Leveraging Variation Theory in Counterfactual Data Augmentation for Optimized Active Learning

Anonymous ACL submission

Abstract

Active Learning (AL) allows models to learn interactively from user feedback. This paper introduces a counterfactual data augmentation approach to AL, particularly addressing the selection of datapoints for user querying, a pivotal concern in enhancing data efficiency. Our approach is inspired by Variation Theory, a theory of *human concept learning* that emphasizes the essential features of a concept by focusing on what stays the same and what changes. Instead of just querying with existing datapoints, our approach synthesizes artificial datapoints that highlight key similarities and differences among labels using a neuro-symbolic pipeline combining large language models (LLMs) and rule-based models. Through an experiment in the example domain of text classification, we show that our approach achieves a comparable accuracy to prevalent AL strategies while necessitating fewer annotations. This research sheds light on integrating theories of human learning into the optimization of AL.

1 Introduction

Active learning (AL) allows users to provide focused annotations to integrate human perception and domain knowledge into machine learning models (Settles, 2009). It relies on user's iterative annotations to build and refine model performance (Budd et al., 2021), as a result, the model's gain in performance with each round of annotation relies on the quality and quantity of annotated examples. In addition, AL faces a cold start problem, where initially, in the absence of sufficient annotated data, the model struggles to make effective learning decisions, impacting its early performance (Yuan et al., 2020). Previous work showed that careful selection of examples to be annotated is instrumental for optimal performance gain (Beck et al., 2013).

Prior work has employed theories in human cognitive learning to inspire how and what models learn (Zhang and Er, 2016). Following this direction, our work explores the use of a theory of human learning—The Variation Theory—to support human-AI collaboration in interactive machine learning. The Variation Theory of learning (Ling Lo, 2012; Marton, 2014; Marton and Booth, 1997) states that human learners can more effectively grasp critical aspects of a concept by experiencing variation along critical features. For instance, to comprehend the concept of a "ripe banana", learners should first encounter bananas alongside examples of other fruit, and then encounter various colors of bananas labeled as more or less ripe, so that they can recognizing the critical qualities of a banana, e.g., "yellowness" and firmness, as critical indicators of ripeness (Seel, 2011). Variation Theory involves two key steps: (1) identifying critical features and conceptual boundaries, and (2) devising new examples to delineate these conceptual boundaries. This work explores the relevance of the Variation Theory of human concept learning in contexts where an AI model is actively learning a concept from human-provided annotations; the variations that Variation Theory proscribes may assist both the machine and the human in this context.

Previous research showed the benefits of counterfactual data augmentation to enhance model performance (Liu et al., 2021; Yang et al., 2022a; Wang and Culotta, 2020; Reddy et al., 2023). However, a consistent challenge has been the scalable generation and selection of augmented data (Liu et al., 2022; Li et al., 2023). To address this, DISCO (Chen et al., 2023) proposed a method for automatically generating counterfactual data using task-agnostic models. Although DISCO provided a robust approach to augmented data, the use of a fully black-box pipeline could make debugging and improving the model difficult. To address this, we adopt a neuro-symbolic approach to define the concept boundaries in user annotations (Gebreegzi-

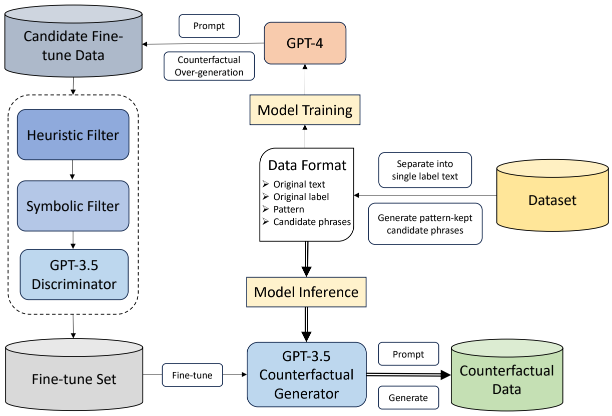

Figure 1: Inspired by Variation Theory of learning, our approach combines neuro-symbolic patterns with in-context learning to generate counterfactual examples for active learning. The single arrow indicates the model training data stream, while the double arrow indicates the model inference data stream.

abher et al., 2023).

In this paper, we combine a neuro-symbolic pattern-based approach (Gebreegziabher et al., 2023) to identify and vary over important features used by a classification model. We use an LLM backend to generate counterfactual data points to be used in consecutive rounds of model re-training. Specifically, we generate examples that change the assigned label into each of the remaining labels while still matching the original neuro-symbolic pattern. To ensure the quality of generated counterfactual examples, we design a three-step automatic filtering pipeline.

This paper makes the following contributions:

Evaluating the effectiveness of Variation Theory in active learning: We assess how the Variation Theory of human learning can enhance the robustness and address the cold-start challenges (Yuan et al., 2020) of early active learning. The results show that using counterfactual-based example selection results in higher accuracy with fewer annotations required compared to other example selection methods.

