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 cold start problem, a pivotal concern in early stages of AL. Our data augmentation 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 data points, our approach synthesizes artificial data points that highlight potential 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 significantly higher performance when there are fewer annotated data. As the annotated training data gets larger the impact of the generated data starts to diminish showing its capability to address the cold start problem in AL. 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 a human's iterative annotations to build and refine model performance (Budd et al., 2021), and as a result, the model's gain in performance of with each round of annotations relies on the quality and quantity of annotated examples. However, the process of labeling data presents a significant bottleneck due to the cost and time associated with annotation (Fredriksson et al., 2020). Additionally, AL faces a cold start problem, where initially, in the absence of sufficient annotated data, the model is unstable and struggles to make effective learning decisions, affecting its early performance (Yuan et al., 2020). Previous work showed that careful selection of examples to be annotated is instrumental to achieve optimal performance gain (Beck et al., 2013).

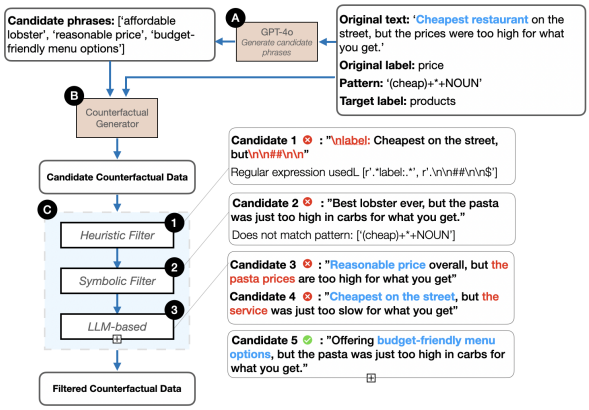


Figure 1: Inspired by Variation Theory of learning, our approach combines neuro-symbolic patterns with in-context learning to generate counterfactual examples for active learning.

The use of human cognitive learning theories as inspiration for how and what models learn has been shown promising in previous work (Zhang and Er, 2016). Following this direction, our work explores the novel 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 the critical aspects of a concept by experiencing variation along critical features. For example, to comprehend the concept of a "ripe banana", learners should first encounter bananas alongside examples of other fruits, and then encounter various colors of bananas labeled as more or less ripe, so that they can recognize the critical qualities of a banana, e.g. "yellowness" and

[*] Co-senior authors contributed equally.

"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). In the context of Variation Theory, synthesized counterfactual data can be more effective in capturing meaningful variations than real data selected from the dataset. However, the scalable generation and selection of augmented data has been a consistent challenge (Liu et al., 2022; Li et al., 2023a). To address this, DISCO (Chen et al., 2023) proposed a method for automatically generating counterfactual data using task-agnostic models. Despite its robust approach to augmented data, DISCO's use of a fully black-box pipeline makes debugging and improving the model difficult and does not allow meaningful presentation of variations that facilitates effective human annotation and sensemaking.

To address this, we propose a counterfactual generation pipeline that uses neuro-symbolic patterns to identify important features and uses them to guide the LLM's counterfactual generation[1]. Specifically, we use a programming-by-example approach (Gulwani, 2011) to generate neuro-symbolic patterns (Gebreegziabher et al., 2023). These patterns capture the syntactic and semantic similarities among similarly labeled examples. We then use the learned patterns to guide the LLM to generate counterfactual examples to be used in consecutive rounds of model re-training. The generated counterfactual examples change the assigned label into a different label while still keeping the original symbolic pattern in the data. In doing so, the generated examples introduce more meaningful variability in the data for subsequent model training. To further ensure the quality of the generated counterfactual examples, we design a three-step automatic filtering pipeline.

This paper makes the following contributions:

[1]ANONYMIZED

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) in 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 in cold start scenarios.

Quality of counterfactual examples generated using neuro-symbolic approaches: Our approach employs Variation Theory to generate counterfactual data that differ from the original data semantically over neuro-symbolic dimensions but maintain syntactic similarity with the original labeled data. We assess the quality of generated counterfactual examples using a three-stage filtering mechanism including the rate at which the symbolic patterns are kept consistent in the generated examples. The results show significant increase in the Soft Label Flip rate (SLFR)—the rate of removal of original labels from counterfactual examples, and a high level of consistency in Label Flip Rate (LFR)—the rate of changing original labels into target labels in generated counterfactual examples. By evaluating how often new examples meaningfully alter the original label and capture valuable variations – by keeping the original neuro-symbolic pattern – we can assess the efficacy of the examples produced.

