errors (Vasudevan et al., 2022) or hypothesis testing (Poliak et al., 2018), or adversarial testing (Ribeiro et al., 2018; Kiela et al., 2021). However, manual inspection requires extensive domain expertise and can be labor intensive. More recent efforts propose automatic frameworks that approach this problem through the lens of *slice discovery* (Eyuboglu et al., 2022; Hua et al., 2023), where a *slice* represents a portion of the dataset where the model more frequently makes errors in inference. Closer to our work, Rajani et al. (2022) propose SEAL, an interactive visualization tool to describe examples that exhibit high errors using natural language. Different from the objective of this work, we propose to use natural language descriptions as an interpretable medium to refine text-based classifiers.

Model Refinement. To tackle the challenge of underperforming subgroups, previous work has also proposed multiple distributionally robust training strategies (Sagawa* et al., 2020; Liu et al., 2021; Sohoni et al., 2020). Note that these objective functions are complementary to our work and in principle could be utilized to enhance model performance (see (Lee et al., 2024) for how to use language explanations to perform robust optimization). However, according to He et al. (2023), these objectives improve the performance of challenging subgroups at the expense of overall accuracy. We follow the recommendation in He et al. (2023) and use data augmentation and active learning to demonstrate the utility of our approach.

LLM Refinement. LLMs, while adept at many tasks without prior training, struggle with more challenging tasks. As a result, recent studies propose to refine LLM predictions through an iterative verification process. SELF-REFINE (Madaan et al., 2023) proposes iterative feedback generation and refinement of predictions to enhance performance in text and code generation tasks, while SELF-DEBUGGING (Chen et al., 2024) advocates leveraging unit test execution results to enhance code quality. In contrast to these studies, we use refinement to understand classifier behavior, not to enhance individual predictions from LLMs.

Data Augmentation with LLMs. With the growing capabilities of LLMs, recent works have proposed to use LLMs to generate examples to supervise machine learning models (Whitehouse et al., 2023; Dai et al., 2023). Our work differs from these augmentation models in that with DISCERN we infer the high-level semantic concept that connects existing examples before performing augmentation. In other words, natural language (NL) statements act as the intermediate for the augmentation step in our procedure. The benefits of the same can be observed throughout our experiments. Our work can be considered as an improvement that can complement methods in Whitehouse et al. (2023) and Dai et al. (2023) that effectively perform example-based augmentation (our No Description baseline).

3 Method

In this section, we first formally define our problem setup (§3.1). Next, we provide detailed descriptions of the key stages in DISCERN (§3.2).

3.1 Problem Setup

We consider a classifier denoted as $f : \mathbb{X} \rightarrow \mathbb{Y}$, where $\mathbb{X}$ represents textual inputs, such as sentences, and $\mathbb{Y}$ denotes the corresponding set of classification labels for a specific task (e.g., sentiment analysis). The classifier has been initially trained on a dataset, $\mathcal{D}_{train}$. However, it is prone to acquiring spurious correlations between the inputs and outputs due to prevalent issues such as annotation artifacts (Gururangan et al., 2018; McCoy et al., 2019) or inadequate support for certain classes within the dataset (Sagawa* et al., 2020). Our goal, given a validation dataset, $\mathcal{D}_{val}$, is to identify and describe clusters where the misclassification rate exceeds the classifier's general misclassification rate. Formally, we identify clusters c such that, $\mathbb{E}_{(x,y)\sim\mathcal{D}_{val,c}}[f(x) \neq y] > \mathbb{E}_{(x,y)\sim\mathcal{D}_{val}}[f(x) \neq y]$ and utilize the examples in these clusters to inform future classifier refinement. Rather than directly leveraging these problematic examples to augment the training dataset with additional labeled instances, we demonstrate the value of generating natural-language explanations as an intermediary in the process. This strategy not only augments the interpretability and understanding of the model, but can efficiently improve classifier performance.

3.2 DISCERN

Broadly, our framework, DISCERN, is composed of four stages: (1) clustering of validation set examples, (2) predicate generation, (3) predicate refinement, and (4) model refinement with dataset

aggregation. Figure 1 illustrates these four stages in our framework.

Clustering validation set examples. In this stage, we target the detection of systematic biases: situations where the model consistently underperforms on data points that exhibit common characteristics or features. Our goal is to elucidate these biases by identifying sub-populations within the data that share similar features. For this, we perform agglomerative clustering[4] over the data points in the validation set based on their sentence embeddings. We use the `text-embedding-3-small` embeddings for clustering, as these embeddings encode semantics of the text, thus ensuring that the systematic biases we identify are substantively grounded in semantic meaning. Following clustering, we compute the classifier misclassification rate on different clusters and generate predicates exclusively for those clusters that have a higher misclassification rate than the base misclassification rate of the classifier (§3.1).

Predicate generation. Using the examples from the clusters that exhibit a high misclassification rate, we prompt an **explainer LLM** (in our experiments, `gpt-3.5-turbo-0125`) to generate descriptions that precisely capture the defining characteristics of the examples within these clusters. Drawing on recent work in prompting for planning and reasoning (Yao et al., 2023), we employ thought-based prompting to effectively guide the model to identify and articulate the common characteristics that link examples in a cluster. A detailed list of prompts used through different stages of the framework is provided in Appendix B.

Predicate refinement. The explainer LLM in the previous step is directed to generate descriptions that recall the distinctive characteristics of examples within a cluster. However, the resultant descriptions often lack *specificity* and encompass examples that belong to multiple clusters. In other words, the descriptions do not accurately capture the factors that cause the classifier to perform poorly in a particular cluster, thus inadequately representing the systematic bias. Past work suggests that augmenting datasets using inadequate strategies can result in a decrease in overall classifier performance (Ribeiro and Lundberg, 2022). To ensure description specificity for understanding classifier behavior on a target data cluster, we need to ensure that the explaining chain can reason over examples within the cluster and those outside it.

To achieve this, we first assess the specificity of the generated descriptions using an evaluator function, which we refer to as the **evaluator LLM**. The evaluator LLM, instantiated using a secondary LLM, guides the explainer LLM by identifying examples within the target cluster and outside of it that align with the description generated previously. To evaluate alignment, we prompt the evaluator LLM to check if each example in the target cluster (and outside it) satisfies the predicate in the description.[5] Subsequently, the explainer LLM uses the information of the in-cluster and out-of-cluster examples to refine its description to be more precise. We repeat this process until the refined description passes a specific threshold, measured by the evaluator LLM. This threshold is based on the percentage of examples that are satisfied within the target cluster versus those outside it by the description generated using the explainer LLM. Through our iterative refinement process, the model can identify specific characteristics of clusters that explain the systematic bias associated with a classifier.

