ARISE: Iterative Rule Induction and Synthetic Data Generation for Text Classification

Yaswanth M[♡0], **Vaibhav Singh**[♠0], **Ayush Maheshwari**[◊*],
Amrith Krishna[♣], **Ganesh Ramakrishnan**[♣♠]
[♠]Indian Institute of Technology Bombay, [♣]BharatGen
[♡]Accenture, [◊]NVIDIA

Abstract

We propose ARISE, a framework that iteratively induces rules and generates synthetic data for text classification. We combine synthetic data generation and automatic rule induction, via bootstrapping, to iteratively filter the generated rules and data. We induce rules via inductive generalisation of syntactic n-grams, enabling us to capture a complementary source of supervision. These rules alone lead to performance gains in both, in-context learning (ICL) and fine-tuning (FT) settings. Similarly, use of augmented data from ARISE alone improves the performance for a model, outperforming configurations that rely on complex methods like contrastive learning. Further, our extensive experiments on various datasets covering three full-shot, eight few-shot and seven multilingual variant settings demonstrate that the rules and data we generate lead to performance improvements across these diverse domains and languages.

1 Introduction

Large language models (LLMs) have facilitated the generation of high-quality synthetic data that often supplement available training data (Lin et al., 2023) or even surpass crowd-sourced annotations (Gilardi et al., 2023; Alizadeh et al., 2023). However, concerns of limited variance in such exemplars, leading to model collapse (Shumailov et al., 2023) or the failure to capture the tail of the true underlying distribution (Ding et al., 2024), remain. Similarly, forming multiple views of the available data by inducing rules, as a complementary source of supervision has shown to benefit various NLP tasks, including text classification (Maheshwari et al., 2021; Dong et al., 2022). In this work, we propose ARISE, a bootstrapping approach to iteratively refine synthetically generated exemplars and automatically induced rules, resulting in high quality entries with respect to a given classification task (Yarowsky, 1995; Varma and Ré, 2018).

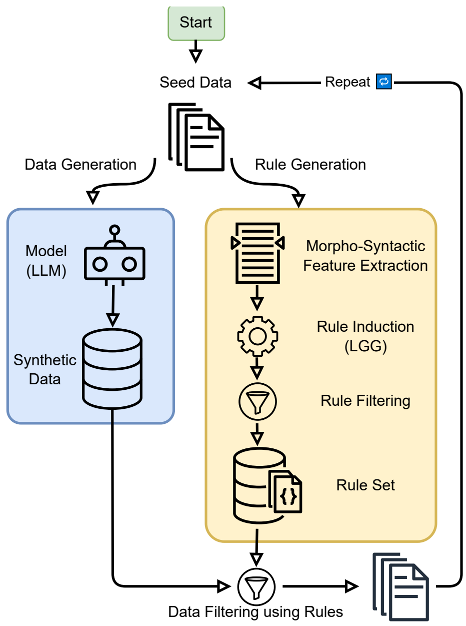

Figure 1: Overview of ARISE (**A**utomatic **R**ule **I**nduction using **S**yntactic tree g**E**neralization).

Figure 1 provides an overview of ARISE. We start by using available training data as our seed. Using LLMs, we leverage in-context learning (ICL), with the seed as input to synthetically generate candidate exemplars (Liu et al., 2022). Similarly, we generate rule candidates, via inductive generalisation using least general generalization (LGG) (Plotkin, 1971; Raza et al., 2014) by extracting syntactic n-grams from the seed. Further, the induced rules are then filtered using a submodular graph cut-based function (Bajpai et al., 2024; Kothawade et al., 2022). The exemplars and the rules we generate are task-specific and each exemplar and rule is associated with a label. Newly

[*]Work done while at IIT Bombay, with BharatGen.
[0]Correspondence to: Yaswanth M <yaswanthm03@gmail.com> and Vaibhav Singh <singhvaibhav@cse.iitb.ac.in>

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6434–6449
April 29 - May 4, 2025

generated exemplars are filtered using rules that are generated from the already validated seed. These filtered exemplars are then added to the seed for the next iteration. Iteratively, we induce rules from synthetically generated data and use the induced rules for data filtering.

In ARISE, we boost supervision signals in two ways. With synthetic data generation we supplement the available training data (Ratner et al., 2017; Pryzant et al., 2022). First, with rule induction, we obtain complementary signals that need not be explicitly captured from the existing data (Maheshwari et al., 2021; Singhal et al., 2023). Second, our rules are induced as generalized syntactic n-grams. Here, we aim to potentially capture morpho-syntactic information from the data, a view of data that need not be explicitly captured by state-of-the-art (SotA) systems in use. A classical NLP pipeline typically represents a string at multiple levels of abstraction which includes Part-of-Speech (PoS) tags, syntactic relations, *etc.* (Manning et al., 2014). ARISE uses higher-order dependency structures as features and generalizes over these features using inductive generalization (Popplestone, 1970) to induce the rules as generalized syntactic n-grams.

We find applicability of both the rules and exemplars from ARISE, with consistent performance gains in various text classification setups. Specifically, we experiment with ICL and fine-tuning setups. In ICL, we focus on long-context ICL (Li et al., 2024; Bertsch et al., 2024) and use the generated data as a pool from which exemplars are retrieved. Further, we incorporate our rules as explanations to the input and the exemplars. Similarly, we use the data for fine-tuning models, which include pre-trained LLMs, Qwen (Yang et al., 2024; Team, 2024) and RoBERTa (Liu et al., 2019).

We perform extensive experiments on multiple text classification datasets, which include three full-shot, and eight few-shot datasets from the FEWMANY benchmark (Yehudai and Bendel, 2024). Further, we perform multilingual experiments on seven languages using the MASSIVE (FitzGerald et al., 2022a) dataset.

Our major contributions are as follows:

- Use of rules and data from ARISE results in statistically significant gains in all the experimental setups, as compared to the corresponding configuration without resources from ARISE. Specifically, we obtain state of the art (SotA) results in our full-shot and few-shot experiments when using ARISE.

- The rules we generate are shown to be effective, both during ICL and fine-tuning. Further, using the rules as explanations under ICL for CDR dataset results in SotA results. Similarly, fine-tuning Qwen jointly with data and the augmented rules from ARISE has shown statistically significant improvements for Qwen and RoBERTa based models.

- Use of augmented data for few-shot setups in the FEWMANY benchmark demonstrate the quality of the augmented data we produce. We show that simply using additional data from ARISE, as low as 20-shot additional data per class, can result in improved performance than incorporating complex approaches such as contrastive representation learning into the training process.