Quality of Counterfactual examples with neuro-symbolic approaches: Our approach employs Variation Theory to generate counterfactual data that differ from the original data semantically over neuro-symbolic dimensions but have high levels of syntactic similarity with the original annotated data. We assess the quality of generated counterfactual examples using a three-stage filtering mechanism. The results show significant increase in Soft Label Flip rate (SLFR) - the rate of removal of original label from counterfactual example, and high level of consistency in Label Flip Rate (LFR) - the rate of changing the original label into the target label in generated counterfactual examples.

In this paper, we assess the impacts of annotation selection, syntactic diversity, and semantic diversity of generated counterfactuals in active learning. We use a classification task to compare the performance of our method with baseline performance. Our method uses generated counterfactual data as augmentation, while the baseline uses existing "real" data along with example selection methods in Active Learning. The results show a promising potential of using counterfactual data to enhance user annotation in early active learning scenarios to bootstrap model learning with fewer human annotation.

2 Related Work

2.1 Data Generation and Augmentation

In domains with scarce annotated data, data augmentation methods aim to enhance the quantity and quality of training data (Yang et al., 2022b). Traditional data augmentation techniques, such as geometric transformations and color space alterations, do not modify the fundamental causal generative process. As a result, they do not counteract biases like spurious correlations (Kaushik et al., 2021).

Counterfactual data augmentation has been widely used to counteract spurious correlations in data (Denton et al., 2020; Liu et al., 2021; Yang et al., 2022a; Wang and Culotta, 2020). This approach employs counterfactual inference to control generative factors, facilitating the generation of samples that can address confounding biases. Many existing strategies uss dataset-specific counterfactual augmentation methods in specific domains such as sentiment analysis (Yang et al., 2022a; Kaushik et al., 2020), named entity recognition (Ghaddar et al., 2021), text classification (Wang and Culotta, 2020), and neural machine translation (Liu et al., 2021). A popular approach to address spurious dependence in NLP datasets is to use human-guided counterfactual augmentation (Kaushik et al., 2021). This approach presents individuals with data and preliminary labels, asking them to modify the data for an alternate label while avoiding unnecessary edits (Kaushik et al., 2020). This method depends on human efforts and expertise to overcome the challenge of automatically translating raw text into important features.

Recent studies examining data augmentation through a causal lens have received increasing attention due to their potential to enhance model performance and stability. For example, in computer vision, methods such as Counterfactual Generative Networks (CGN) (Sauer and Geiger, 2021) and CycleGANs (Zhu et al., 2020) were used to create counterfactual data points, building on the premise that the original training data contains learnable patterns. Similarly in natural language processing, prevalent techniques generate counterfactual samples by pinpointing and altering causal terms in sentences, which subsequently change their labels (Madaan et al., 2022; Liu et al., 2021; Yang et al., 2022a). However, most of these methods rely solely on internal data and may not ensure robustness against out-of-distribution (OOD) scenarios, especially if augmentations overlook context (Mouli et al., 2022). Joshi and He (2022) emphasized that limited diversity in these perturbations compromises the efficacy of counterfactually augmented data (CAD) in OOD scenarios, pointing to the necessity for innovative crowdsourcing approaches to elicit diverse perturbation of examples.

LLMs have shown to possess extensive generative capacity, making them a useful tool for counterfactual data generation. Li et al. (2023) introduced a method utilizing Language Models (LLMs) to generate domain-specific counterfactual samples through prompt design, highlighting the alignment between the efficacy of LLMs in domain-specific counterfactual generation and their overall proficiency in that domain. Although in-context learning has been a promising direction to get LLMs to perform different tasks Min et al. (2022) found that demonstrating the label space, the distribution of the input text, and the overall format of the sequence as important factors for the performance of in-context learning.

A consistent challenge in counterfactual generation has been the scalable generation and selection of augmented data (Liu et al., 2022; Li et al., 2023). To address this, DISCO (Chen et al., 2023) introduced a method for automatically generating high-quality counterfactual data using task-agnostic "teacher and student" models to allow classifier models to learn casual representation. DISCO uses a neural syntactic parser to select the spans of the sentence to vary on to generate data using Large Language Models (LLMs). Although DISCO provides more robust models trained on augmented data, the use of black-box approaches to generate data could make model debugging and improvement harder. To address this, we adopt a neuro-symbolic approach to define the concept boundaries in user annotations (Gebreegziabher et al., 2023).

2.2 Example-based Learning via Variation Theory

Based on previous studies on LLMs as counterfactual generators, our work seeks to learn from human cognition and example-based learning to better guide LLMs for generating higher quality data. *Will educational theories that work for human learners also work for AI?* Decades of research have demonstrated that utilizing example-based learning constitutes an effective instructional strategy for human acquiring new skills (Gog and Rummel, 2010). Similarly, few-shot learning is an example-based learning method used by LLMs.