Model refinement with dataset aggregation. Given the descriptions that have been generated for the classifier, we now focus on how to utilize these descriptions to improve the classifier. In this work, we adopt two different strategies for improving the classifiers given descriptions: (1) **synthetic dataset augmentation** – where we prompt a **data-generator LLM** using the iteratively refined descriptions to generate new examples for the classification task, and (2) **active learning** – where we assume access to a pool of unlabeled examples and augment the training set with annotations for examples that match our descriptions.

4 Experiments

In this section, we outline our experimental procedures to evaluate DISCERN.

Datasets. We use three multiclass text-classification datasets: (1) TREC (Li and Roth, 2002) – a six-class classification task comprising of questions labeled according to the type of the question, (2) AG News (Zhang et al., 2015) – a collection of news articles labeled according to the category of the article, and (3) COVID Tweets

[4]We use the sklearn implementation of AgglomerativeClustering with a distance-based threshold.

[5]The prompt to achieve this can be found Appendix §B.

(Tattle, 2020) – a sentiment-classification task that classifies tweets related to COVID.

Classifiers. Our framework is designed for developers who need to provide low-latency, high-throughput ML solutions with minimal bias. For our experiments, we hence use `distilbert-base-uncased` and `roberta-large`, as they are sufficient to handle complex tasks while being light enough for mobile deployment. This ensures that our approach is practical and effective for real-world applications, enabling users to benefit from advanced ML capabilities on mobile devices offline, without compromising performance or fairness. Hence, we assume only the developer has access to the LLMs while the user at test-time does not have access to a model as complex.

These classifiers are initially trained on a subset of the complete training set to simulate realistic learning scenarios with limited data. Validation sets are subsampled to match these subsets, and the remaining training data is used as an unlabeled pool for active learning experiments.

Metric. We assess the utility of precise and semantically meaningful natural language explanations from DISCERN by evaluating the performance of classifiers trained with augmented data on the validation set. We report the average performance and standard deviation across five random seeds, unless otherwise noted.

Baselines. For the dataset augmentation experiments, we use two baselines to compare with DISCERN. The first baseline is a naive augmentation (or no descriptions) baseline (No Desc.), where we generate additional instances that adhere to the style and semantic content of the cluster exemplars. This baseline helps us to evaluate the role of natural language as a bottleneck for successful model debugging. To establish the value of refinement, our second baseline uses the explanations generated without iterative refinement to augment the training dataset. We refer to this baseline as DISCERN-F[6]. It is worth noting that this baseline, while sharing similarities with the visualization approach proposed in Rajani et al. (2022), is distinct in its application for classifier improvement.[7]

Experimental Setup. We utilize `gpt-3.5-turbo-0125` as our **explainer LLM** and **data generator LLM**. As our **evaluator LLM**, we use `Mixtral-8x7B-Instruct` (Jiang et al., 2024), a recent open source instruction-tuned large language model.[8] Choosing the evaluator LLM to be different from the explainer LLM, allows us to be leverage the diverse perspectives from different models and avoid confirmation bias (Panickssery et al., 2024). This strategic choice also serves as a safeguard against the potential pitfalls of confirmation bias, thus ensuring the quality and accuracy of cluster characterizations. We set the refinement threshold as recognizing more than 80% of examples within a target cluster and less than 20% of the examples outside the cluster and the maximum number of refinement iterations to five. For fair comparison across methods, we only perform dataset augmentation for those clusters that have passed the refinement threshold. Additionally, we do not alter training hyperparameters between pre-debugging and post-debugging stages.

A full list of hyperparameters used in our experiments can be found in Appendix §A.

5 Results and Analyses

Generating synthetic examples using DISCERN descriptions leads to significant classifier improvement. We evaluate the accuracy of the `distilbert` classifier, fine-tuned with examples generated by various methods. Table 1 shows the impact of using 500 and 1000 augmented examples on classifier performance. First, we observe that descriptions of both DISCERN and DISCERN-F improve over the naive augmentation baseline in most settings, with DISCERN showing marginal statistical significance (paired t-test; p-value$= 0.05$) across three datasets and augmentation configurations. This highlights the utility of language descriptions in designing classifier debugging frameworks. Second, our proposed method, DISCERN, consistently outperforms DISCERN-F, showing the benefit of high-precision descriptions with marginal statistical significance (paired t-test; p-value$= 0.09$). Furthermore, in the AG-News news classification task, we note that the addition of 1000 synthetic examples leads to a substantial improvement ($\sim 3\%$ absolute) in classifier accuracy.[9]

[6] F for first explanation generated by the explainer LLM.

[7] This is not an exact replication of SEAL as we use more recent LLMs with the thought-based prompting.

[8] We evaluate other choices of predicate evaluators in §6.

[9] The accuracy improvements obtained for DISCERN in this setting is statistically significant compared to naive aug-

Dataset → # Aug. Ex.	TREC (2000) 500	TREC (2000) 1000	AGNews (1500) 500	AGNews (1500) 1000	Covid (4000) 500	Covid (4000) 1000
Base	58.48		75.8		47.68	
No Exp.	$77.09_{(2.18)}$	$78.04_{(1.74)}$	$80.03_{(1.51)}$	$80.68_{(1.08)}$	$51.07_{(0.66)}$	$48.08_{(1.14)}$
DISCERN-F	$76.99_{(2.49)}$	$78.98_{(1.88)}$	$79.75_{(1.25)}$	$80.96_{(1.98)}$	$51.12_{(0.64)}$	$48.60_{(1.36)}$
DISCERN	$\mathbf{77.20_{(1.80)}}$	$\mathbf{79.21_{(1.53)}}$	$80.39_{(1.35)}$	$\mathbf{83.44_{(1.00)}}^{\dagger}$	$\mathbf{51.55_{(0.47)}}$	$49.06_{(0.79)}$

Table 1: Accuracy of `distilbert-base-uncased` classifier after augmenting the training set with examples that have been generated using different approaches. Numbers in brackets next to dataset names indicate the number of training examples used for learning the initial classifier. **Bold** numbers indicate the best average classifier accuracy across five runs. † indicates statistically significant improvement over other approaches using t-test.

Method	TREC	Covid
Base	100.00	72.73
No Desc.	3.17	30.95
DISCERN-F	4.76	40.91
DISCERN	0.00	27.78

Table 2: Median misclassification rates for erroneous clusters before (Base) and after training of a `distilbert` classifier with 1000 augmented examples using different approaches.

Dataset → # Aug. Ex.	TREC (1500) 500	TREC (1500) 1000	AG News (500) 500	AG News (500) 1000
Base	70.85		41.2	
No Desc.	$71.99_{(14.04)}$	$85.26_{(3.13)}$	$\mathbf{58.88_{(10.57)}}$	$61.96_{(9.47)}$
DISCERN-F	$72.25_{(11.95)}$	$86.51_{(1.98)}$	$58.60_{(10.35)}$	$64.28_{(10.67)}$
DISCERN	$\mathbf{78.11_{(3.38)}}$	$\mathbf{88.54_{(0.90)}}$	$55.44_{(14.50)}$	$\mathbf{67.00_{(10.39)}}$

Table 3: Accuracy of `roberta-large` classifier after augmenting the training set with examples that have been generated using different approaches. Numbers in brackets next to the names of the dataset indicate the number of training examples used to learn the initial classifier. **Bold** numbers indicate the best average classifier accuracy across five runs. Full results in Table 16.