- Our extensive experiments show that ARISE is generalizable across multiple domains and multiple languages. We report a 7.21% increase in performance, compared to the model without any resources from ARISE, averaged across seven different languages.

- We show that leveraging syntactic information as weak supervision for rule induction, brings a complementary source of supervision, which otherwise need not be captured by using string level data directly (§5.1).

2 ARISE - Automatic Rule Induction Using Syntactic Tree Generalization

Distributional hypothesis (Firth, 1957) is often realized using vector space models defined over a feature space (Turney and Pantel, 2010). Inputs can be encoded as dense contextualized vectors (Peters et al., 2018; Devlin et al., 2019) or a sparse semantic space consisting of lexical n-grams, syntactic n-grams (Goldberg and Orwant, 2013), higher order dependency features (Koo and Collins, 2010), or even graph motifs (Biemann et al., 2016).

We induce rules that can capture complementary information that is not explicitly captured in pre-trained neural models. Hence, we focus on incorporating structured grammatical information typically used in a traditional NLP pipeline (Manning et al., 2014) such as Part-of-Speech (PoS) and syntactic information. From dependency parses of

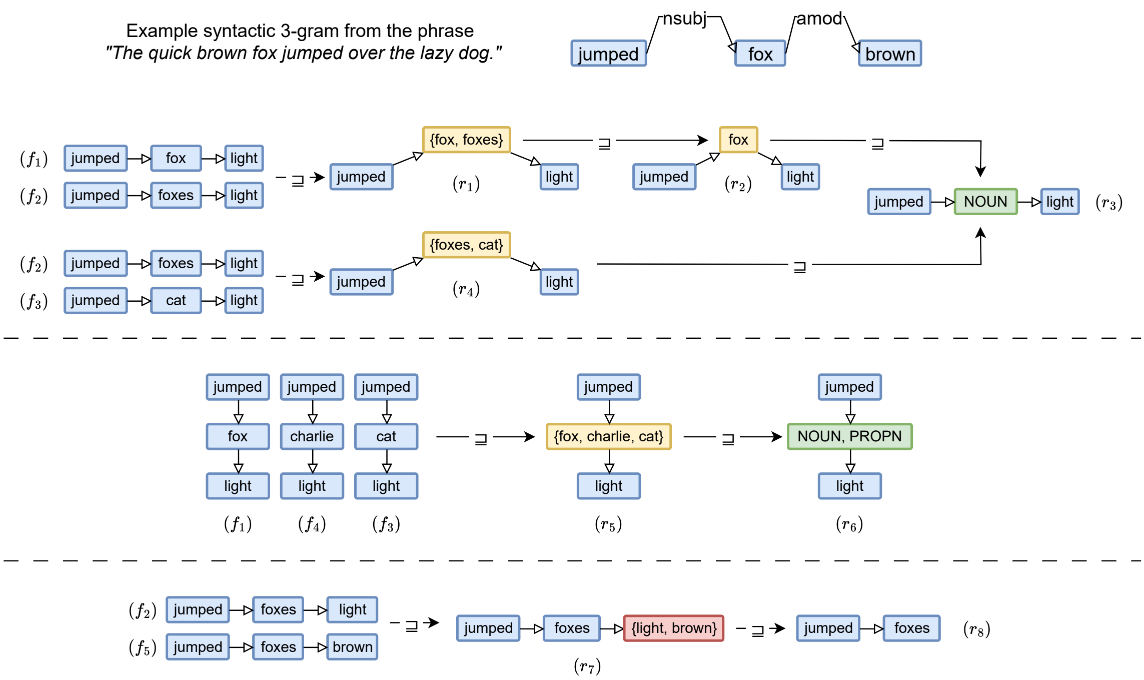

Figure 2: We induce rules via inductive generalization on syntactic n-grams, as shown (dependency relations omitted for brevety). The symbol '$\sqsupseteq$' denote a generalization operation. Trees labeled from f_1 to f_5 are instances of features. Similarly, trees labeled from r_1 to r_8 are rules.

input sentences, we extract induced subtrees as features. Each such feature is a syntactic n-gram, with the nodes as the words and the edges labeled with the dependency relations. We then induce rules via the inductive generalization of these features, using LGG (Raza et al., 2014; Thakoor et al., 2018).

For a text classification task with k labels, we assume the availability of a labeled dataset $\mathcal{D}$, where $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, x_i is an input document and $y_i \in \{l_1, l_2, ..., l_k\}$ is a label. We obtain sentence-level dependency parses for each $x_i \in \mathcal{D}$. A feature space $\mathcal{F}_{t=1}^{f}$ is defined over higher-order factorization of the dependency parses in $\mathcal{D}$. Each feature $f_t \in \mathcal{F}$ is an induced subtree of the parses for sentences in $\mathcal{D}$. In Figure 2, f_1 to f_5 denote instances of features in our feature space. These are syntactic n-grams extracted from sentence-level dependency parses of the input. A feature covers a set of documents in which that feature occurs at least once.

Rules are generalizations of features. Now, r_1 to r_8, in Figure 2, show various generalized rules induced from the features f_1 to f_5. If a generalized rule subsumes multiple features, then it covers a union of all the sets of documents corresponding to those features. Our rules are induced as the least general generalization (LGG) over a set of features (Plotkin, 1970, 1971). A feature can be a rule in itself, i.e. $\mathcal{F} \subseteq \mathcal{R}$, though it will be the most specific form of a rule. Previously, LGG was extensively used in information extraction (Califf and Mooney, 1997; Nagesh et al., 2012), program synthesis (Raza et al., 2014; Kitzelmann, 2010), and in several other areas of relational learning (Muggleton et al., 1992; Zelle et al., 1994).

We define two forms of generalizations for forming the rules, both structural and linguistic. If rule r_i is an induced subtree of r_j, then we can say that r_i is more general than r_j. Linguistic generalization include, substitution (Raza et al., 2014; Thakoor et al., 2018), of the nodes containing words with their corresponding stems, and PoS tags (Galitsky and Ilvovsky, 2019).

Figure 2 shows illustrative cases of generalization. Let us consider a corpus from which features f_1 to f_5 are extracted. Rules r_1 to r_7 show linguistic generalization. Similalry, rule r_8 shows structural generalization from r_7. Consider rules r_1, r_4, r_5 and r_7. These rules contain nodes with a group of words. Similarly, r_6 represents a rule that has a group of PoS tags in one of the nodes. In linguistic generalization, multiple trees are generalized to a single tree by grouping words or PoS that differ in these individual trees. Here, r_1 is a generalisation of f_1 and f_2. Similarly, r_4 is a generalization of f_2 and f_3.