How can we use human learning theories to support the annotation of data and training of LLM classifiers? Variation Theory, rooted in phenomenography, gives us insights from human experience (Cheng, 2016). The core concept of this theory involves presenting sets of examples that vary along a specific dimension, enabling learners to identify and use that dimension as a useful coordinate space for describing the underlying concept. This aligns with the foundational principle of counterfactual data augmentation in machine learning.

3 Approach

Drawing on Variation Theory, we propose using neuro-symbolic patterns for LLM in-context learning, aiming to create counterfactual examples for AL. We define learning spaces and concept boundaries through domain-specific patterns, which are executable syntactic representations of user annotations. Using these patterns and human labels, we fine-tune GPT-3.5 to produce data points that match the patterns but differ from user labels.

Intuitively, the generated counterfactual items are *syntactically similar* to an item known to be label X, predicted to be label X by an explainable pattern-based symbolic model, but predicted to be *not* label X by an LLM.

To ensure quality, we apply a three-level filtering mechanism (Fig. 2): heuristic regex for common LLM errors, symbolic filtering to verify rule compliance, and LLM-based discrimination to assess label change.

We evaluate our pipeline in a simulated interactive annotation task in AL, using the fine-tuned model to generate variations of human-annotated data. For example, for a concept A, with some annotated data, our approach generates a set of neuro-symbolic patterns based of pre-defined domain specific language adapted from Gebreegziabher et al. (2023) that characterize the concept (See Fig. 3 Step-1). At inference time, we prompt the fine-tuned GPT-3.5 to generate counterfactual data that changes an annotated data from concept A to a different concept B, based on the learned patterns (See Fig. 3 Step-2). This systematic approach helps our model identify the most relevant factors for the learning objective. We then use the generated examples as part of the training set in the classifier model and measure the accuracy.

3.1 Defining Concept Space with Neuro-symbolic Patterns

We use a neuro-symbolic approach to define and demonstrate learning space and concept boundaries for large language models (LLMs), allowing the generation of high-quality counterfactual data at scale. During annotation, we used PaTAT's (Gebreegziabher et al., 2023) interactive program synthesis approach to generate domain-specific pattern rules that match human annotated examples. The pattern rules represent the lexical, syntactic, and semantic similarities of data under the same label. This method generates a collection of regex-like (but with semantically-enhanced tags) that match with the annotated positive examples while excluding the annotated negative examples. For example, for data points in the domain of restaurant review *"Good food with great variety."* and *"The food was amazing."* both labeled "products" by the annotator, PaTAT learns patterns that match both sentences like *"[food]+*+ADJ"*, *"(amazing)+*"*. Below we show examples of PaTAT's pattern language:

- Part-of-speech (POS) tags: `VERB`, `PROPN`, `NOUN`, `ADJ`, `ADV`, `AUX`, `PRON`, `NUM`
- Word stemming: `[WORD]` (e.g., `[have]` will match all variants of have, such as *had*, *has*, and *having*)
- Soft match: `(word)` (e.g., `(pricey)` will match synonyms such as *expensive* and *costly*, etc.)
- Entity type: `$ENT-TYPE` (e.g., `$LOCATION` will match phrases of location type, such as *Houston, TX* and *California*; `$DATE` will match dates; `$ORG` will match names of organizations)
- Wildcard: `*` (will match any sequence of words)

Using the generated patterns for each concept, we apply zero-shot prompting with GPT-4 to generate counterfactual data points that match the pattern but match different concepts or labels present in the annotated data.

3.2 Generating Counterfactual Data with Fine-tuned LLM

Variation Theory says students learn by looking at the differences and similarities of certain features of a concept (Bussey et al., 2013). To generate counterfactual variants from original data point, the core is building conceptual understanding through small, connected steps that highlight the representational variances and invariances. However, real-world texts may be annotated with multiple labels, making it difficult to build conceptual understanding of them in small steps. Therefore we start our approach by creating single labeled examples that represent a single concept. To separate multi-labeled data into single-labeled examples, we utilize zero-shot GPT-4 with prompt to complete data preprocessing (See Fig. 3 Step-1).

Following this, we generate pattern rules by simulating iterative annotation using the ground truth labels. The generated patterns provide a syntactic and semantic representation for the annotated texts, using a rule-based, executable symbolic language. During counterfactual generation, we start by generating candidate phrases that adhere to these patterns (§A.1), ensuring the original syntactic integrity is preserved in the generated counterfactual variants. The generated phrases are then used as a constraint to be included in the generated counterfactual example. This constraint ensures that counterfactual examples remain within the syntactic boundaries set by the patterns with variations and distribution in the semantic content.