This paper assesses 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 four baseline conditions, i.e., random selection and cluster-based selection, uncertainty-based selection, and counterfactuals without Variation Theory. Our method uses generated counterfactual data as augmentation, while the baseline uses existing "real" data along with example selection methods to train a multiclass classification model. The results across three datasets and two models show that the use of counterfactual generated data results in at least two times higher performance with fewer number of annotations(<70) compared to the other conditions. As the number of annotated data increases, the impact of the augmented data starts to diminish showing the efficacy of the approach in cold-start scenarios.

2 Related Work

2.1 Active Learning

Active Learning (AL) in machine learning is an approach in which the learning algorithm selectively chooses informative data points for mode training. Although most sampling strategies rely on a pool of unlabeled data (Fu et al., 2013), there are strategies that synthesize data points in real time for annotation (Schumann and Rehbein, 2019). The second approach, also called Membership Query Synthesis (MQS) creates new examples that inform the mode with more representative scenarios by either modifying existing instances (Wu et al., 2023, 2021) or generating new instances (Schumann and Rehbein, 2019).

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 use 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 through crowdsourcing (Kaushik et al., 2021; Joshi and He, 2022). 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.

LLMs have have been shown to possess extensive generative capacity, making them useful tools for counterfactual data generation. Li et al. (2023a) introduced a method utilizing 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., 2023a). 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 to generate higher quality data. *Will educational theories that work for human learners also work for AI?* Decades of research have demonstrated that using example-based learning constitutes an effective instructional strategy for human acquiring new skills (Gog and Rummel, 2010). Few-shot learning is an example-based learning method commonly used by LLMs.

How can we use human learning theories to support the annotation of data and training of LLM classifiers? Variation Theory (Marton, 2014), rooted in phenomenography, gives us insights from human experience, e.g., (Cheng, 2016). The core concept of this theory involves presenting sets of examples that vary along specific dimensions, enabling learners to identify and conceptualize the dimensions as a useful coordinate space for describ-

ing instantiations of the underlying concept. This aligns with the foundational principle of counterfactual data augmentation in machine learning.

3 Approach

Our approach applies the Variation Theory of human learning to machine learning in the context of active learning (AL). In order to adopt Variation Theory to AL we propose a new approach of counterfactual data generation by combining neuro-symbolic methods and LLMs. Specifically we use domain-specific neuro-symbolic patterns to learn syntactic representation of similarly labeled data that define a neuro-symbolic model's learning space and concept boundaries. We then use the learned patterns to guide the generation of augmented data that helps a classification model learn important nuances about each label (Fig. 1-A,B).

Through this approach we generate counterfactual data that are *syntactically similar* to their original counterpart but semantically belong to a different label. To ensure the quality of the generated counterfactuals, we apply a three-level filtering mechanism (Fig. 1-C).

3.1 Using Neuro-symbolic Patterns to Define Concept Space

Variation Theory suggests that humans learn a concept most effectively when they are shown examples that vary in only one specific dimension at a time, while all other aspects stay the same. Therefore, an important aspect of Variation Theory is determining which features should vary to emphasize their effects in the learning process. We achieve this by learning critical features from labeled data by generating neuro-symbolic patterns and make small modifications on the original sentence while maintaining consistency along the generated pattern.

3.1.1 Learning Neuro-symbolic Patterns

We use a programming-by-example (Lieberman, 2001) approach to establish the boundaries of concepts defined by data points and their associated ground truth labels. While our simulation study currently relies on ground truth labels, these will be substituted with human annotations in forthcoming interactive systems. After we randomly select a few annotations, we use PaTAT's (Gebreegziabher et al., 2023) interactive program synthesis approach to generate domain-specific pattern rules that match the annotated examples. These pattern rules represent the lexical, syntactic, and semantic similarities of data under the same label. PaTAT's pattern language includes the following components:

- 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)

Although the fundamental patterns are suitable for general domain text data, it is feasible to expand the pattern language to include specialized or domain-specific patterns.

This method generates a collection of regex-like patterns (but with semantically-enhanced tags) that match with the labeled positive examples while excluding the labeled negative examples. For example, if two data points in the domain of restaurant review *"Good food with great variety."* and *"The food was amazing."* have the same label "products", PaTAT learns up to 5 patterns that collectively match the set of examples annotated with that label. In this case, two patterns match both sentences, i.e., *"[food]+*+ADJ"*, *"(amazing)+*"*.

3.1.2 Using Neuro-symbolic Patterns for Counterfactual Data Generation

Using the learned neuro-symbolic patterns, we generate counterfactual examples by modifying the original text to be about a different label while still keeping the original pattern. To ensure minimal modifications and to make sure the reason for the original label is kept, we begin by generating candidate phrases for segments of the original sentence that matched the neuro-symbolic pattern (Fig. 1-A).