We currently restrict the groupings at a node to be homogeneously typed, i.e. a set can either be that of inflected word forms, stems or of PoS tags, but not a mix of those. Further, the cardinality of such a group is set to an arbitrary upper bound, to avoid trivial generalisations. The rules we generate belong to $\mathcal{R}^r_{t=1}$. Here, for every input in $x_i \in \mathcal{D}$ it should either predict a label from $\{1, ..., k\}$, if the rule is applicable to the input. Else, it should abstain from making a prediction.

2.1 Rule Induction via LGG

We obtain features from dependency parses of the dataset $\mathcal{D}$. We consider only those subtrees that exactly have one of the six core dependency relations in them (de Marneffe et al., 2014; Nivre et al., 2020). These core dependency relations are direct or indirect objects, nominal or clausal subjects, as well as clausal or open clausal complements. We partition the features into six mutually exclusive subsets, with each subset corresponding to one of the core relations.

A complete lattice is constructed out of each partition, by adding a supremum and infimum element to the partition. Here, we add a rule '$* \xleftarrow{rel} *$', where 'rel' is the core-relation corresponding to the partition. It is the supremum for any element in the partition, as every element in the partition is subsumed by it and covers any document that has the relation present in it. We also define 'ϵ' as the infimum and it represents an empty rule that rules out any document in the input. The complete lattice provides a search space of rules over which the partial ordering is provided. Here, any two pair of subtrees have a least general generalization or a least upper bound (Raedt, 2010). In Figure 2, r_1 is the LGG of f_1 and f_2. r_1 represents all the sentences that either have f_1 or f_2 in their dependency parses. Similarly, r_2 and r_3 are also generalizations of f_1 and f_2, but not their LGG.

For every rule in the lattice, we compute its label-PMI vector, following Singhal et al. (2023) and Jin et al. (2022). Label-PMI vector is a vector of the pointwise mutual information scores of the rule corresponding to each label. From the vector, we consider its maximum score, denoted as L-PMI. The label corresponding to L-PMI is then assigned to the rule. From the lattice, we start bottom up and compute the LGG for every pair of rules. We induce the LGG as a rule, only if it has a higher L-PMI than the individual rules in the pair. These induced rules form our candidate set of rules. For a given label y_j, the pointwise mutual information for the rule r_t is given by,

$$\text{PMI}(y_j, r_t) = \frac{log|\mathcal{D}| \times \text{Count}(r_t, \mathcal{D}^{y_j})}{\text{Count}(r_t, \mathcal{D}) * |\mathcal{D}^{y_j}|}$$

Here, Count(a, b) implies the count of input documents having both a and b. Similarly, $\mathcal{D}^{y_j}$ implies the set of documents with the label y_j.

Text classification tasks need not always have single sentence inputs. We generally assume a feature may appear in any of the sentences in the input, unless these sentences clearly have a predefined role in the task. For instance, interchanging the premise and hypothesis in natural language inference (Berant et al., 2011; Dagan et al., 2010) generally leads to different outcomes. In such cases, we induce rules for premise sentences and hypothesis sentences separately. Further, the L-PMI is applied a second time, this time for a pair of rules, one induced from the premise and the other from the hypothesis. The 2-step L-PMI approach enables to reduce the combinatorial explosion which otherwise may happen, and is trivially extendable to tasks with multiple roles.

3 ARISE Framework

3.1 Rule Filtering

We induce rules from a set of input documents (§2), which are expected to be noisy. Two rules may even predict conflicting labels to a given input, akin to labeling functions in data programming (Ratner et al., 2017; Zhang et al., 2022a). Ideally, the final set of filtered rules needs to be accurate, diverse and high in coverage (Bajpai et al., 2024).

For rule filtering, we use the submodular graph-cut (GC) function (Kothawade et al., 2022), as proposed by Bajpai et al. (2024). Using GC, we select a final set of representative and diverse rules $\mathcal{R}_f$, from the set of candidate rules $\mathcal{R}$. For $\mathcal{R}_f \subseteq \mathcal{R}$, we define the GC function as:

$$f_{GC}(\mathcal{R}_f) = \sum_{r_i \in \mathcal{R}, r_j \in \mathcal{R}_f} s_{ij} - \lambda \sum_{r_i, r_j \in \mathcal{R}_f} s_{ij}$$

Here, $\lambda \in [0, 1]$ governs the diversity-representation trade-off, where higher λ implies higher diversity in $\mathcal{R}_f$. s_{ij} is the similarity score for rule pair r_i and r_j. It is calculated as the weighted sum of the precision, coverage, and agreement between the pair of rules:

$$s_{ij} = \alpha(r_i)+\alpha(r_j)+w*\beta(\{r_i,r_j\})+\gamma*\mu(r_i,r_j)$$

Here, $\alpha(r_i) = \text{Precision}(r_i)$, $\mu(r_i, r_j)$ is the agreement, calculated as the fraction of instances where both rules agree. $\beta(\{r_i, r_j\})$ is the coverage, calculated as the fraction of instances labeled by at least one of the rules.

Our objective function is $\max_{|\mathcal{R}_f|\leq k} f_{GC}(\mathcal{R}_f)$, where k is a fixed budget (Kothawade et al., 2022). We employ a greedy approach to choose a rule that maximizes the marginal utility $argmax_{r_i\in\{\mathcal{R}-\mathcal{R}_f\}} f_{GC}(\mathcal{R}_f \cup \{r_i\}) - f_{GC}(\mathcal{R}_f)$. Please note that Bajpai et al. (2024) starts with an empty set, whereas we start with the existing rule set obtained from the previous round of bootstrapping. One round of filtering is completed until the fixed budget k is exhausted.

3.2 Bootstrapping Rules and Synthetic Data

Varma and Ré (2018) previously employed a bootstrapping based rule induction approach for labeling available unlabeled data. In ARISE, combining synthetic data generation and automated rule induction presents us with an opportunity to bootstrap and expand our labeled dataset and rules iteratively.

We start our bootstrapping with the training split of the available gold labeled data as the seed, as shown in Figure 1. We synthetically generate new data for each class using prompt demonstrations, with the demonstrations retrieved from the seed set (Zhang et al., 2022b; Peng et al., 2024). Similarly, we perform rule induction (§2) from the seed. The induced rule candidates are then filtered using the validation split of the available gold data and added to the rule set (Figure 1). Similarly, data filtering for the synthetically generated exemplars is performed using the rule set. In data filtering, only those exemplars that match their generated label with the predicted label from the generative model are filtered. Finally, the seed set is expanded with the filtered data. The seed set and the rule set are expanded after every iteration of bootstraping.