In Table 2, we show that DISCERN substantially reduces misclassification rates in the underperforming clusters on the other two datasets. Specifically, for the TREC and Covid datasets, DISCERN achieves perfect classification and reduces the misclassification rate to 27.78%, respectively. Compared to the baseline of naive augmentation (No Desc.), we observe that DISCERN has a substantial improvement in misclassification rates. More notably, we observe the value of *precision* in language descriptions by comparing the result with DISCERN-F, whose misclassification rates are worse than the naive augmentation baseline.

Figure 2 presents descriptions generated by DISCERN-F and DISCERN for the AGNews datasets (examples for other datasets in Figure 5 in the Appendix). From the descriptions, we can observe the ability of DISCERN to capture the nuances that enable targeted improvement. In particular, DISCERN descriptions provide a more precise observation of *"debate"* in the corresponding news articles, as opposed to DISCERN-F. Put together, these findings underscore the potential of DISCERN to improve classifiers by addressing systematic errors.

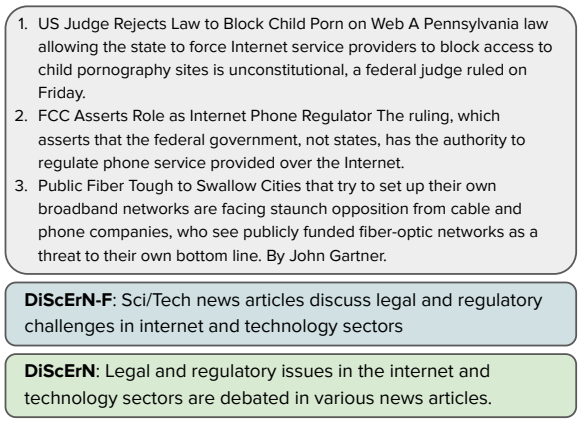

Figure 2: Example of descriptions generated by DISCERN and DISCERN-F for an underperforming cluster in the AGNews dataset. Examples for descriptions with other datasets can be found in the Appendix.

DISCERN improvements generalize across models. We evaluate the performance of a different classifier model, `roberta-large`, to assess the generalization of the observed improvements. In Table 3, we compute the accuracy of the classifier following the augmentation of the training set with examples generated through different approaches. Similar to the results for the `distilbert` classifier, we observe consistent improvements in

mentation and DISCERN-F baselines using an independent samples t-test ($p < 0.05$).

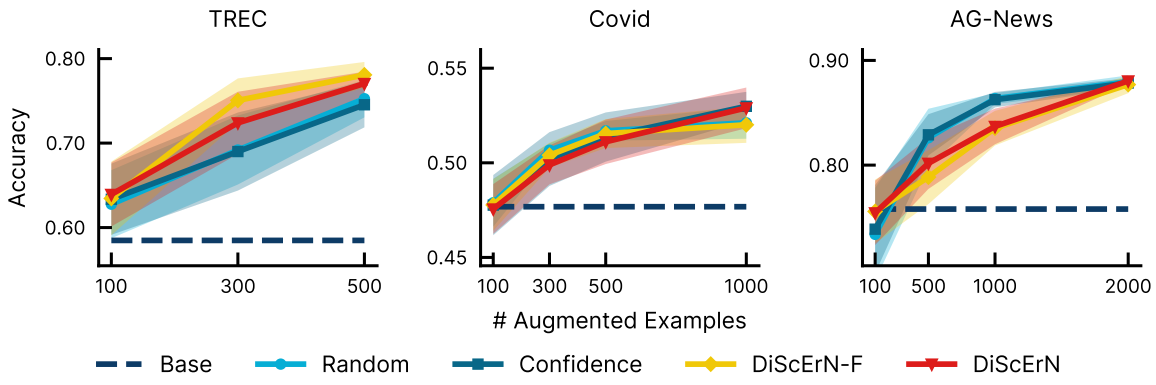

Figure 3: Average accuracy of `distilbert-base-uncased` classifiers after augmenting the training set with examples identified and annotated from a large unlabeled pool using different approaches. Shaded regions indicate the standard deviation over five runs.

classifier accuracy using examples generated using DISCERN descriptions as opposed to the no-description baseline. This highlights the classifier-agnostic utility of our framework in identifying systematic errors and rectifying them through data augmentation.

Active Learning using DISCERN Descriptions. Language descriptions derived using our method can also be used to identify examples from an unlabeled pool that could help improve classifier performance. Consequently, we employ this strategy to identify examples from the unlabeled pool of each of these datasets. Specifically, given the language descriptions, from DISCERN and DISCERN-F, we use the `Mixtral-8x7B-Instruct` model to identify examples that satisfy the predicate mentioned in the description. All examples identified through this process are then added to the training set to retrain the classifier. We measure the classifier accuracies post training with new training set. We use two standard active learning baselines: (a) *random* – annotating and augmenting random examples from the unlabeled pool, and (b) *confidence* – selecting examples predicted with least classifier confidence for annotation and augmentation. Recent work indicates these strategies remain competitive for active learning with large language models (Margatina et al., 2023).

In Figure 3, we plot the accuracy of the classifier as a function of the number of annotated examples incorporated into the training process.[10] We make a couple of observations. First, the addition of examples suggested by DISCERN is better than the addition of random samples to the training set, especially on the TREC and Covid datasets. This suggests that DISCERN is adept at identifying informative examples to improve the classifier. Second, the confidence-based approach predominantly outperforms description-based approaches, particularly when few examples are added to the training set. However, it is noteworthy that the improvement achieved through the DISCERN suggested examples gradually catches up as the number of annotated examples increases.

DISCERN outperform keyword-based approaches. Prior work in NLP has proposed to identify clusters using manually prescribed keywords, typically provided by domain experts (Hua et al., 2023). Here, we compare the performance of DISCERN against the approach (Hua et al., 2023). To this end, we utilize `gpt-3.5-turbo` to generate keywords describing the semantic content for the datasets we use in our experiments. Next, we assign clusters (obtained from agglomerative clustering) towards one of the keywords. Finally, we use the keywords for the cluster to guide the generation for new examples and re-train the classifier. This approach roughly follows the work in DEIM (Hua et al., 2023) and hence we name it DEIM*. In Table 4, we present the results from the comparison. On average, we observe that DISCERN beats the keyword-based approach with statistical significance in the AGNews dataset. This further underscores the advantage of free-form language descriptions of underperforming clusters.