We use the generated candidate phrases as a constraints to be included in the generated sentence. For example in Fig. 1, the pattern *(cheap)+*+NOUN* has candidate phrases *['affordable lobster', 'reasonable price', 'budget-friendly*

menu']. When generating the counterfactual example we instruct the LLM to always include one of those phrases in the modified sentence. This constraint ensures that counterfactual examples that vary in semantic content remain within the syntactic boundaries set by the pattern, which defines, at least in part, the particular label for which counterexamples are being generated (Fig. 1-B).

3.2 Filtering Generated Counterfactual Data

The ideal counterfactual examples is a complete and coherent sentence that should keep the patterns of the original text, and successfully flip the original label to the target label. To ensure the quality of the fine-tuning dataset we implement a three-stage filtering mechanism:

3.2.1 Regex Heuristic Filtering

We use a heuristic-based filter to identify and remove counterfactual data with common generation flaws. This filter ensures that the generated sentences are coherent and complete. This method uses regular expressions to detect common generation errors observed during our experimentation (Fig. 1-C1). We define rules to identify error patterns such as repetition of prompt, inaccurate formatting, and incomplete generation, which were some common pitfalls we observed during generation.

3.2.2 Neuro-symbolic Filtering

The neuro-symbolic filter ensures that the generated counterfactual examples retain the original learned pattern. The original patterns represent features the model learns as useful conceptual boundaries. Therefore, keeping them in the counterfactually generated examples challenges the model's current boundary. To achieve this we implement the filter using executable neuro-symbolic patterns defined in § 3.1. Specifically, we check whether each generated counterfactual example matches its original counterpart's neuro-symbolic pattern (Fig. 1-C2). This filter excludes generated counterfactual examples that do not match with the provided pattern from being used in the consecutive training pipeline. To quantify this over the generated counterfactual examples we calculate the pattern keeping rate (PKR) 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 counterfactual data, and N is the size of the counterfactual data.

3.2.3 LLM-based Discriminator Filtering

Finally, we apply a filter using a GPT-4o discriminator. This filter removes counterfactuals that still keep their original label and all counterfactuals that do not change the label to the target label (Fig. 1-C3). This filter makes sure that the generated counterfactual examples have enough semantic changes that changes the original label to the target label. We adopt two matrices (Chen et al., 2023) to quantify this: the Label 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-4o discriminator, L_n is the target label, l_n is the original label.

SLFR measures the rate at which the generated counterfactual remove their original label, and LFR evaluates how often the counterfactual examples successfully adopt the target label.

4 Experiments

We evaluate the generated counterfactuals using two experiments. First, we evaluate the quality of generated counterfactual examples using the PKP, LFR, and SLFR metrics in § 3.2.

In the second experiment, we compare our proposed approach to other example selection techniques in a standard classification task, using two pre-trained models. We use five different data selection techniques in interactive AL: random selection, cluster-based selection, uncertainty-based selection, counterfactual examples generated without Variation Theory, and our proposed counterfactual based example selection. We use each dataset's original label as ground truth and use GPT-4o and a BERT model as the target classification models.

To further understand the impact of each component of our filtering pipeline, we conduct an ablation study. In this study to understand the impact of each individual filter on the pipeline's performance

of downstream model training. Additional details can be found in Appendix B.

4.1 Datasets

- **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.
- **Emotions**: Includes a collection of English Twitter messages annotates with 6 emotions: anger, fear, joy, love, sadness, and surprise (Elgiriyewithana, 2024). For this experiment we randomly selected 500 examples while balancing the number of labels.

4.2 Experiment 1: Generated Counterfactual Quality

We evaluate the generated counterfactuals using two experiments. First, we evaluate the quality of generated counterfactual examples using the PKP, LFR, and SLFR metrics in § 3.2.

4.2.1 Results

	YELP	MASSIVE	Emotions
Pattern Keeping Rate	0.94	0.88	0.81
Soft Label Flip Rate	0.45	0.71	0.58
Label Flip Rate	0.98	0.86	0.86

Table 1: Generated counterfactual data quality evaluation.

Our findings indicate that our proposed pipeline maintains the quality of generated counterexamples, as measured by Pattern Keeping Rate (PKR) and Label Flip Rate (LFR). Across datasets, the PKR remains high, demonstrating the generated counterfactual examples effectively keep the original pattern rules. The LLM-based Discriminator Filtering achieves robust performance in LFR across datasets, confirming that most counterfactual examples successfully adopt the target label. However, the Soft Label Flip Rate (SLFR) varies, particularly with the MASSIVE dataset showing the highest rate and the others on the lower side. This suggests that the degree of semantic change required to remove the original label can be dataset-dependent.

4.3 Experiment 2: Generated Counterfactuals in Downstream Model Training

In the second experiment we compare our counterfactual generation approach with five other sampling strategies in AL.