Seed and validation set during few-shot: Zhu et al. (2023) observe that while weak supervision systems use limited training data, they heavily rely on the availability of a clean gold-labeled validation data for their performance gains. Hence, in our few-shot setups we do not use any validation split of the data and instead use only the few-shot training splits. Here, the gold labeled data, i.e. the few-shot training split, is used only as the validation data for rule filtering. Further, the initial iteration of synthetic generation happens in a zero-shot setup without demonstrations. Moreover, the rule candidates induced for the initial iteration is also from the synthetically generated samples, similar to subsequent iterations.

Data Augmentation: We use prompt demonstration, long context ICL or few-shot depending on the task setup, to synthetically generate new labeled sentences using LLMs. For each label, we sample m instances each of positive and negative samples from the seed set and then use it for generating new data samples (Smith et al., 2024; Lin et al., 2023). Our prompt demonstration approach includes label information, positive examples, and negative examples for synthetic generation. In addition to generating new data samples, we also perform paraphrasing of data samples in the seed set. By paraphrasing, we gain diverse syntactic structures for better rule induction.[1]

4 Experiments

Dataset: We use three datasets, namely, DISCOVERY (Sileo et al., 2019), CDR (Davis et al., 2017; Zhang et al., 2021) and ANLI (Nie et al., 2020), as shown in Table 1, for our full-shot setup. Here, we use the full training split, unless hit by an upper bound of 15,000 labeled instances, when fine-tuning the models. For ANLI, we focus on the R3 Dataset. CDR is a binary true/false classification problem for a given document with mentions of chemicals and diseases tagged. Similarly, ANLI is a 3-class, natural language inference task. DISCOVERY attempts at identifying the appropriate discourse marker from a set of 174 classes for a given pair of statements. For few-shot, we use the FEW-MANY Benchmark (Yehudai and Bendel, 2024), consisting of eight multiclass classification datasets. Here, we only use the 5-shot labeled data points from the training splits of the datasets involved. Finally, the multilingual experiments are performed using the MASSIVE dataset (FitzGerald et al., 2023) with intent classification as the task. Here, we use seven typologically diverse languages including Chinese, English, French, German, Hindi, Japanese, and Spanish.

[1]For more details, refer §A.3

Dataset	Train	Dev	Test	# Labels
DISCOVERY	1,566,000	87,000	87,000	174
ANLI	100,459	1,200	1,200	3
CDR	8,430	920	4,673	2
MASSIVE	11,514	2,033	2,974	60

Table 1: Dataset statistics of the original datasets. for ANLI and DISCOVERY, we perform class-wise stratified sampling and do not use more than 15,000 labeled instances from the training split for fine-tuning setups.

Data Generation: We use GPT-3.5, GPT-4, and Claude 3 Opus for synthetic data generation. We generate label-specific data by prompt demonstration. Here, Using Wu et al. (2023), we perform k-NN retrieval from the seed data, with $k = min(n, 150)$, where n is the available data for a given label in the seed for positive demonstrations, and and equal amount of randomly sampled out of class samples as negative examples (Bertsch et al., 2024; Liu et al., 2022). We use RoBERTa based sentence-embedding (Reimers and Gurevych, 2019) for sentence representation. For multilingual experiments, we experiment with *direct* generation of the synthetic data in the target language, and also via *translation* of synthetically generated English sentences. For translation, in addition to the three aforementioned LLMs we use NLLB-54B (Team et al., 2022) and Google Translate. For translation in Hindi, we use Gala et al. (2023).

4.1 Experimental Setup

We incorporate the rules and exemplars from ARISE in diverse text classification settings.

In-context learning (ICL): We experiment with three different configurations under long-context ICL using LLMs. One, is a **zero-shot** setup where we provide the input only with an explanation without any retrieved exemplars. Here, the explanations are obtained by phrasing the generated rules and their predictions as reasoning statements similar to the group of prompting techniques collectively referred to as thought generation prompting (Schulhoff et al., 2024). Two is the ***k*-shot** setup where we add prompt demonstrations from the generated data into the prompt. Third is the ***k*-shot-XP** setup where we append explanations in addition to the prompt demonstrations. For demonstrations, we provide rules leading to both correct and wrong label predictions for it, similar to that in Contrastive CoT (Chia et al., 2023). We reuse the retrieval setup used for data generation.

Fine-tuning (FT): We fine-tune an open-weight LLM (Team, 2024) and a smaller pre-trained LM (PLM, Liu et al., 2019; Conneau et al., 2020) in different configurations, under the **full-shot** setup. For fine-tuning the LLM, we employ PEFT using QLoRA. Our configurations for FT include; FT-base* where only the training data is used; FT-DA, where additional data from ARISE are used; FT-J where only the rules corresponding to the training data are incorporated via Joint Learning using SPEAR (Maheshwari et al., 2021) and finally, FT-JDA, where both the data and rules from ARISE is used. We also experiment with FT-JDX, a variation where the rules are incorporated both as part of the input prompt and via Joint Learning.

Joint Learning: For incorporating the rules into our fine-tuning process, we follow SPEAR (Maheshwari et al., 2021), a Joint Learning framework that learns over a feature-based classification model and a label aggregation (LA) model. The feature model is an LLM or a PLM and LA is a generative model (Chatterjee et al., 2020), learned via data programming, using the automatically induced rules as labeling functions.

LA is denoted as $P_\theta(\mathbf{l}_i, y)$, where $\mathbf{l}_i$ a vector that represents the firing of all LFs for an input $\mathbf{x}_i$. Each firing, l_{ij} can be either 0 (abstain) or class label k (Chatterjee et al., 2020). Our Joint Learning objective incorporates three different loss components for learning from labeled data. We provide a brief overview of each loss component below, while encouraging interested readers to Maheshwari et al. (2022) for detailed information. The first component of the loss is the standard cross-entropy loss for the model P^f_ϕ. The second component is the negative log-likelihood on the dataset. The third is the KL-Divergence between the predictions of the LA and P^f_ϕ models, which enforces consensus by aligning their predictions.

$$\min_{\theta,\phi} \sum_{i \in \mathcal{L}} L_{CE}\left(P^f_\phi(y|\mathbf{x}_i), y_i\right) + LL_s(\theta|\mathcal{L}) + \sum_{i \in \mathcal{L}} KL\left(P^f_\phi(y|\mathbf{x}_i), P_\theta(y|\mathbf{l}_i)\right)$$

Base Models: Our experiments are performed on one representative model for each of the following categories: (1) a closed-source LLM with access only via API, (2) an open-weight LLM, and (3) a pre-trained LM. Models like RoBERTa are still preferred in resource and latency conscious industry use cases, such as customer support, to larger models with few billion parameters. Factors include low latency, need for low hardware configuration and low cost, while still being competitive in several use cases. We choose GPT-4 Turbo (OpenAI et al., 2024), Qwen2.5-72B-Instruct (Team, 2024), and RoBERTa-large (Liu et al., 2019), respectively, based on their category-wise performance on preliminary experiments using B77 and AP106 datasets. For multilingual setup, we employ XLM-RoBERTa (Conneau et al., 2020) based on observations from (FitzGerald et al., 2022b). Additionally, we utilize Zhang et al. (CPFT, 2022c) for our contrastive learning (Chen et al., 2020) based baseline.