Fine-tuning smaller language models, such as GPT-3.5, can achieve results comparable to, or even surpassing, more advanced models like GPT-4. This approach is not only cost-effective but also particularly advantageous in large-scale commercial applications. As of December 2023, the cost of using a fine-tuned GPT-3.5 is just a tenth of employing GPT-4. To fine tune a GPT-3.5 counterfactual generator, we follow a three-step process (See Fig. 1): first, we prompt a GPT-4 model to generate counterfactual dataset over user assigned label and pattern rules (§A.1), then we filter the generated data over a three-stage criteria (Section 3.3), lastly using the set of filtered dataset we fine-tune a GPT-3.5 model to be used as a counterfactual generator during interactive annotation.

3.3 Filtering Generated Counterfactual Data

The ideal counterfactual variants should keep the pattern of original text, and successfully flip the original label to the target label. In our fine-tuning pipeline, we first generate counterfactual data 20 times the size of the original dataset. To ensure the quality of the fine-tuning dataset we implement a three-stage filtering mechanism:

3.3.1 Regex Heuristic Filtering

We use a heuristic-based filter to identify and remove low quality generations. This method uses regular expressions to detect common generation errors observed during our experimentation. We define rules to identify error patterns such as repetition of prompt, inaccurate formatting, which are common pitfalls in text generation systems, as indication of suboptimal output. This process functions autonomously, providing a seamless quality assurance layer that operates in real-time to generate the fine-tuning dataset without human intervention.

3.3.2 Neuro-symbolic Filtering

In the context of Variation Theory, it is crucial to strategically vary certain elements of an example while maintaining consistency in others. This practice serves to underscore the critical attributes of the feature under examination. In our study, the identified neuro-symbolic patterns serve as indicators of the key features that the classifier model considers significant within a sentence. To teach the importance of the feature and push the concept boundaries boundaries between inclusion and exclusion of a sentence beyond the identified patterns, it is important that the generated counterfactuals match the pattern of the original item. To ensure this, we implement a neuro-symbolic filtering method using executable domain specific neuro-symbolic patterns in § 3.1. We quantify this through the pattern keeping rate (PKP) as defined below.

$$PKR = \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}(\hat{p}_n = p_n)$$

where p_n is original pattern, $\hat{p}_n$ is the pattern given to the generated data point.

3.3.3 LLM-based Discriminator Filtering

Finally, we apply a filter using a GPT-3.5 discriminator that retains only generated counterfactuals that have effectively changed from the original label to the desired target label. We adopt two matrices (Chen et al., 2023) to quantify this - the Lable Flip Rate (LFR), and the Soft Label Flip Rate (SLFR) as defined below:

$$LFR = \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}\left(\hat{l}_n = L_n\right)$$

$$SLFR = \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}(\hat{l}_n \neq l_n)$$

where $\hat{l}_n$ is the label given by GPT-3.5 discriminator, L_n is the target label, l_n is the original label.

4 Experiments

We evaluate the generated counterfactuals in two phases: an automated filtering mechanism to detect the rates in which the generated data changes its label and though a standard classification task using a pre-trained model. We simulate and evaluate the effects of four different annotation selection in interactive AL: random selection, rule-based selection, counterfactual based example selection. We use each dataset's original label as ground truth and use GPT-3.5 to simulate human annotation of generated counterfactuals (Xiao et al., 2023).

4.1 Conditions

We investigate the implications of counterfactual example selection and other selection methods in interactive AL. Specifically, we use three conditions:

- **Condition 1: Random example selection** - In this condition random labeled examples are selected for each annotation iteration to train the classification model, serving as the baseline condition.

- **Condition 2: Clustering-based example selection.** This condition adopts lowest-confidence-first method, common in active learning approaches (Fu et al., 2013). To ensure data balance, original examples are initially transformed into word vectors. These vectors are then grouped using k-means, and the input order is ultimately generated by rotation among the different clusters.

- **Condition 3: LLM generated counterfactual example with filtering -** In this condition each selected example is paired with counterfactual examples generated by a fine-tuned GPT-3.5 model, where the fine-tuned data was filtered using the three step filtering method (§ 3.3).

4.2 Dataset

In order to simulate the subjectivity in human data annotation we chose datasets that exhibit high intra-coder reliability, but low inter-coder reliability. That is to say different annotators may hold controversial opinions on the same example, but for a single annotator, examples are of low ambiguity.

- **YELP**: The YELP dataset (Asghar, 2016) consists of user reviews of different businesses and services. The dataset itself provides 4 ground-truth categories (i.e. service, price, environment and products), we randomly sampled 495 examples for this experiment.

- **MASSIVE**: The MASSIVE (FitzGerald et al., 2022) virtual assistant utterances with 18 labeled intents as ground-truth (e.g. audio, cooking, weather, recommendation etc). For this experiment we randomly selected 30 examples from each category, making up a total of 540 examples.

4.3 Counterfactual Evaluation with Active Learning

To evaluate the generated counterfactual examples, we employ a simulated active learning task to fine-tune a BERT model (Devlin et al., 2018) for a multi-class classification task. We use the example selection conditions defined in § 4.1 to define a subset of 10, 15, 30, and progressively increasing upto 120 data points (referred to as 'shots'), alongside their corresponding ground truths. After finetuining the model we evaluate it against a holdoff set of the dataset.