- **Random** Examples are randomly selected for each annotation iteration to train the classification model.
- **Cluster** Examples selected from a k-means clustered, pretrained Sentence Transformer model by iterating through the clusters in rotation.
- **Uncertainty** We use model confidence on the training set to choose data with the lowest confidence to be labeled. We use verbal uncertainty (Lin et al., 2022) to get model confidence in GPT-4o and model logits for the BERT model.
- **ALPS (Yuan et al., 2020)** We use ALPS a sampling strategy that addresses the cold start problem in AL.
- **Counterexamples without Variation Theory (nonVT)** We generate counterexamples without using the neuro-symbolic pipeline defined in Fig 1.

4.3.1 Protocol

To evaluate the generated counterfactual examples, we employ a simulated active learning task to train and evaluate a BERT model (Devlin et al., 2018) and few-shot prompting GPT-4o model for a multi-class classification task. We use the example selection conditions defined in § 4.3 to define a subset of 10, 15, 30, and progressively increasing upto 170 data points (referred to as 'shots'), alongside their corresponding ground truths to be used as training sets. We then evaluate the classifier model using a hold-off 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 is used to enrich the distribution and quality of the training data, hypothesizing that the inclusion of counterfactuals would enhance the model's learning and

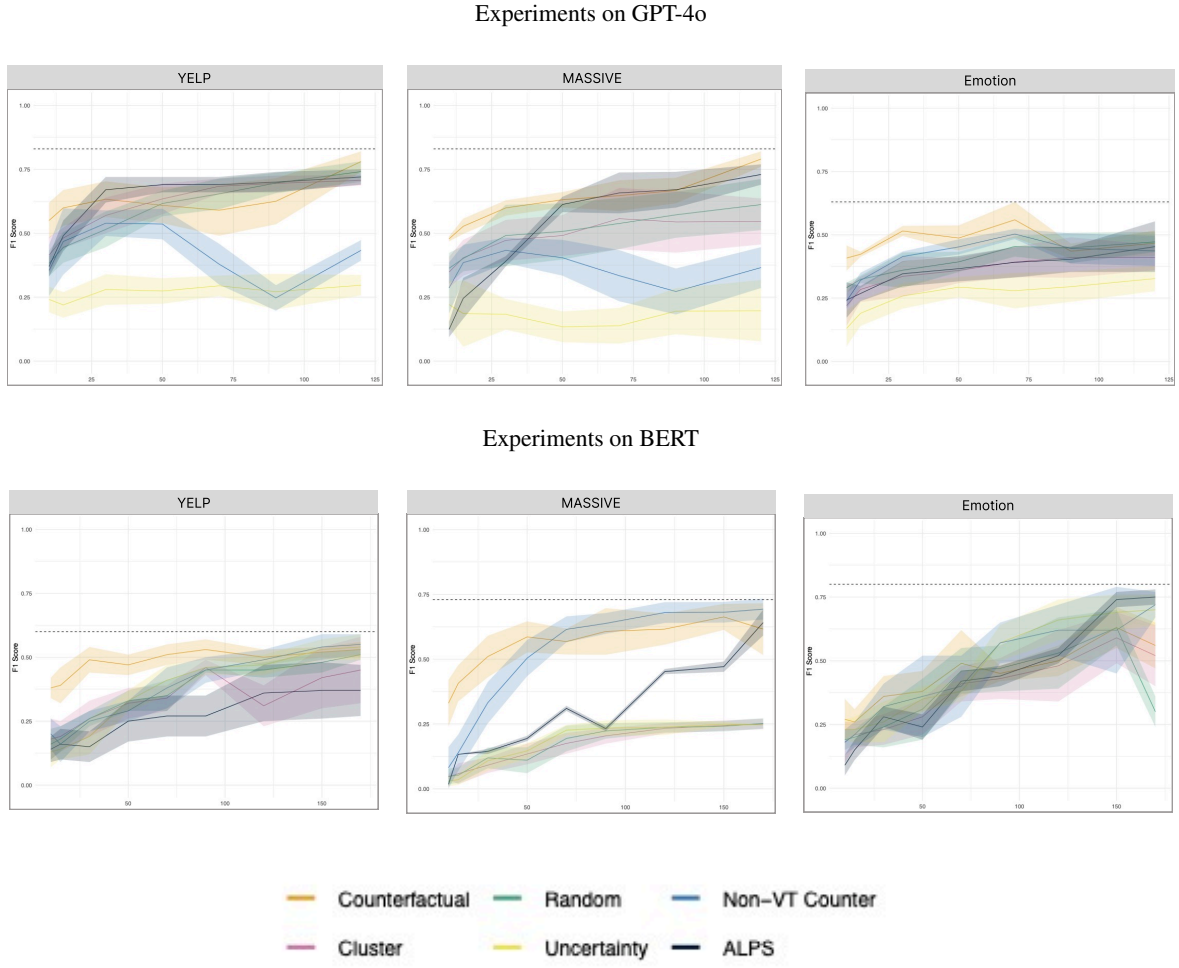


Figure 2: Experiment results across different datasets and conditions. Shown statistically significant difference between the counterfactual condition and the cluster condition. + indicates p-value<0.1, * indicates p-value<0.05, ** indicates p-value<0.01, and *** indicates p-value<0.0001.