[10]The x-axis varies based on the size of the unlabeled pool for each dataset and the number of labeled examples identified by the description-based methods.

Dataset → # Aug. Ex.	TREC (2000) 500	TREC (2000) 1000	AGNews (1500) 500	AGNews (1500) 1000	Covid (4000) 500	Covid (4000) 1000
Base	58.48		75.8		47.68	
No Exp.	$77.09_{(2.18)}$	$78.04_{(1.74)}$	$80.03_{(1.51)}$	$80.68_{(1.08)}$	$51.07_{(0.66)}$	$48.08_{(1.14)}$
DEIM*	$71.91_{(4.20)}$	$75.69_{(3.14)}$	$\mathbf{80.52_{(0.77)}}$	$80.76_{(1.33)}$	$51.16_{(0.34)}$	$\mathbf{49.29_{(0.80)}}$
DISCERN-F	$76.99_{(2.49)}$	$78.98_{(1.88)}$	$79.75_{(1.25)}$	$80.96_{(1.98)}$	$51.12_{(0.64)}$	$48.60_{(1.36)}$
DISCERN	$\mathbf{77.20_{(1.80)}}$	$\mathbf{79.21_{(1.53)}}$	$80.39_{(1.35)}$	$\mathbf{83.44_{(1.00)}}^{\dagger}$	$\mathbf{51.55_{(0.47)}}$	$49.06_{(0.79)}$

Table 4: Accuracy of `distilbert-base-uncased` classifier after augmenting the training set with examples that have been generated using different approaches. Numbers in brackets next to dataset names indicate the number of training examples used for learning the initial classifier. **Bold** numbers indicate the best average classifier accuracy across five runs. † indicates statistically significant improvement over other approaches using t-test.

Language descriptions facilitate a more effective and efficient understanding of biases among users. Here we explore how language descriptions help users understand biases in classifiers and identify likely misclassified instances. This concept aligns with simulatability from previous explainability research (Hase and Bansal, 2020; Menon et al., 2023), which assesses users' comprehension of classifier predictions.

To do this, we conduct a user study, in which users are shown examples or DISCERN descriptions of clusters where the classifier has a higher misclassification rate than its base rate. Based on the information provided, users are tasked with identifying if new examples, drawn from one to two erroneous clusters per dataset, match the characteristics of given descriptions or cluster exemplars. The test uses new examples from within and outside the erroneous clusters, the latter having a high BERTScore (Zhang* et al., 2020) similarity with at least one example in the cluster. Participants provided predictions for six new examples in each HIT, and we measured the accuracy of predicting examples belonging to the erroneous cluster. 24 workers took part in this study conducted on Prolific and were compensated at $12/hr.

Method	Acc. (↑)	Time (↓)	Help. (↑)
No Desc.	62.5%	185s	3.00
DISCERN	**79.2%**	**177s**	**3.83***

Table 5: User evaluations in understanding classifier biases based on cluster exemplars (No Desc.) vs DISCERN descriptions.* = marginal statistical significance with t-test (p-value < 0.1).

In our results (Table 5), show that after reviewing DISCERN descriptions, users accurately predict new examples that exhibit characteristics similar to those in the erroneous clusters in 79.2% of cases, compared to 62.5% without descriptions. Further, users provided with descriptions required less time to perform the task and found them more helpful. These findings underscore the potential of language descriptions in enhancing users' understanding of systematic biases in classification models, a crucial step towards designing fairer and equitable models for real-world deployment.

6 Ablations

Impact of Embeddings used during Clustering. We examine the impact of different embeddings on the initial stage of the DISCERN framework, specifically focusing on clustering datapoints in the validation set. We compare the OpenAI embeddings Ada and v3 here for the TREC dataset. As shown in Table 6, across all methods, utilizing the v3 embedding consistently yields higher accuracy compared to Ada. This finding underscores the importance of choosing effective embeddings for identification of biases using our framework. Additionally, we observe that DISCERN outperforms the naive augmentation (No Descriptions) baseline even when employing weaker embeddings, highlighting the versatility of our framework.

Method	Ada	v3
No Descriptions	65.76	**78.04**
DISCERN-F	67.61	**78.98**
DISCERN	68.18	**79.21**

Table 6: Classiifer accuracies post synthetic data augmentation using different embeddings to cluster validation set datapoints on the TREC dataset.

Stronger Explainers enhance Classifier Performance. Table 7 evaluates the impact of different

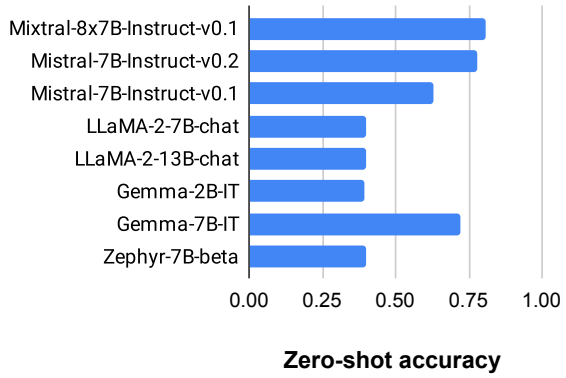

Figure 4: Zero-shot performance of different language models used as predicate evaluators for our task.

language models used for describing underperforming clusters and their subsequent classifier improvement. Specifically, it compares the accuracy of a `distilbert` classifier trained using cluster descriptors derived from two distinct language models: `gpt-3.5-turbo-0125` and `gpt-4-0125-preview`. We compute classifier improvements by adding 1000 synthetic instances generated using the different descriptions. Both DISCERN-F and DISCERN show marked improvements across datasets, highlighting the potential for improvement of our approach with larger and more capable language models.

`distilbert` accuracy with explainer LLM changing from `gpt-3.5-turbo` → `gpt-4-turbo`

Method	TREC	AG News
Base	58.48	75.8
DISCERN-F	78.98 → 81.12	80.96 → 86.44
DISCERN	79.21 → 79.62	83.44 → 86.85

Table 7: Accuracies post synthetic data augmentation using different language models for describing clusters.

Predicate Evaluators. The predicate evaluator in our framework provides signal to the explainer about the alignment of the generated explanations with the examples. Hence, we need the predicate evaluator to accurately predict whether a predicate applies to a given example. For this evaluation, we sampled 10 clusters from our three datasets, along with a random collection of datapoints from within and outside these clusters. We obtain the "ground-truth" annotations for the alignment between an explanation and a datapoint using `gpt-4-turbo` (Achiam et al., 2023)[11]. In Figure 4, we evaluate the performance of different open-source LLMs for this task. We observe that `Mixtral-8x7B-Instruct` has the highest agreement with the ground-truth. Consequently, we used it for evaluations in our experiments.

[11] GPT-4 judgments, found to align well with human judgments, serve as our proxy (Rafailov et al., 2023).