To augment the model's training with generated counterfactual examples we pair each original data with its generated counterfactual examples and their assigned target label. This pairing aimed to enrich the training data, hypothesizing that the inclusion of counterfactuals would enhance the model's learning and predictive accuracy in early stages of annotation addressing the cold start problem (Yuan et al., 2020). Similarly, the performance of the model, in this case trained with both original and counterfactual dataset, was again evaluated against the same holdoff set. This comparative analysis aimed to quantify the impact of counterfactual examples on the model's ability to generalize and make accurate predictions on unseen data in early active learning scenarios.

4.4 Results

4.4.1 Automatic Generation Quality Evaluation

As shown in Table 1 we evaluate the quality of the generated counterfactual data using the two datasets. Building on the work of Chen et al. (2023), the efficacy of the counterfactuals was measured based on three metrics: Pattern Keeping Rate, Soft Label Flip Rate, and Label Flip Rate. These metrics were examined in two conditions: using GPT-4 to generate counterfactuals and using a fine-tuned GPT-3.5 counterfactual generator as defined in Fig 1. The results show that for both datasets, the multi-filtering and fine-tuning pipeline based on GPT-3.5 maintains or even improves the quality of generated counterfactuals on all metrics. Specifically, the Soft Label Flip Rate, which assesses the ability of counterfactuals to eliminate their original label, shows an increase rate of 7 when using the fine-tuned generator method compared to the GPT-4 generator for YELP and similarly a rate increase of 20 for the MASSIVE dataset. The Pattern Keeping Rate, which assesses whether the counterfactuals maintain the original data pattern indicating their syntactic similarity, also improves over raw GPT-4 generation, suggesting that the multi-filtering and fine-tuning pipeline enables generated data to retain its essential structure while changing its label. The absolute value of pattern retention is relatively low as we over generate counterfactuals on all target labels without checking whether the task itself is meaningful.

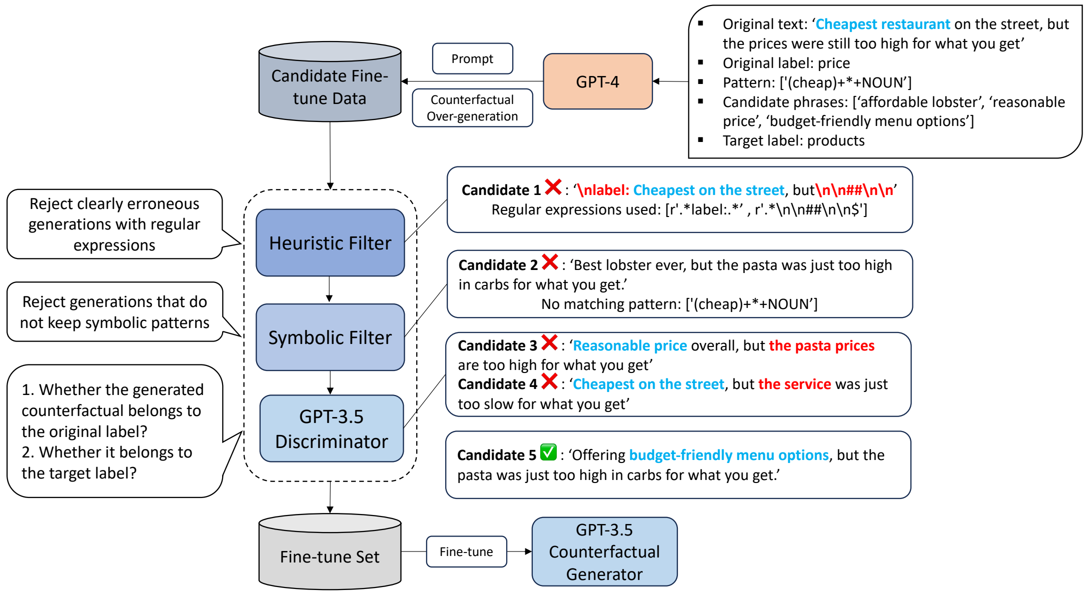

Figure 2: The candidate fine-tune data (raw GPT-4 generation) is first filtered by a heuristic filter, a symbolic filter and a GPT-3.5 discriminator. Then the filtered data will be used to fine-tune a GPT-3.5 counterfactual generator.

	Pattern Keeping Rate	
Method	YELP	MASSIVE
GPT-4 generation	7.86	22.08
Fine-tuned generation	13.64	22.85
	Soft Label Flip Rate	
Method	YELP	MASSIVE
GPT-4 generation	30.13	42.30
Fine-tuned generation	37.83	63.75
	Label Flip Rate	
Method	YELP	MASSIVE
GPT-4 generation	99.14	97.58
Fine-tuned generation	96.27	98.7

Table 1: Generated counterfactual data quality evaluation on raw GPT-4 generation vs. Fine-tuned generation.