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 hold-off 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.

4.3.2 Results

We present our findings on the efficacy of generated counterfactuals in active learning as defined in § 4.3.1. We report the macro F1-scores for the three datasets across different shots and conditions (Appendix: YELP dataset (Table 2), MASSIVE dataset (Table 3), and emotions dataset (Table 4)) using two models - few shot earning with GPT-4o and fine-tuning a BERT model. We use training shots ranging from 10 to 120 shots for GPT-4o to stay with-in OpenAI's token limit and 10 to 170 for the BERT model.

We conducted a pair-wise t-test between the counterfactual condition and the other baseline conditions to understand the impact of the proposed approach. The results across the three datasets highlight the strong initial impact that the counterfactual condition has in addressing the cold start problem in active learning (see Fig. 2). We consistently observe a statistically significant advantage of the counterfactual condition in lower shot numbers (see Table 2-4). As the number of annotate examples increases (50 shots and above in most cases), the difference in average F1-score decreases, suggesting the advantage of the counterfactual condition diminishes when more data become available. Similarly, we observe significant impacts

of the counterfactual condition when using a few-shot approach with the GPT-4o. However, we did not find results that consistently indicated a substantial difference between the random, cluster, and counterfactual without variation theory conditions after 50 shots of examples have been labeled. The results demonstrated the **performance advantage** of our proposed neuro-symbolic variation theory-based counterfactual data augmentation approach in cold-start scenarios for active learning tasks.

Our approach introduces useful data to address the lack of label distribution and representation in cold start scenarios. Compared to the *non-VT counter* condition, the counterexamples generated through Variation Theory have significantly higher F1-score, showing the impact of the pipeline in generating useful data in early AL. Moreover, the ablation study in Appendix B evaluating the impact of the filtering components in the pipeline shows there is statistically significant difference in downstream performance of a model trained on filtered data compared to data that does not have the complete filtering pipeline.

As we get more annotated data, we observe either minimal improvement or a decline in the model's performance. We believe that this occurs because after a certain point, the generated counterfactuals begin to replicate previously observed patterns, and there is a limit to the amount of information that can be extracted from these patterns. We also see similar patterns of model decline in the *non-VT counter condition*. This ultimately can have the model overly rely on the model, resulting in the performance not scaling. To address this, it is important to heuristically understand the amount of data distribution that can be captured by generated data and switch gears back to using real data when needed.

5 Conclusion

Li et al. (2023b) find that the performance of synthetic data is highly dependent of the distribution of the generated data, suggesting that enhancing data diversity could significantly improve the utility of synthetic data. Our approach achieves this by generating counterfactual examples along dynamic neuro-symbolic boundaries to allow the synthetic data to represent underlying concepts for better generalizability. This approach leverages the richness of the data's semantic structure, allowing for a more robust learning process during counterfactual generation by the LLM.

In our evaluation, we find that models trained on counterfactual examples have a statistically significant advantage in the early stage of active learning, where there is a limited number of annotated data. When there is only a small amount of annotated data available, the representation of a label's distribution does not sufficiently cover the latent space. The improvement in performance when using counterfactual data points highlights that the introduction of systematically generated counterfactual data adds the necessary variability for model training. In our experiment, both the GPT-4o and BERT classification models showed higher performance under the counterfactual condition across most datasets; however, the YELP dataset on GPT-4o emerged as an exception to this trend (Table 2).

Notably, the performance benefit of the counterfactual condition begins to decline when more than 70 labeled data points are used in model training. This reduction in advantage could potentially be attributed to model collapse. This happens when the model fails to capture the full diversity of the data on which it is trained (Wang et al., 2023; Su et al., 2023). With the introduced distribution shift, after the 70 shots threshold, the model might overfit to the specific characteristics of the synthetic examples it has seen, rather than generalizing to the broader real data distribution. This could lead to a decreased ability to handle new or slightly different data types introduced in later stages of training. As a result, the performance gains from using counterfactual examples no longer are significant because the model's adaptability is compromised. Identifying the optimal threshold for introducing counterfactual examples could be crucial, allowing us to strategically adapt our training approach based on the number of annotated real data available. This approach can particularly be applicable to handle cold start problems in active learning with data that require domain-specific, user-specific, or ambiguous annotation.