Accuracy is our primary evaluation metric. We use Dozat and Manning (2016), a dependency parser, to extract syntactic n-grams from input. We obtain induced subtrees of up to 3 nodes as rules. For few-shot, we perform all our experiments using 5 random splits and report the average (Yehudai and Bendel, 2024). For Joint Learning, we use 20% of the synthetically generated data as the validation split, while using all the gold data as the training data. For learning the parameters for our rule filtering step (§3.1), we use the few-shot gold data as validation. For full-shot setups we use the standard train-validation-test splits.[2]

5 Results

Use of data and rules from ARISE, results in statistically significant gains for all the datasets, both under full-shot and few-shot setups, including multilingual few-shot scenario. In full-shot setup, we outperform SotA models for both CDR, Discovery and ANLI, with more than an 8% (Zhao et al., 2024), 7%[3] (MTL; Sileo et al., 2019) and 18% (Kavumba et al., 2023) increase, respectively.

Table 2 shows the results for various configurations where data and rules from ARISE are used. FT-base* is the only configuration where no information from ARISE is used. FT-base* however is fine-tuned on the available training splits of the corresponding datasets. Qwen FT-JDX, the configuration that uses Joint Learning with rules, rules as explanations and augmented data reports the best results for ANLI and Discovery with an absolute gain of more than 30% and 3%, respectively. For Discovery, gains from rules are not significant, as Qwen FT-DA achieves comparable results to Qwen FT-JDX. Similarly, GPT 4 k-shot-XP achieves the best results for CDR, outperforming even the fine-tuned version with an absolute gain of roughly 10% as compared to Qwen FT-base*. Qwen FT-JDA and FT-JDX, both using Joint Learning with rules is the second best model. For CDR and ANLI, using both additional data and rules lead to statistically significant gains.[4]

Model	Configuration	CDR	ANLI	DISC.
GPT 4 (ICL)	zero-shot	85.89	79.53	2.34
	k-shot	88.95	81.59	31.10
	k-shot-XP	**92.13**	<u>86.78</u>	35.44
Qwen (ICL)	zero-shot	82.86	71.27	8.59
	k-shot	84.24	73.19	39.70
	k-shot-XP	86.84	84.05	47.36
Qwen (FT)	FT-base*	82.05	58.20	92.29
	FT-J	85.27	63.08	92.70
	FT-JXP	85.63	85.19	92.40
	FT-DA	87.76	75.69	95.33
	FT-JDA	<u>90.16</u>	78.30	<u>95.72</u>
	FT-JDX	90.08	**88.37**	**95.81**
PLM (FT)	FT-base*	81.78	53.82	90.60
	FT-J	84.72	57.78	90.88
	FT-JXP	84.58	57.85	90.67
	FT-DA	86.94	62.35	93.02
	FT-JDA	86.61	62.87	93.26
	FT-JDX	86.74	62.95	93.43

Table 2: Results in ICL and FT setups. Numbers in **boldface** and <u>underline</u> represent best and the second-best configurations, respectively. Here, PLM refers to RoBERTa-large. FT-base* is the only configuration that does not incorporate ARISE.

Qwen models benefit from Joint Learning for CDR and ANLI, even after when their results saturate with additional data (FT-JDA). Here, both additional data and rules have shown to benefit the models and bring in complementary supervision signals. However, Joint Learning does not lead to statistically significant gains, once fine-tuning is performed with additional data for RoBERTa (FT-DA vs. FT-JDA for RoBERTa)

Within ICL, GPT 4 outperforms Qwen in both CDR and ANLI, but Qwen outperforms GPT 4 in

[2]For the prompt and hyperparameter details, refer: https://sites.google.com/view/ariserules/

[3]Not a comparable model

[4]Statistical significance is performed by t-test ($p < 0.05$)

Dataset	Configuration	Vanilla	ARISE
CDR	zero-shot	74.56	**85.89**
	k-shot	83.43	**88.95**
	k-shot + Aug.	83.56	**88.95**
	k-shot-XP + Aug.	89.35	**92.13**
DISC.	zero-shot	0.84	**2.34**
	k-shot	8.73	**31.10**
	k-shot + Aug.	26.97	**31.10**
	k-shot-XP + Aug.	32.11	**35.44**

Table 3: ICL Experiments in GPT-4 that compares both ARISE, and ARISE-less scenarios under comparable conditions

Discovery. Discovery has a large label space of 174 labels, and these are common terms which are typically used as markers between two statements. Qwen being an open-weight model, we were able to constrain the output space using constrained decoding. While there exist similar approaches with structured output generation in GPT 4, we have limited control with GPT 4 compared to an open-weight model with constrained decoding.

Table 3 compares ICL results with both ARISE, and ARISE-less scenarios under comparable conditions for both CDR and DISCOVERY datasets. We observe gains with ARISE, for all the casess we compared. Here, *Vanilla zero-shot* does not use rules as explanations for the input from ARISE, whereas *Vanilla k-shot* retrieve examples only from the training split and uses no augmented data at all. *Vanilla k-shot + Aug.* uses data augmentation proposed by Lin et al. (2023). Finally, *k-shot-XP + Aug.* uses the above augmentation setting, but adds explanation from ARISE. We find consistent performance improvements in all the configurations when ARISE components are increasingly used.

5.1 Impact of Rules

We previously claimed that we obtain complementary supervision signals with rules compared to a setup like FT-base*. We validate the claim and observe statistically significant gains from the rules in ICL for all the three datasets, and in two of three datasets except for Discovery in fine-tuning.

Rules used for Joint Learning: FT-J and FT-JXP are two settings, in Table 2, which are trained jointly using the rules from ARISE, but only with the original training split. For CDR, FT-J results in a percentage increase of 3.92 and 3.59 for both Qwen and RoBERTa, respectively, compared FT-base*. Similarly for ANLI, we observe a percentage increase of 8.38 and 7.35, respectively, for Qwen and RoBERTa respectively. Moreover, ANLI reports statistically significant gains with Qwen for FT-JXP, XP implying rules used also as explanations, compared to FT-J. We did not see any additional gains with RoBERTa when adding rules as explanations. We hypothesize this is due to lack of instruction tuning.