7 Conclusion

In this work, we propose a framework DISCERN, to address systematic biases and improve the performance of text classifiers. Using large language models to generate precise natural language descriptions of errors, DISCERN surpasses example-based augmentation techniques to identify and rectify systematic biases in multiple classifiers across diverse datasets. Through extensive experimentation, we have demonstrated the capability of DISCERN in reducing misclassification rates and improving classifier accuracy, consistently outperforming alternative approaches. Further, our human evaluations indicates user preference for understanding bias using natural language descriptions. Overall, our findings underscore the potential of DISCERN as a powerful tool to improve the performance of text classifiers, thus enabling the design of more reliable and equitable machine learning systems in various domains. Building on our results, future research directions can explore ways to enhance other applications using the refinement approach used in our work, integrate DISCERN into informing training recipes (such as, large language model training), and investigate biases transferred between different classifiers.

Limitations

The exact instantiation of our framework in this work makes use of proprietary large language models. The accessibility of these models is contingent upon evolving corporate policies of the respective entities. Nevertheless, we believe that with the increasing capabilities of smaller open-source large language models such as `Mixtral`, we should be able to achieve very similar performance with newer models while being accessible to everyone.

Our agglomerative clustering approach also depends on the distance threshold hyperparameter which can affect the granularity of the explanations. Future work can look into top-down approaches that can explain classifier biases at the right levels of granularity, thereby enabling interpretability and advancing the efficacy of our framework.

Acknowledgements

We would like to thank Kerem Zaman, Jack Goldsmith, Anika Sharma, and the anonymous reviewers for feedback and suggestions on the draft. This work was supported in part by NSF grant DRL2112635. The views contained in this article are those of the authors and not of the funding agency.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1962–1981.

Sabri Eyuboglu, Maya Varma, Khaled Kamal Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Re. 2022. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online. Association for Computational Linguistics.

Zexue He, Marco Tulio Ribeiro, and Fereshte Khani. 2023. Targeted data generation: Finding and fixing model weaknesses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8506–8520, Toronto, Canada. Association for Computational Linguistics.

Wenyue Hua, Lifeng Jin, Linfeng Song, Haitao Mi, Yongfeng Zhang, and Dong Yu. 2023. Discover, Explain, Improve: An Automatic Slice Detection Benchmark for Natural Language Processing. *Transactions of the Association for Computational Linguistics*, 11:1537–1552.

Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. 2023. Distilling model failures as directions in latent space. In *The Eleventh International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Nicolas Kayser-Bril. 2020. Google apologizes after its vision ai produced racist results. *AlgorithmWatch. Retrieved August, 17:2020.*

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Yoonho Lee, Michelle S Lam, Helena Vasconcelos, Michael S Bernstein, and Chelsea Finn. 2024. Clarify: Improving model robustness with natural language corrections. *arXiv preprint arXiv:2402.03715*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International*

Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.

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

Rakesh Menon, Kerem Zaman, and Shashank Srivastava. 2023. MaNtLE: Model-agnostic natural language explainer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13493–13511, Singapore. Association for Computational Linguistics.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Nazneen Rajani, Weixin Liang, Lingjiao Chen, Margaret Mitchell, and James Zou. 2022. SEAL: Interactive tool for systematic error analysis and labeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 359–370, Abu Dhabi, UAE. Association for Computational Linguistics.

Marco Tulio Ribeiro and Scott Lundberg. 2022. Adaptive testing and debugging of NLP models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3253–3267, Dublin, Ireland. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Adi Robertson. 2024. Google apologizes for 'missing the mark' after gemini generated racially diverse nazis. *The Verge*.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Chandan Singh, Aliyah R Hsu, Richard Antonello, Shailee Jain, Alexander G Huth, Bin Yu, and Jianfeng Gao. 2023. Explaining black box text modules in natural language with language models. *arXiv preprint arXiv:2305.09863*.

Nimit Sohoni, Jared Dunnmon, Geoffrey Angus, Albert Gu, and Christopher Ré. 2020. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352.

Chloe Rose Stuart-Ulin. 2018. Microsoft's politically correct chatbot is even worse than its racist one. *Quartz Ideas, 31*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Data Tattle. 2020. Covid Tweets Dataset (Retrieved February, 2024).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vijay Vasudevan, Benjamin Caine, Raphael Gontijo-Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. 2022. When does dough become a bagel? analyzing the remaining mistakes on imagenet. In *Advances in Neural Information Processing Systems*.

Chenxi Whitehouse, Monojit Choudhury, and Alham Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

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

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Appendix

In the Appendix, we provide details regarding the compute and training hyperparameters for our experiments (Section A), prompt templates (Section B), discuss extended related work (Section C), provide additional analysis (Section D), and show example templates used in human evaluation (Section E).

A Training Details

In Table 8, we detail the hyperparameters used for fine-tuning the `distilbert-base-uncased` and `roberta-large` models. We maintain the same hyperparameters during re-training of the model using the augmented training set. We implement these classifiers in Pytorch (Paszke et al., 2019) using the Huggingface library (Wolf et al., 2020). Classifiers were fine-tuned with full precision on a single NVIDIA A100-PCIE-40GB GPU, 400GB RAM, and 40 CPU cores.

Hyperparameters	Values
train_batch_size	32
eval_batch_size	512 (`distilbert`) 64 (`roberta`)
gradient_acc._steps	1
learning_rate	1e-5 (`distilbert`) 2e-5 (`roberta`)
weight_decay	0.01
adam_beta1	0.9
adam_beta2	0.999
adam_epsilon	0.0
max_grad_norm	1.0
num_train_epochs	3
lr_scheduler_type	linear
warmup_ratio	0.0
warmup_steps	600
seed	42
optim	adamw_torch

Table 8: Hyperparameters used for fine-tuning pre-trained models used across different datasets.

In Table 9, we report the hyperparameters for the cluster description generation and synthetic data augmentation. Using the agglomerative clustering algorithm and our preset distance thresholds, the number of examples in a cluster typically varies between 10-60. We use the OpenAI API to make calls to the GPT-3.5 and GPT-4 models. We also load the `Mixtral` model with 16-bit precision (bfloat16). The same system configuration, as used for classifier training, is used for these experiments.

In Algorithm 1, we summarize the explain and refine iterative setup used in DISCERN.

Note: Since the submission of this work and its eventual acceptance, the codebase for the Mistral tokenizer has been **modified** in a way that irreversibly affects its functionality. Unfortunately, due to these changes, it is no longer possible to load or utilize the original version of the `transformers` (original: `4.38.0`) that was used in the experiments described in this paper, resulting in differences in numbers.