6 Limitations

Our neuro-symbolic pipeline enables the automatic, real-time creation of counterfactual data using a pattern-based program synthesis approach. This method defines the concept space varied during counterfactual generation. Although the current pattern building blocks are designed for general domains, they rely on predefined rules, which may

need augmentation with domain-specific lexical rules for specialized applications. Additionally, our use of a GPT-based discriminator to assign target labels for each counterfactual introduces potential biases or limitations inherent to the discriminator model itself. Future work could focus on understanding how human annotators undestand and label the generated counterfactual examples.

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.

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*.

Nidula Elgiriyewithana. 2024. Emotions Dataset.

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.

Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. 2020. Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In *International Conference on Product-Focused Software Process Improvement*, pages 202–216. Springer.

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.

Sumit Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. *ACM Sigplan Notices*, 46(1):317–330.

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. 2023a. Large language models as counterfactual generator: Strengths and weaknesses.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023b. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*.

Henry Lieberman. 2001. *Your wish is my command: Programming by example*. Morgan Kaufmann.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

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.

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.

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

Raphael Schumann and Ines Rehbein. 2019. Active learning via membership query synthesis for semi-supervised sentence classification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 472–481.

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

Burr Settles. 2009. Active learning literature survey.

Yi Su, Yixin Ji, Juntao Li, Hai Ye, and Min Zhang. 2023. Beware of model collapse! fast and stable test-time adaptation for robust question answering. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. 2023. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*.

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

Sherry Wu, Hua Shen, Daniel S Weld, Jeffrey Heer, and Marco Tulio Ribeiro. 2023. Scattershot: Interactive in-context example curation for text transformation. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 353–367.

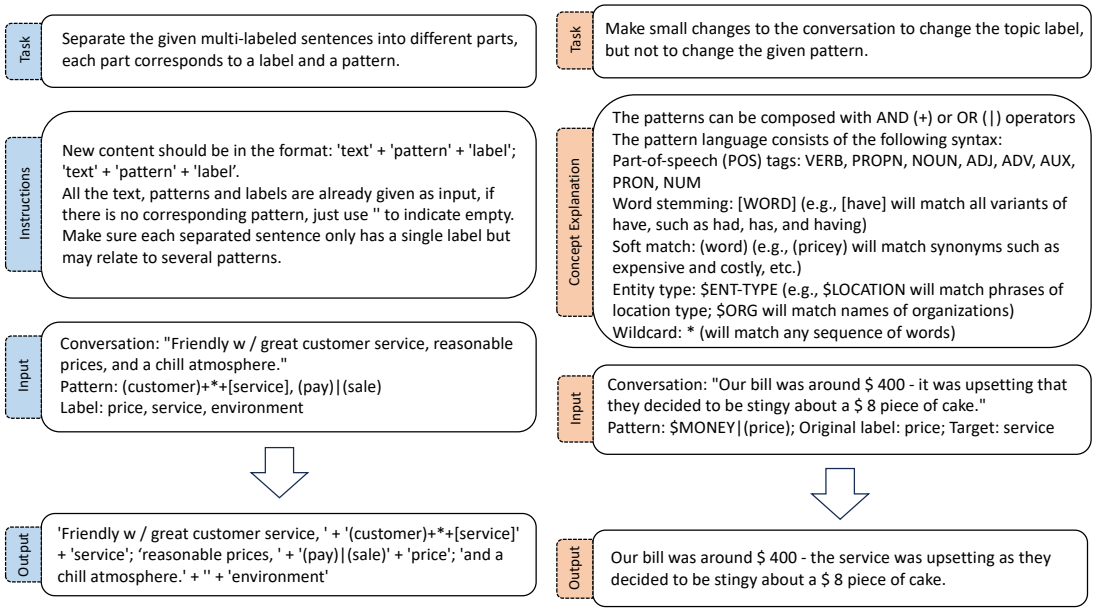
Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*.

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.



Step 1: separate multi-labeled text

Step 2: generate pattern-kept counterfactual text

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

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-4o 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 separate 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-4o 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-4o 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-4o Counterfactual Generator