For other datasets, FT-JXP configuration does not lead to statistically significant gains. For Discovery, the gains in absolute terms were not statistically significant for both Qwen and RoBERTa. However, there was no performance degradation for this dataset. Summarily, we find overall gains in using Joint Learning while fine-tuning task-specific models. Qwen and RoBERTa both show similar trends and comparable gains with Joint Learning as compared to simple fine-tuning.

Rules used as explanations: We use our rules, along with their label predictions, as an explanation for the input. k-shot-XP, uses a subset of exemplars comapred to k-shot, irrespective of the source pool from which it is retrieved. In spite of having lesser number of exemplars, adding contrastive explanations leads to further gains in our experiments for both Qwen and GPT 4. Further, use of explanations in k-shot settings with GPT 4 led to the highest performance for CDR among configurations. Similarly, we report the second best performance for ANLI using GPT 4, which has a percentage increase of more than 16 from the previous SotA. Previous SotA was a fine-tuned model, with 1/3rd of total training data, while our configuration under discussion is purely under ICL.

5.2 Impact of Generated Data

We find that using generated data, both for training and for ICL, leads to statistically significant gains in all configurations for the three datasets. In our experiments, we generate synthetic data in multiples of the original training data size. We generate data from $1\times$ to $6\times$ of the original data.

Fine-tuning: For all the three datasets, adding additional data beyond the full training data during fine-tuning leads to significant gains. For both discovery and CDR, we observe gains until $1.5\times$ times more data is added to the training data. For ANLI, we observe gains by doubling the training data size, i.e. $1\times$ the training data. We observe that the training saturates after when more data is being added to these datasets, until we tried with 3× more data. Our observations hold true for both Qwen and RoBERTA, where tried the fine-tuning.

ICL: There are more exemplars than that can fit into the 128K context windows for all the three datasets. However, increasing the pool of available data leads to improved outcomes in ICL, with the help of retrieval. For ANLI and CDR, we consistently had less than 40% presence of instances from the original training data for cases with 1× and above. With Discovery, we observe similar patterns but only after 1.5×. We stop our experiments with 3× data as the overlap between retrieved ICL exemplars was more than 90% by then.

5.3 Few-shot Setup

Few-shot learning setups are particularly valuable in industry applications involving text classification tasks with a large number of classes (> 50). The high annotation cost and resource demands in such settings can be mitigated by adopting few-shot strategies. Previous research has explored contrastive learning methods (Zhang et al., 2022c) for learning better semantic space for input representation, weak supervision techniques such as the joint learning framework proposed by Maheshwari et al. (2021), and synthetic data generation strategies (Lin et al., 2023). In this work, we evaluate the effectiveness of ARISE's data augmentation method in comparison to the other two approaches. Specifically, we examine the data augmentation thresholds at which the benefits of competing techniques become statistically insignificant.

We only use 5-shot gold data for each class, and augment data from 1× to 256× in multiples of 4 (Lin et al., 2023). For all datasets we find statistically significant gains to Joint Learning until 32× augmented data is used. However, with more supplementary data at 64× and beyond, we do not find statistical significance between models that use Joint Learning compared to the one not using Joint Learning. The only exception in the benchmark here is Amazon products, for which we find statistically significant gains to Joint Learning even with 200× data, but the gains disappears at 256× data. Similarly gains from contrastive learning in our CPFT baseline starts to disappear with 25-shot data (4× augmented) itself for all the datasets.

Moreover, we observe that fine-tuned variants of RoBERTa and Qwen models report comparable performances and do not have any statistical significance between their results. RoBERTa and Qwen report an average accuracy of 94.18 and 95.04 respectively. Here, the only dataset with a difference in statistical significance between these two are Amazon Products. 62.07% and 67.84% respectively are the accuracy for RoBERTa and Qwen for Amazon Products. While Qwen-ICL performs worse than Qwen-FT with an average accuracy of 90.38%, GPT 4 reports scores similar to the fine-tuned variants 95.46%.

Multilingual Experiments: On an average ARISE reports an absolute improvement of 7.21% points compared to the base model, on the 5-shot gold and $128x$ augemented data per class. The results show that our approach is applicable across a typologically diverse set of languages. We find *translation* of synthetically generated English sentences leads to empirically better results as compared to *direct* generation of data in the target language. The latter approach results in an absolute drop of 1.27% points. Moreover, GPT 4, under ICL reports an average of 84.15% accuracy as compared to the 80.4% accuracy reported by the RoBERTa Model.

6 Conclusion

We propose ARISE, a framework that iteratively generates and refines both synthetic data and rules. Overall we find gains in using our rules and data in both ICL and FT for more than 15 datasets we consider. Further, ARISE outperforms strong competitive baselines under comparable conditions. We also show the effectiveness of combining diverse sources of supervision that enable incorporating complementary and supplementary information beyond the available gold data to achieve SotA results.

Limitations

A major challenge with ARISE, currently is the overhead with the rule induction. We currently use syntactic n-grams with upto 3 nodes as our features. The search space exponentially increases as the nodes of the subtree increase, limiting our ability to induce higher-order tree structures as rules. While we currently rely on labeled instances of synthetically generated data, a strength of weak supervision is to incorporate unlabeled data. Several real-world scenarios often come up where unlabeled data is readily available. It needs to be further investigated whether synthetically generated labeled data can match the quality of real-world unlabeled data in the context of weak supervision. The current work does not explore this line of work, though it appears to be an important question that requires further investigation.

Ethics Statement

All experiments conducted in this study utilize publicly available datasets. We use publicly hosted APIs of GPT and Claude for synthetic data generation. The prompts used in this study included guardrails in the form of instructions to avoid generating problematic content.

Acknowledgements

We acknowledge BharatGen and IIT Bombay for providing resources and support to Vaibhav Singh. Yashwanth M. acknowledges Accenture for their support during this research. We also extend our appreciation to the reviewers for their valuable feedback.

References

Meysam Alizadeh, Maël Kubli, Zeynab Samei, and Shirin Dehghani. 2023. Open-source large language models outperform crowd workers and approach chatgpt in text-annotation tasks. *arXiv preprint arXiv:2307.02179*, 101.

Divya Jyoti Bajpai, Ayush Maheshwari, Manjesh Hanawal, and Ganesh Ramakrishnan. 2024. FAIR: Filtering of automatically induced rules. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 573–588, St. Julian's, Malta. Association for Computational Linguistics.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. 2024. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*.

Chris Biemann, Lachezar Krumov, Stefanie Roos, and Karsten Weihe. 2016. Network motifs are a powerful tool for semantic distinction. *Towards a Theoretical Framework for Analyzing Complex Linguistic Networks*, pages 83–105.