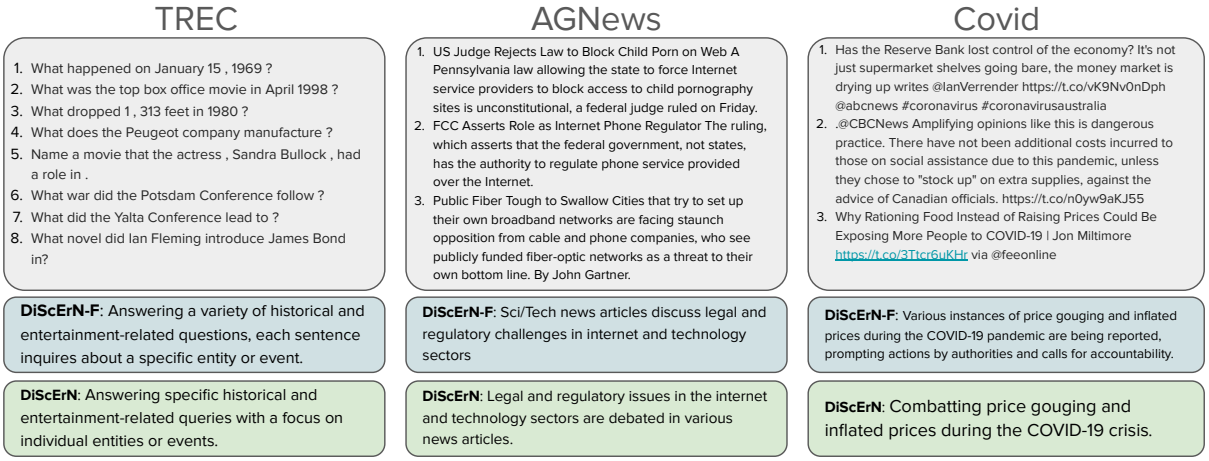

Figure 5: Descriptions generated using DiScErN-F and DiScErN for erroneous clusters in different datasets using the `distilbert-base-uncased` classifier.

Hyperparameters	Values
Clustering Alg.	Agglomerative Clustering
Distance Threshold	2 (`openai-v3`) 1.2 (`openai-ada`)
Explainer LLM	`gpt-3.5-turbo-0125`
Explainer Temperature	0.1
Explainer top-p	1
Max explainer generation tokens	512
In-cluster description threshold	0.8
Out-of-cluster description threshold	0.2
Num. In-cluster Examples in Prompt	64
Num. Out-of-cluster Examples in Prompt	32
Evaluator LLM	`Mixtral-8x7B Instruct-v0.1`
Max evaluator generation tokens	1
Evaluator precision	bfloat16
Data generator LLM	`gpt-3.5-turbo-0125`
Generator Temperature	0.7
Generator top-p	1
Generator seed	0
Max generator tokens	4096
Max generated examples (per cluster)	100

Table 9: Hyperparameters used for generation and refinement of cluster descriptions + synthetic data augmentation.

Objective	Reference
Predicate Generation	Table 11
Predicate Refinement	Table 12
Example Evaluation	Table 13
Data Generation - Examples	Table 14
Data Generation - Explanations	Table 15

Table 10: Legend for prompts used in the various stages of DiScErN.

B Prompt Templates

In this section, we present the prompt templates used during the different stages of our framework. The table below provides a legend to the exact prompts used for each scenario.

C Extended Related Work

Model Explainability. Explainability methods aim to uncover the relevant features that influence model predictions. The majority of works in this area emphasize local explanations of model predictions (Sundararajan et al., 2017; Ribeiro et al., 2016). Although local explanations help to understand model behavior on specific instances, they do not provide a global understanding of model behavior. More recently, Singh et al. (2023); Menon et al. (2023) proposed approaches to provide language explanations for the behavior of a model. However, they generate explanations from a restricted set of features, either n-grams or tabular features.

```
Here are a group of sentences:

{samples_in_prompt}

Generate a single-line predicate description that incorporates the specific word
or label '{label}'.

Your response should be formatted in the following manner:
Thoughts:
1. The sentences are mainly <type of sentences>.
2. The sentences talk about <topic>.
3.  I will also focus on the following attributes about the sentences in the
generated predicate to be precise: <list of attributes>
PREDICATE:
- ''<predicate>"

Try to make sure that the generated predicate is precise and will only satisfy
the examples mentioned above.

Thoughts:
```

Table 11: Prompt used to elicit the first set of explanations given cluster examples alone.

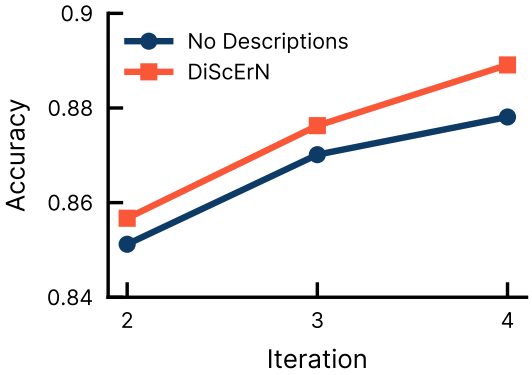

Figure 6: Average `distilbert-base-uncased` accuracy when successively improved with the application of DISCERN and the naive augmentation (No Desc.) baseline. Remarkably, the enhancement in classifier performance achieved by the No Desc. baseline in four iterations is attainable with DISCERN in merely three iterations.

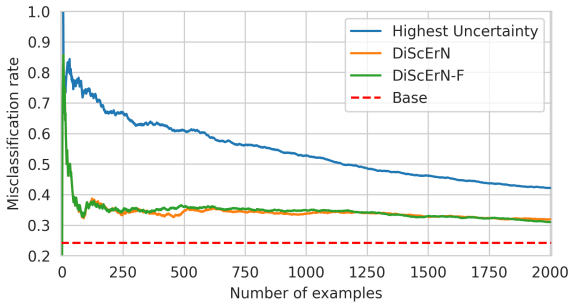

Figure 7: Misclassification rate of the top-K examples as sorted based on embedding match.

D Additional Analysis

Successive use of DISCERN enhances the classifier even more. We investigate whether the performance of a text classification model can be improved through the successive application of our framework, DISCERN, over multiple iterations. We hypothesize that by iteratively refining the model using the explanations provided by DISCERN, and then reevaluating the model's performance, we can achieve further improvements in the classifier's accuracy. In Figure 6, we present the results of applying this iterative process over four successive rounds. The figure shows that through the repeated use of our framework and also the naive augmentation approach, the performance of the classifier continues to increase with each iteration, demonstrating the its effectiveness. Interestingly, the enhancement in classifier performance achieved by the naive augmentation baseline in four iterations is attained with DISCERN in merely three iterations.