4.4.2 Counterfactual Evaluation on Active Learning

In presenting our findings on the efficacy of generated coutnerfactuals in active learning as defined in § 4.3, we report the Macro F1-scores for two datasets (Table 2): YELP and MASSIVE. The results are stratified across different sizes of training data, ranging from 10 to 120 shots. For each size, we compare the performance of models trained on random samples of data, cluster based selection, andcounterfactuals augmented training. The F1-scores are accompanied by their respective standard deviations (SD), providing insights into the variability of the model performance.

The inclusion of counterfactuals with filtering consistently outperforms the baseline random selection across all data sizes. This trend is particularly less pronounced as the number of shots increases giving us a theoretical insight into how these generated counterfactuals can address training cold-starts in active learning. For the YELP dataset starting from an F1-score of 0.25 compared to 0.14 with random sampling in 10 shots. In the MASSIVE dataset, a similar trend is observed, with counterfactuals again showing a clear advantage over random selection. Starting with an F1-score of 0.144 at 10 shots, the model consistently achieves higher performance compared to the other conditions.

5 Limitations and Future work

Our proposed neuro-symbolic pipeline allows automatic and real-time generation of counterfactual data, however this ability is restrained to specific domains (business review and virtual utterance) and English language in our experiments. As the rule-based program synthesis in the data preprocessing process is designed exclusively for English, additional difficulties may arise when adapting our pipeline to other languages. We also point out that our fine-tuned counterfactual generators were built only from a single LLM, i.e. GPT-3.5. Even though our efforts are limited to Active Learning, we be-

	Macro F1-scores (YELP)						
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
Random	0.14	0.15	0.25	0.42	0.46	0.63	0.59
SD	0.12	0.11	0.04	0.18	0.12	0.09	0.20
Cluster	0.20	0.29	0.34	0.39	0.63	**0.81**	0.70
SD	0.14	0.15	0.09	0.10	0.19	0.12	0.11
Counterfactuals	**0.25**	**0.22**	**0.35**	**0.46**	**0.53**	0.65	**0.73**
SD	0.17	0.07	0.08	0.12	0.13	0.13	0.02
	Macro F1-scores (MASSIVE)						
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
Random	0.013	0.026	0.039	0.102	0.109	0.148	0.198
SD	0.011	0.019	0.011	0.040	0.063	0.065	0.036
Cluster	0.050	0.040	0.046	0.104	0.157	0.336	0.315
SD	0.028	0.032	0.024	0.109	0.038	0.035	0.067
Counterfactuals	**0.144**	**0.146**	**0.302**	**0.366**	**0.457**	**0.368**	**0.428**
SD	0.084	0.068	0.037	0.048	0.059	0.035	0.089

Table 2: Average F1-score with increasing numbers of annotations(shots) and the standard deviations(SD) across five independent experiments

lieve that leveraging LLMs for counterfactual data generation has the potential to benefit a wider array of tasks.

6 Conclusion

In this paper, we use Variation Theory to generate counterfactual examples over neuro-symbolic patterns to optimize annotation needs of Active Learning (AL). Our neuro-symbolic approach defines the concept boundaries between concepts in an interpretable way and helps large language model (LLM) based classifier models. We present a pipeline for generating counterfactual data using large language models (LLMs). This pipeline involves fine-tuning the LLMs on data generated by GPT-4, which is then filtered through a combination of a GPT-3.5 discriminator and an executable neuro-symbolic filter. This paper introduces the use of neuro-symbolic patterns as a means to define conceptual boundaries that play a role in determining the quality of generated counterfactual data. Through a simulated evaluation, we show that counterfactual datapoints generated by our proposed neuro-symbolic pipeline enable LLM-based classifiers to achieve a level of accuracy similar to widely used AL strategies while requiring fewer annotations. Our results show models using counterfactual examples perform better than models using random order example selection or cluster-based example selection. Furthermore, we provide a framework for generating and using counterfactual data with the original data to address challenges faced by lack of annotated data in early active learning scenarios.

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Daniel Beck, Lucia Specia, and Trevor Cohn. 2013. Reducing annotation effort for quality estimation via active learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 543–548.

Samuel Budd, Emma C Robinson, and Bernhard Kainz. 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062.

Thomas J Bussey, MaryKay Orgill, and Kent J Crippen. 2013. Variation theory: A theory of learning and a useful theoretical framework for chemical education research. *Chemistry Education Research and Practice*, 14(1):9–22.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. Disco: Distilling counterfactuals with large language models.

In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528.

Wai Lun Eddie Cheng. 2016. Learning through the variation theory: A case study. *The International Journal of Teaching and Learning in Higher Education*, 28:283–292.