Mary Elaine Califf and Raymond J. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *CoNLL97: Computational Natural Language Learning*.

Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. 2020. Robust data programming with precision-guided labeling functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3397–3404.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *arXiv preprint arXiv:2311.09277*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

IDO Dagan, BILL DOLAN, BERNARDO MAGNINI, and DAN ROTH. 2010. Recognizing textual entailment: Rational, evaluation and approaches – erratum. *Natural Language Engineering*, 16(1):105–105.

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Benjamin L King, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2017. The comparative toxicogenomics database: update 2017. *Nucleic Acids Research*, 45(Database issue):D972.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. *arXiv preprint arXiv:2403.02990*.

Kuicai Dong, Aixin Sun, Jung-Jae Kim, and Xiaoli Li. 2022. Syntactic multi-view learning for open information extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4072–4083, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

John Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022a. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *Preprint*, arXiv:2204.08582.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2023. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Jack FitzGerald, Christopher Hench, Charith Peris, and Kay Rottmann. 2022b. Massively multilingual natural language understanding 2022 (MMNLU-22) workshop and competition. In *Proceedings of the Massively Multilingual Natural Language Understanding Workshop (MMNLU-22)*, pages 83–87, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Boris Galitsky and Dmitry I Ilvovsky. 2019. Least general generalization of the linguistic structures. In *FCA4AI@ IJCAI*, pages 39–44.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of English books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247, Atlanta, Georgia, USA. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Yiping Jin, Dittaya Wanvarie, and Phu T. V. Le. 2022. Learning from noisy out-of-domain corpus using dataless classification. *Natural Language Engineering*, 28(1):39–69.

Pride Kavumba, Ana Brassard, Benjamin Heinzerling, and Kentaro Inui. 2023. Prompting for explanations improves adversarial NLI. is this true? Yes it is true because it weakens superficial cues. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2165–2180, Dubrovnik, Croatia. Association for Computational Linguistics.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5464–5474. PMLR.

Emanuel Kitzelmann. 2010. *A Combined Analytical and Search-Based Approach to the Inductive Synthesis of Functional Programs+ H16709*. Ph.D. thesis, Otto-Friedrich-Universität Bamberg, Fakultät Wirtschaftsinformatik und ....

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11, Uppsala, Sweden. Association for Computational Linguistics.

Suraj Kothawade, Vishal Kaushal, Ganesh Ramakrishnan, Jeff Bilmes, and Rishabh Iyer. 2022. Prism: A rich class of parameterized submodular information measures for guided data subset selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10238–10246.

Amrith Krishna, Bishal Santra, Sasi Prasanth Bandaru, Gaurav Sahu, Vishnu Dutt Sharma, Pavankumar Satuluri, and Pawan Goyal. 2018. Free as in free word order: An energy based model for word segmentation and morphological tagging in Sanskrit. In *Proceedings of the 2018 Conference on Empirical Methods* in *Natural Language Processing*, pages 2550–2561, Brussels, Belgium. Association for Computational Linguistics.

Amrith Krishna, Bishal Santra, Ashim Gupta, Pavankumar Satuluri, and Pawan Goyal. 2021. A graph-based framework for structured prediction tasks in sanskrit. *Computational Linguistics*, 46(4):785–845.

Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81:53–67.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael Mahoney, Kurt Keutzer, and Amir Gholami. 2024. LLM2LLM: Boosting LLMs with novel iterative data enhancement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6498–6526, Bangkok, Thailand. Association for Computational Linguistics.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 590–599, Portland, Oregon, USA. Association for Computational Linguistics.

Yen-Ting Lin, Alexandros Papangelis, Seokhwan Kim, Sungjin Lee, Devamanyu Hazarika, Mahdi Namazifar, Di Jin, Yang Liu, and Dilek Hakkani-Tur. 2023. Selective in-context data augmentation for intent detection using pointwise V-information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1463–1476, Dubrovnik, Croatia. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ayush Maheshwari, Oishik Chatterjee, Krishnateja Killamsetty, Ganesh Ramakrishnan, and Rishabh Iyer. 2021. Semi-supervised data programming with subset selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4640–4651.

Ayush Maheshwari, Krishnateja Killamsetty, Ganesh Ramakrishnan, Rishabh Iyer, Marina Danilevsky, and Lucian Popa. 2022. Learning to robustly aggregate labeling functions for semi-supervised data programming. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1188–1202.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR.

Stephen Muggleton, Cao Feng, et al. 1992. Efficient induction of logic programs. *Inductive logic programming*, 38:281–298.

Ajay Nagesh, Ganesh Ramakrishnan, Laura Chiticariu, Rajasekar Krishnamurthy, Ankush Dharkar, and Pushpak Bhattacharyya. 2012. Towards efficient named-entity rule induction for customizability. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 128–138, Jeju Island, Korea. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben

Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. *arXiv e-prints*, pages arXiv–2401.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

GD Plotkin. 1970. A note on inductive generalization. machine intelligence, 5: 153-163, 1970.

Gordon D Plotkin. 1971. A further note on inductive generalization, machine intelligence 6. *Elsevier North-Holland, New York*, 101:124.

RJ Popplestone. 1970. An experiment in automatic induction. *Machine Intelligence*, 5:203–215.

Reid Pryzant, Ziyi Yang, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Automatic rule induction for efficient semi-supervised learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 28–44, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Luc De Raedt. 2010. *Logic of Generality*, pages 624–631. Springer US, Boston, MA.

Kavel Rao, Liwei Jiang, Valentina Pyatkin, Yuling Gu, Niket Tandon, Nouha Dziri, Faeze Brahman, and Yejin Choi. 2023. What makes it ok to set a fire? iterative self-distillation of contexts and rationales for disambiguating defeasible social and moral situations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12140–12159, Singapore. Association for Computational Linguistics.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160.

Mohammad Raza, Sumit Gulwani, and Natasa Milic-Frayling. 2014. Programming by example using least general generalizations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhavuk Singhal, Ashim Gupta, V P Shivasankaran, and Amrith Krishna. 2023. IntenDD: A unified contrastive learning approach for intent detection and discovery. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14204–14216, Singapore. Association for Computational Linguistics.

Ryan Smith, Jason A. Fries, Braden Hancock, and Stephen H. Bach. 2024. Language models in the loop: Incorporating prompting into weak supervision. *ACM / IMS J. Data Sci.*, 1(2).

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Shantanu Thakoor, Simoni Shah, Ganesh Ramakrishnan, and Amitabha Sanyal. 2018. Synthesis of programs from multimodal datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

Paroma Varma and Christopher Ré. 2018. Snuba: Automating weak supervision to label training data. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 12, page 223. NIH Public Access.