The GPT-4o 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: "}

A.2 Experiment Results

	[YELP] Macro F1-scores (GPT-4o)						
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
Random	0.38^{***}	0.44^{***}	0.51^{***}	0.61	0.65	0.69^{+}	0.74
SD	0.05	0.06	0.07	0.05	0.06	0.04	0.04
Cluster	0.41^{***}	0.48^{***}	0.57	0.63	0.68^{*}	0.69^{+}	0.70
SD	0.07	0.04	0.07	0.06	0.03	0.03	0.02
Uncertainity	0.23^{***}	0.21^{***}	0.27^{***}	0.28^{***}	0.29^{***}	0.28^{***}	0.29
SD	0.04	0.05	0.06	0.05	0.04	0.06	0.05
ALPS	0.37^{**}	0.49^{*}	**0.66**	0.68	**0.69**	**0.70**	0.72
SD	0.04	0.06	0.05	0.03	0.03	0.04	0.03
Counterfactuals without VT	0.35^{***}	0.46^{*}	0.54^{*}	0.53^{*}	0.39^{***}	0.25^{***}	0.31
SD	0.10	0.13	0.05	0.06	0.08	0.05	0.05
Counterfactuals	**0.55**	**0.59**	0.63	**0.69**	0.59	0.65	**0.78**
SD	0.08	0.07	0.07	0.07	0.10	0.05	0.04

	[YELP] Macro F1-scores (BERT)								
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**	**150**	**170**
Random	0.16^{*}	0.18^{***}	0.26^{***}	0.33^{***}	0.35^{***}	0.45	0.45	0.48	0.51
SD	0.06	0.05	0.03	0.04	0.06	0.01	0.03	0.04	0.02
Cluster	0.18^{***}	0.19^{***}	0.26^{***}	0.32^{***}	0.34^{+}	0.46	0.31	0.42	0.45
SD	0.08	0.06	0.07	0.06	0.05	0.03	0.08	0.1	0.1
Uncertainty	0.13	0.14	0.19	0.33	0.41	0.46	0.47	0.53	0.54
SD	0.06	0.04	0.07	0.04	0.06	0.03	0.04	0.04	0.05
ALPS	0.14	0.16	0.15	0.25	0.27	0.27	0.36	0.37	0.37
SD	0.05	0.06	0.06	0.08	0.08	0.08	0.11	0.11	0.10
Counterfactuals without VT	0.20	0.16	0.25	0.29	0.38	0.45	0.49	**0.54**	**0.55**
SD	0.06	0.07	0.04	0.04	0.08	0.05	0.04	0.05	0.04
Counterfactuals	**0.38**	**0.39**	**0.49**	**0.47**	**0.51**	**0.53**	**0.50**	0.52	0.53
SD	0.04	0.07	0.05	0.04	0.04	0.04	0.03	0.02	0.03

Table 2: Average F1-score with increasing numbers of annotations(shots) and the standard deviations(SD) across 8 independent experiments using a fewshot prompting with OpenAI's GPT-4o and fine-tuned BERT model for classification on YELP dataset. + indicates p-value<0.1, * indicates p-value<0.05, ** indicates p-value<0.01, and *** shows p-value<0.0001 between the condition and the counterfactual condition.

	[MASSIVE] Macro F1-scores (GPT-4o)						
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
Random	0.36^{***}	0.40^{*}	0.49	0.51	0.54^{*}	0.57^{***}	0.61
SD	0.06	0.05	0.12	0.11	0.10	0.09	0.10
Cluster	0.35^{***}	0.40^{*}	0.47	0.49	0.56^{*}	0.54^{*}	0.55
SD	0.06	0.07	0.08	0.08	0.12	0.12	0.09
Uncertainty	0.22^{***}	0.19^{***}	0.18^{***}	0.13^{***}	0.14^{***}	0.19^{***}	0.20
SD	0.08	0.1	0.07	0.06	0.07	0.09	0.1
ALPS	0.12^{*}	0.24	0.39	**0.61**	**0.65**	**0.67**	0.72
SD	0.03	0.08	0.02	0.03	0.08	0.07	0.04
Counterfactuals without VT	0.26^{***}	0.37^{*}	0.43^{*}	0.40	0.34	0.27^{*}	0.37
SD	0.10	0.07	0.05	0.07	0.10	0.09	0.08
Counterfactuals	**0.48**	**0.52**	**0.59**	0.63	0.64	0.66	**0.79**
SD	0.01	0.03	0.03	0.03	0.06	0.05	0.03

	[MASSIVE] Macro F1-scores (BERT)								
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**	**150**	**170**
Random	0.048^{***}	0.052^{***}	0.12^{***}	0.11^{***}	0.19^{***}	0.22^{***}	0.23^{***}	0.24^{***}	0.25^{*}
SD	0.03	0.03	0.04	0.05	0.03	0.02	0.02	0.02	0.02
Cluster	0.046^{***}	0.058^{***}	0.091^{***}	0.13^{***}	0.18^{***}	0.20^{***}	0.23^{***}	0.24^{***}	0.25
SD	0.01	0.04	0.03	0.04	0.04	0.03	0.02	0.02	0.02
Uncertainty	0.029^{***}	0.035^{***}	0.11^{***}	0.14^{***}	0.22^{***}	0.23^{***}	0.24^{***}	0.25^{***}	0.25^{***}
SD	0.02	0.02	0.04	0.03	0.02	0.03	0.03	0.03	0.02
ALPS	0.017^{***}	0.13^{***}	0.14^{***}	0.19^{***}	0.31	0.23	0.45	0.45	0.64
SD	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.05
Counterfactuals without VT	0.09^{***}	0.15^{***}	0.33^{***}	0.50^{*}	$\mathbf{0.61}^{+}$	**0.64**	**0.68***	**0.68**	$\mathbf{0.69}^{+}$
SD	0.08	0.07	0.08	0.07	0.05	0.04	0.04	0.04	0.03
Counterfactuals	**0.33**	**0.40**	**0.51**	**0.58**	0.56	0.60	0.61	0.66	0.62
SD	0.09	0.07	0.08	0.06	0.05	0.09	0.06	0.05	0.1