Can DISCERN explanations identify the most important examples? In Section 5, we present a novel active learning approach that leverages an

```
You were asked to provide a single-line predicate description for a set of
examples (let's call this CLUSTER_1) shown below:

{samples_in_prompt}

You generated the following description: ``{description}"

This description satisfied the following examples:

{in_cluster_satisfied_examples}

However, the description also identifies with the following examples (that it
should not ideally) (let's call this CLUSTER_2 examples):

{out_of_cluster_satisfied_examples}

In other words, the current description explains {pass_rate:.1f}

Please re-write the description that explain only examples from CLUSTER_1 while
excluding examples from CLUSTER_2.

Try to make descriptions simple and general. For example, you could focus on
the syntax, topic, writing style, etc.
First, for the failing description above, explain why the description does not
accomplish the goal of describing only the examlpes in CLUSTER_1. Output this
reasoning as:
Thoughts:
1. The examples in CLUSTER_1 and CLUSTER_2 talk about one common topic: {label}.
2. The examples in CLUSTER_1 emphasize on <CLUSTER_1 description>.
3. Whereas, the examples in CLUSTER_2 emphasize on <CLUSTER_2 description>.
4. The previous description failed because <reason>.
5.  The examples in CLUSTER_2 are about "<reason>" which is not present in
CLUSTER_1. I will focus on mentioning this reason in the new predicate.
Then output the description so that it explains only examples in CLUSTER_1, using
the following format:
NEW PREDICATE:
- ``<more precise-yet-simple CLUSTER_1 description that highlights difference
with CLUSTER_2>"

Note: The new predicate has to be strictly different from the previous one.
Note: Do not mention the words CLUSTER_1 or CLUSTER_2 in your new predicate. It
should be part of your thought process however.


Thoughts:
1. The examples in CLUSTER_1 and CLUSTER_2 talk about one common topic: {label}.
```

Table 12: Prompt used to refine explanations given in-cluster examples and out-of-cluster examples.

```
Check if this statement '{example}' satisfies the given condition:
'{description}'. Provide only 'Yes' or 'No'. When unsure, respond with 'No'.
```

Table 13: Prompt used to check alignment of example with the generated description.

```
In this task, you will be shown some examples sentences that share some property.
Your task is to generate 100 more diverse examples that satisfy the shared
property of these texts.

The examples you generate should follow the style and content of the examples
mentioned below:
{list_of_examples}

Consider the linguistic style, content, length, and overall structure of the
provided examples. Your generated examples should resemble the provided set in
terms of these aspects. Aim to produce sentences that convey similar information
or ideas while maintaining consistency in tone, vocabulary, and grammatical
structure.

Feel free to vary the details and specifics while ensuring that the generated
examples capture the essence of the provided set. Pay attention to context,
coherence, and any relevant patterns present in the examples to produce outputs
that closely align with the given set.

Your response:
-
```

Table 14: Prompt used to generate synthetic instances for the classification task using only cluster exemplars.

```
In this task, you will be shown some examples sentences that share a property
given by the predicate below. Your task is to generate 100 more diverse examples
that satisfy the predicate.

Predicate: {predicate}

The examples you generate should follow the style and content of the examples
mentioned below:
{list_of_examples}

Consider the linguistic style, content, length, and overall structure of the
provided examples. Your generated examples should resemble the provided set in
terms of these aspects. Aim to produce sentences that convey similar information
or ideas while maintaining consistency in tone, vocabulary, and grammatical
structure.

Feel free to vary the details and specifics while ensuring that the generated
examples capture the essence of the provided set. Pay attention to context,
coherence, and any relevant patterns present in the examples to produce outputs
that closely align with the given set.

Your response:
-
```

Table 15: Prompt used to generate synthetic instances for the classification task using descriptions.

instruction-tuned model to identify whether an example complies with a given instruction through a binary classification task. Here, we explore an alternative retrieval-style method that utilizes large language model (LLM) embeddings to identify examples that are most likely to be misclassified. Specifically, we employ the LLM2Vec embeddings (BehnamGhader et al., 2024) to compute the semantic similarity between the instruction descriptions (queries) and the unlabeled examples (passages) in the dataset. By sorting the examples in decreasing order of their cosine similarity to the descriptions, we are able to plot the misclassification rate for the top-K examples as the value of K is varied.

The results presented in Figure 7 demonstrate that our retrieval-based approach, using both DISCERN and DISCERN-F descriptions, is capable of identifying examples that achieve a higher misclassification rate compared to randomly selecting examples. This finding suggests that our method is effective in identifying potentially erroneous examples. Interestingly, we observe that the examples selected based on the highest classifier uncertainty are consistently more challenging and exhibit a significantly higher misclassification rate. These insights motivate future research directions that explore approaches to more consistently and effectively select harder subpopulations of data, potentially outperforming the highest classifier uncertainty approach.

Extending to more recent LLMs. In Section 6 and Table 7, we demonstrate improvements in classifier performance facilitated by descriptions derived from the more stronger `gpt-4-turbo` model. Building upon this analysis, we extend our investigation to include the more recent `4o` model series, assessing their influence on the performance of the `distilbert-base-uncased` classifier across the AG-News and TREC datasets.[12] As can be observed in Figure 8, using the newer variants in `gpt-4o` and `chatgpt-4o-latest`, we can obtain marked improvements over the `gpt-3.5-turbo` model. However, utilizing `gpt-4o-mini` results in slightly lower performance of the classifier in AG-News. This underscores the importance of using the strongest variants of language models rather than their distilled counterparts. Taken together, this experiments points to the potential of stronger language models contributing more accurate repre-

[12]Note: these experiments were run after the version change mentioned in Appendix §A.

Dataset →	TREC (1500)		AG News (500)		Covid (4000)	
# Aug. Ex.	500	1000	500	1000	500	1000
Base	70.85		41.2		55.1	
No Desc.	$71.99_{(14.04)}$	$85.26_{(3.13)}$	$\mathbf{58.88_{(10.57)}}$	$61.96_{(9.47)}$	$50.76_{(1.42)}$	$46.45_{(3.07)}$
DISCERN-F	$72.25_{(11.95)}$	$86.51_{(1.98)}$	$58.60_{(10.35)}$	$64.28_{(10.67)}$	$\mathbf{52.35_{(1.48)}}$	$\mathbf{48.08_{(1.39)}}$
DISCERN	$\mathbf{78.11_{(3.38)}}$	$\mathbf{88.54_{(0.90)}}$	$55.44_{(14.50)}$	$\mathbf{67.00_{(10.39)}}$	$51.75_{(1.76)}$	$47.64_{(0.91)}$

Table 16: Accuracy of `roberta-large` classifier after augmenting the training set with examples that have been generated using different approaches. Numbers in brackets next to the names of the dataset indicate the number of training examples used to learn the initial classifier. **Bold** numbers indicate the best average classifier accuracy across five runs.