Emily Denton, Ben Hutchinson, Margaret Mitchell, Timnit Gebru, and Andrew Zaldivar. 2020. Image counterfactual sensitivity analysis for detecting unintended bias.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

Yifan Fu, Xingquan Zhu, and Bin Li. 2013. A survey on instance selection for active learning. *Knowledge and information systems*, 35:249–283.

Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L. Glassman, and Toby Jia-Jun Li. 2023. Patat: Human-ai collaborative qualitative coding with explainable interactive rule synthesis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Abbas Ghaddar, Philippe Langlais, Ahmad Rashid, and Mehdi Rezagholizadeh. 2021. Context-aware Adversarial Training for Name Regularity Bias in Named Entity Recognition. *Transactions of the Association for Computational Linguistics*, 9:586–604.

Tamara Gog and Nikol Rummel. 2010. Example-based learning: Integrating cognitive and social-cognitive research perspectives. *Educational Psychology Review*, 22:155–174.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Divyansh Kaushik, Amrith Setlur, Eduard Hovy, and Zachary C. Lipton. 2021. Explaining the efficacy of counterfactually augmented data.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2023. Large language models as counterfactual generator: Strengths and weaknesses.

Mun Ling Lo. 2012. *Variation theory and the improvement of teaching and learning*. Göteborg: Acta Universitatis Gothoburgensis.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qi Liu, Matt Kusner, and Phil Blunsom. 2021. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 187–197, Online. Association for Computational Linguistics.

Nishtha Madaan, Srikanta Bedathur, and Diptikalyan Saha. 2022. Plug and play counterfactual text generation for model robustness.

Ference Marton. 2014. *Necessary conditions of learning*. Routledge.

Ference Marton and Shirley A Booth. 1997. *Learning and awareness*. psychology press.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

S Chandra Mouli, Yangze Zhou, and Bruno Ribeiro. 2022. Bias challenges in counterfactual data augmentation.

Abbavaram Gowtham Reddy, Saketh Bachu, Saloni Dash, Charchit Sharma, Amit Sharma, and Vineeth N Balasubramanian. 2023. Rethinking counterfactual data augmentation under confounding.

Axel Sauer and Andreas Geiger. 2021. Counterfactual generative networks.

Norbert M Seel. 2011. *Encyclopedia of the Sciences of Learning*. Springer.

Burr Settles. 2009. Active learning literature survey.

Zhao Wang and Aron Culotta. 2020. Robustness to spurious correlations in text classification via automatically generated counterfactuals.

Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 75–78.

Linyi Yang, Jiazheng Li, Pádraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2022a. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis.

Suorong Yang, Weikang Xiao, Mengcheng Zhang, Suhan Guo, Jian Zhao, and Furao Shen. 2022b. Image data augmentation for deep learning: A survey.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535*.

Yong Zhang and Meng Joo Er. 2016. Sequential active learning using meta-cognitive extreme learning machine. *Neurocomputing*, 173:835–844.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2020. Unpaired image-to-image translation using cycle-consistent adversarial networks.

A Appendix

A.1 Generation Pipeline

In this section we provide the details of all the prompts and models we use to construct the whole counterfactual generation pipeline.

A.1.1 GPT-4 Multi-label Separator

As shown in Fig. 3 Step-1, we utilize zero-shot GPT-4 to preprocess the raw data, in order to separate the given multi-labeled sentences into several single-labeled parts. We call GPT-4 through the API provided by OpenAI, set the temperature parameter to 0 and restrict the maximum token number to 512, which ensures the reliability of the generated results. The prompt used is shown below:

- {"role": "system", "content": "The assistant will seperate the given multi-labeled sentences into different parts, each corresponds to a label and a pattern (if the pattern is viable)"}
- {"role": "user", "content": "The assistant will make conversations based on the following example. New content should be in the format: 'text' + 'pattern' + 'label'; 'text' + 'pattern' + 'label'. All the text, patterns and labels are already given as input, if there is no corresponding pattern, just use '' to indicate empty."}
- {"role": "user", "content": "Each separated text must only have a single label, but may contain several patterns. Each label or pattern must appear at least once in the completion. The patterns can be composed with AND (+) or OR (|) operators."}
- {"role": "user", "content": "Conversation: Great customer service, reasonable prices, and a chill atmosphere. Pattern: ['(customer)+*+[service]', '(pay)|(sale)', '(environment)'] Label: price, service, environment"}
- {"role": "assistant", "content": "'Great customer service, ' + '(customer)+*+[service]' + 'service'; 'reasonable prices, ' + '(pay)|(sale)' + 'price'; 'and a chill atmosphere.' + '(environment)' + 'environment'"}
- {"role": "user", "content": "Conversation: {**text**} Pattern: {**pattern**} Label: {**label**}"}

A.1.2 GPT-4 Turbo Candidate Phrases Generator

As we are generating counterfactuals that keeps neuro-symbolic patterns, the first step of this task is to generate candidate phrases that keep the pattern but variate semantically, which make up crucial branches of generated counterfactual variations. For this part, we call GPT-4 Turbo through the API provided by OpenAI, set the temperature parameter to 0 and restrict the maximum token number to 256. The prompt used is shown below:

- {"role": "system", "content":"The assistant will create a list of phrases that match the given domain specific language based on the given definition."}
- {"role": "user", "content": "For the following text and pattern, generate as many diverse example phrases that match the given pattern and can be part of the given target label. Try to not use the word {**label**} or {**target_label**} in the phrases you generate. Separated your answer by a comma"}
- {"role": "user", "content": "text: {**matched_phrase**}, pattern: {**pattern**}, current label: {**label**} target label: {**target_label**}"}
- {"role": "user", "content": "The word '{**match**}' is a soft match, you can only use {**soft-match_words**} as its synonyms to replace it. You can not use other words for {**match**}"}

A.1.3 GPT-4 Turbo Counterfactual Generator

The GPT-4 Turbo generator will finish the second step of counterfactual generation, making use of candidate phrases generated in the first step and combining these semantic pieces into reasonable sentences. We set the temperature parameter to 0 and restrict the maximum token number to 256. The prompt used is shown below:

- {"role": "system", "content": "The assistant will generate a counterfactual example close to the original sentence that contains one of the given phrases."}
- {"role": "user", "content": "'Your task is to change the given sentence from the current label to the target.

  For example: 'Find me a train ticket next monday to new york city' with original label "transport" would be turned to 'Play me a song called New York City by Taylor Swift' with a label "audio".

  You can use the following phrases to help you generate the counterfactuals. Please make the sentence about {**target_label**}. Make sure that the new sentence is not about {**label**}. You must use one of the following phrases without rewording it in the new sentence: {**generated_phrases**}"'}
- {"role": "user", "content": "'You must follow three criteria:

  criteria 1: the phrase should change the label from {**label**} to {**target_label**} to the highest degree.

  criteria 2: the modified sentence can not also be about {**label**} and make sure the word {**target_label**} is not part of the modified sentence.

  criteria 3: the modified sentence should be grammatically correct."'}
- {"role": "user", "content": "If you find that you cannot generate new sentence that fulfill all the requirements above, just response 'cannot generate counterfactual' and don't feel bad about this"}
- {"role": "user", "content": "original text:{**text**}, original label:{**label**}, modified label:{**target_label**}, generated phrases:{**generated_phrases**}, modified text:"}

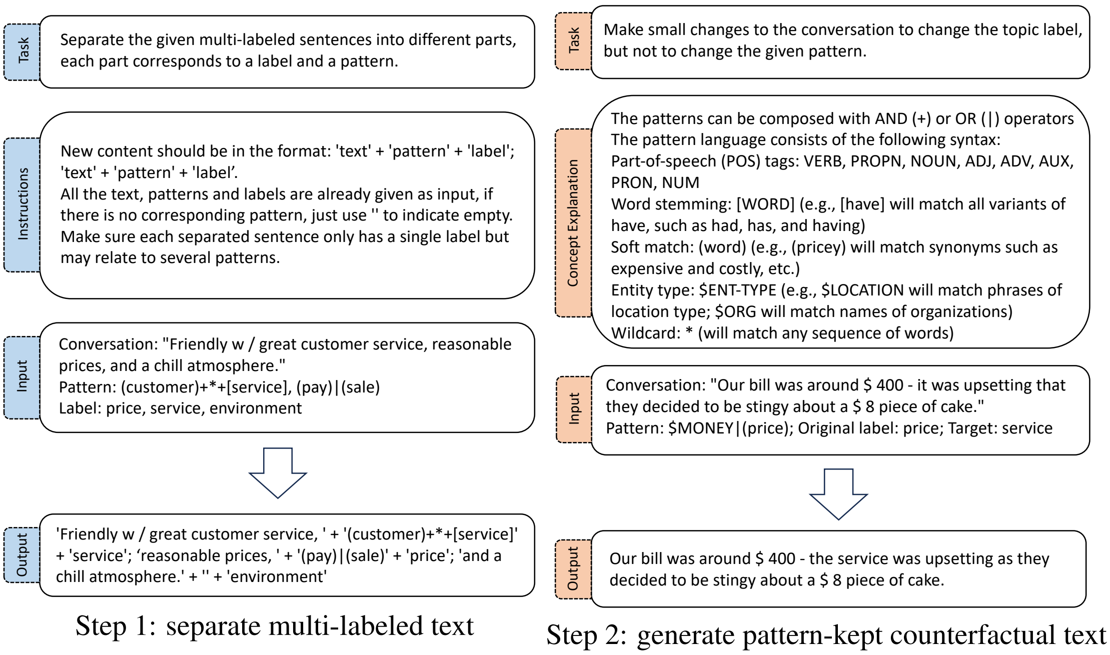

Figure 3: Illustration of LLM prompts used for preparing training datapoints and generating counterfactual datapoints