Ruida Wang, Wangchunshu Zhou, and Mrinmaya Sachan. 2023. Let‘s synthesize step by step: Iterative dataset synthesis with large language models by extrapolating errors from small models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11817–11831, Singapore. Association for Computational Linguistics.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1954–1963, Lille, France. PMLR.

Wikipedia. Category:21st-century English writers - Wikipedia — en.wikipedia.org. `https://en.wikipedia.org/wiki/Category:21st-century_English_writers`. [Accessed 01-10-2024].

Zhenyu Wu, Yaoxiang Wang, Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Jingjing Xu, and Yu Qiao. 2023. OpenICL: An open-source framework for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 489–498, Toronto, Canada. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Asaf Yehudai and Elron Bendel. 2024. When llms are unfit use fastfit: Fast and effective text classification with many classes. *Preprint*, arXiv:2404.12365.

John M Zelle, Raymond J Mooney, and Joshua B Konvisser. 1994. Combining top-down and bottom-up techniques in inductive logic programming. In *Machine Learning Proceedings 1994*, pages 343–351. Elsevier.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. 2022a. A survey on programmatic weak supervision. *Preprint*, arXiv:2202.05433.

Jieyu Zhang, Yue Yu, , Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. Wrench: A comprehensive benchmark for weak supervision. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. 2022c. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4892–4903, Dublin, Ireland. Association for Computational Linguistics.

Theodore Zhao, Mu Wei, J. Preston, and Hoifung Poon. 2024. Pareto optimal learning for estimating large language model errors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10513–10529, Bangkok, Thailand. Association for Computational Linguistics.

Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. Weaker than you think: A critical look at weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada. Association for Computational Linguistics.

A Appendix

A.1 Related Work

ARISE uses syntactic n-grams as its rules. Use of syntactic contexts in constructing feature space for downstream NLP tasks has been extensively explored in several of the past works (Liang et al., 2011; Goldberg and Orwant, 2013; Biemann et al., 2016). Goldberg and Orwant (2013) released a large scale collection of syntactic n-grams obtained from 3.4 million books. Biemann et al. (2016) looks from a network science perspective and focuses on graph motifs. Learning feature functions using morphosyntactic information as horn clauses has shown to benefit under a low-resource setting for languages such as Czech and Sanskrit, often requiring less than 10% of labeled training data required for neural counterparts (Krishna et al., 2021, 2018).

Using syntactic context, we incorporate signals that may not otherwise be explicitly captured in large language models. Further, we automate the generation and filtering of such rules by relying extensively on rule induction approaches (Varma and Ré, 2018; Bajpai et al., 2024; Lao and Cohen, 2010). Additionally, we consider our rule generation approach as a restricted instance of program synthesis via least general generalization as demonstrated in Raza et al. (2014), and Thakoor et al. (2018).

Data augmentation and generation in text has become effortless with LLMs (Ding et al., 2024). However, that does not ensure obtaining data with relevant supervisory signals, highlighting the need for targeted data filtering or generation (Killamsetty et al., 2021; Mirzasoleiman et al., 2020). This may include data scoring and ranking (Lin et al., 2023), iterative data generation (Rao et al., 2023), bootstrapping (Varma and Ré, 2018) or targeted subset selection (Wei et al., 2015). Wang et al. (2023) and Lee et al. (2024) explore similar themes by utilizing errors from language models to iteratively refine a synthetic training dataset. Similarly, (Hoang et al., 2018) discussed back-translation in the context of machine translation to augment training data. In ARISE, we use bootstrapping approach for data filtering and apply our filtering on synthetically generated data, instead of unlabeled data from an existing corpus.

A.2 Joint Learning with Rules

The few-shot classifier is trained using SPEAR (Maheshwari et al., 2021), a Joint Learning framework that learns a feature-based classification model and a label aggregation (LA) model. The feature model is a pre-trained neural network and LA is a generative model (Chatterjee et al., 2020), learned via PWS, using the automatically induced rules as labeling functions. Formally, LA is denoted as $P_\theta(\mathbf{l}_i, y)$, where $\mathbf{l}_i$ a vector that represents the firing of all LFs for an input $\mathbf{x}_i$. Each firing, l_{ij} can be either 0 (abstain) or class label k (Chatterjee et al., 2020). The model learns K parameters $\theta_{j1}, \theta_{j2}, \ldots, \theta_{jK}$ for each class corresponding to each LF l_j.

$$P_\theta(\mathbf{l}_i, y) = \frac{1}{Z_\theta} \prod_{j=1}^{m} \psi_\theta(l_{ij}, y) \tag{1}$$

$$\psi_\theta(l_{ij}, y) = \begin{cases} \exp(\theta_{jy}) & \text{if } l_{ij} \neq 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

$$\begin{aligned} Z_\theta &= \sum_y \prod_j \sum_{l \in \{1,0\}} \psi_\theta(l, y) \\ &= \sum_{y \in \mathcal{Y}} \prod_j (1 + \exp(\theta_{jy})) \end{aligned} \tag{3}$$

Following Maheshwari et al. (2021), our Joint Learning objective incorporates three different loss components for learning from labeled data. We provide a brief overview of each loss component below, while encouraging interested readers to (Maheshwari et al., 2021) for detailed information.

$$\begin{aligned} \min_{\theta,\phi} \sum_{i \in \mathcal{L}} L_{CE}\left(P_\phi^f(y|\mathbf{x}_i), y_i\right) + LL_s(\theta|\mathcal{L}) \\ + \sum_{i \in \mathcal{L}} KL\left(P_\phi^f(y|\mathbf{x}_i), P_\theta(y|\mathbf{l}_i)\right) \end{aligned}$$

The first component of the loss is the standard cross-entropy loss for the model P_ϕ^f. The second component is the negative log-likelihood on the dataset. The third is the KL-Divergence between the predictions from LA and P_ϕ^f, which enforces consensus by aligning their predictions.

A.3 Paraphrasing for Diverse Rules

We employ various techniques to generate diverse syntactic structures for our pool of available features to be used in the rule induction stage. First, we perform active to passive voice sentence phrasing and vice versa using LLMs. Second, we perform dependency tree morphing (Şahin and Steedman, 2018), to obtain simplified morphed dependency trees. Here, we remove peripheral relations like adjectives, such that the core semantics of the sentence is still preserved. Third, we apply role prompting (Schulhoff et al., 2024), by prompting LLMs to rewrite sentences in the style of well-known authors (Wikipedia). Role prompting was exclusively applied during monolingual experiments.