Algorithm 1 DISCERN

Require: Explainer LLM $\mathcal{E}$
Require: Evaluator LLM P
Require: Data Generator LLM $\mathcal{D}_{\mathcal{G}}$
Require: validation dataset $\mathcal{D}_{val} = (X_{val}, Y_{val})$
Require: classifier f

// **Get validation set predictions**
$Y_{pred} = f(X_{val})$
$\mathcal{Y} = \text{UNIQUE}(Y_{val})$
// **Cluster $\mathcal{D}_{val}$ for each class**
$X_{val,y} \leftarrow \{x : (x, y') \in \mathcal{D}_{val}, y = y'\}, \forall y \in \mathcal{Y}$ // Split dataset based on ground-truth label
$\mathcal{C}_{1:m,y} \leftarrow \text{AGGLCLUSTERING}(X_{val,y}), \forall y \in \mathcal{Y}$
for $c \in \mathcal{C}_{1:m,y}$ **do**
    iterations = 0
    $c_{out} = \{\}$
    while not refinement_threshold_met **or** iterations $<$ max_iterations **do**
        if iterations > 0 **then**
            $c_{out} = \text{SAMPLEEXAMPLES}(\mathcal{C}_{1:m,y} - \{c\})$ // sample out of cluster examples
        end if
        $e_c \leftarrow \mathcal{E}(c, c_{out})$
        // in-cluster evaluation
        $r_{in\_cluster} = \text{P}(c, e_c)$
        // out-of-cluster evaluation
        $r_{out\_cluster} = \text{P}(\mathcal{C}_{1:m,y} - \{c\}, e_c)$
        refinement_threshold_met = ($r_{in\_cluster}$ $>$ in-cluster threshold) **and** ($r_{out\_cluster}$ $<$ out-cluster threshold)
        iterations = iterations + 1
    end while
    // Generate data using the explanation
    $X', y' \leftarrow \mathcal{D}_{\mathcal{G}}(e_c, c)$
    X_{train}, y_{train}.append(X', y')
end for
$\text{RETRAINCLASSIFIER}(f, X_{train}, Y_{train})$

sentations of systematic bias in text classifiers.

E Human Evaluation Templates

In Figure 9, we provide screenshots of the templates used for human evaluation.

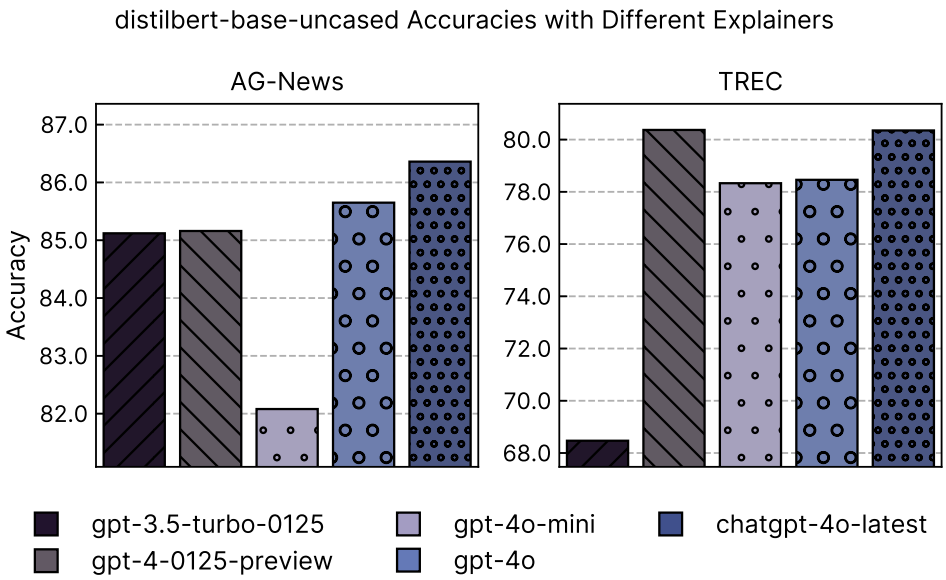

Figure 8: Accuracies post synthetic data augmentation using different language models for describing clusters. 1000 examples were augmented to the classifier based on each explanations. Results show the mean across five runs.

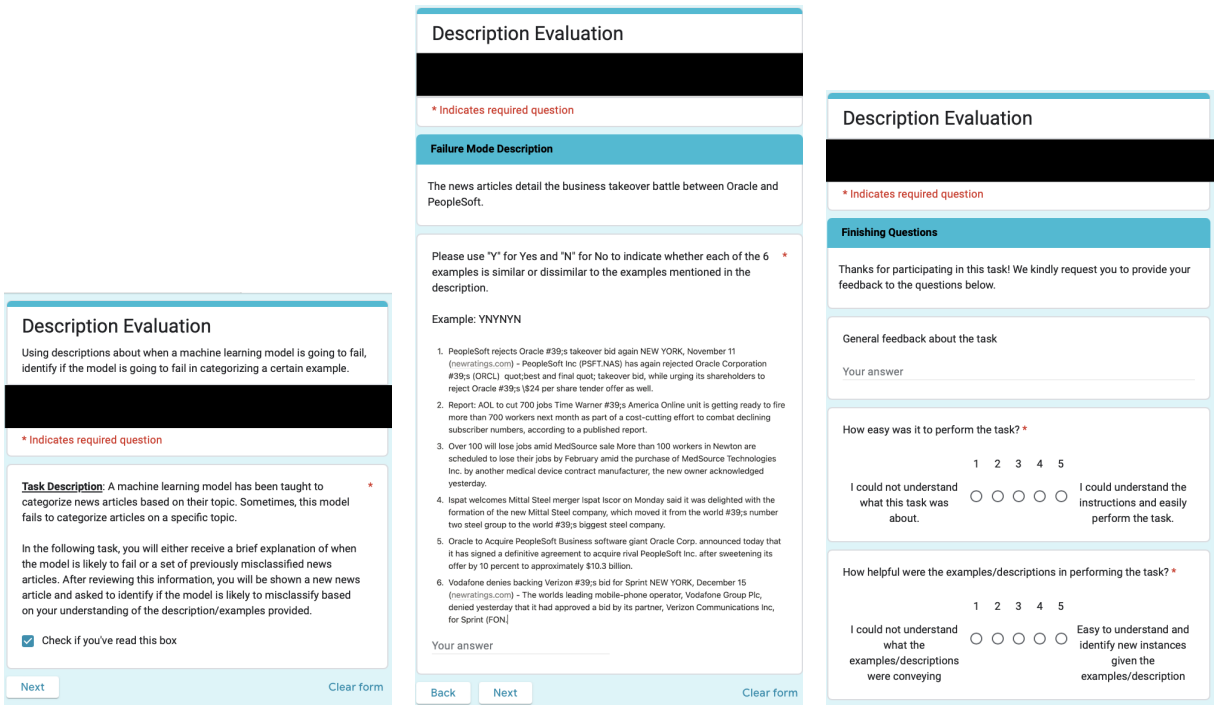

Figure 9: Example templates used for human evaluation of cluster descriptions and examples of the AGNews dataset in Section 5.