Table 3: Average F1-score with increasing numbers of annotations(shots) and the standard deviations(SD) across 8 independent experiments using a fewshot prompting with OpenAI's GPT-4o and a BERT model for classification on the MASSIVE dataset. **+** indicates p-value<0.1, ***** indicates p-value<0.05, ****** indicates p-value<0.01, and ******* shows p-value<0.0001 between the condition and the counterfactual condition.

	[Emotions] Macro F1-scores (GPT-4o)						
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
Random	0.29	0.32	0.36^{***}	0.39^{***}	0.45^{*}	0.45	0.47
SD	0.1	0.1	0.07	0.04	0.04	0.06	0.04
Cluster	0.32	0.38	0.36^{***}	0.39^{***}	0.42^{*}	0.42	0.41
SD	0.04	0.04	0.08	0.12	0.09	0.08	0.05
Uncertainty	0.21^{***}	0.19^{***}	0.25^{***}	0.29^{***}	0.28^{***}	0.29	0.33
SD	0.07	0.05	0.05	0.04	0.07	0.06	0.05
ALPS	0.23	0.26	0.34	0.36	0.39	0.40	0.44
SD	0.07	0.03	0.05	0.05	0.06	0.05	0.10
Counterfactuals without VT	0.28	0.35	0.46	0.48	0.49	0.36	0.39
SD	0.06	0.10	0.12	0.13	0.12	0.08	0.07
Counterfactuals	**0.34**	**0.43**	**0.54**	**0.51**	**0.58**	**0.47**	**0.52**
SD	0.08	0.1	0.1	0.05	0.1	0.03	0.05

	[Emotions] Macro F1-scores (BERT)								
No. shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**	**150**	**170**
Random	0.19^{*}	0.20^{***}	0.24^{*}	0.31	0.46	0.47	0.53	0.63	0.30
SD	0.04	0.03	0.08	0.12	0.09	0.09	0.14	0.07	0.06
Cluster	0.18^{*}	0.21^{*}	0.23^{***}	0.28^{*}	0.41	0.43	0.48	0.59	0.52
SD	0.02	0.03	0.02	0.03	0.05	0.08	0.06	0.05	0.12
Uncertainty	0.23^{***}	0.23	0.26^{*}	0.35	0.38^{+}	$\mathbf{0.57}^{***}$	$\mathbf{0.66}^{***}$	0.69	0.70^{*}
SD	0.04	0.05	0.08	0.05	0.04	0.07	0.08	0.07	0.06
ALPS	0.09	0.15	0.28	0.24	0.42	0.44	0.52	**0.74**	**0.75**
SD	0.04	0.04	0.04	0.05	0.04	0.03	0.03	0.03	0.03
Counterfactuals without VT	0.18^{*}	0.21^{*}	0.32	0.36	0.40	0.57^{***}	0.62	0.62	0.72^{*}
SD	0.05	0.05	0.09	0.12	0.13	0.08	0.1	0.2	0.05
Counterfactuals	**0.27**	**0.26**	**0.36**	**0.38**	**0.49**	0.45	0.50	0.63	0.56
SD	0.07	0.09	0.05	0.12	0.05	0.15	0.06	0.06	0.07

Table 4: Average F1-score with increasing numbers of annotations(shots) and the standard deviations(SD) across 8 independent experiments using a fewshot prompting with OpenAI's GPT-4o and a BERT model for classification on the emotions dataset. + indicates p-value<0.1, * indicates p-value<0.05, ** indicates p-value<0.01, and *** shows p-value<0.0001 between the condition and the counterfactual condition.

No of Shots	**10**	**15**	**30**	**50**	**70**	**90**	**120**
No Filters	0.10	0.12	0.15	0.23	0.23	0.21	0.21
SD	0.03	0.04	0.05	0.04	0.04	0.03	0.03
Herustic Filter	0.15	0.17	0.19	0.28	0.27	0.28	0.28
SD	0.08	0.1	0.1	0.07	0.09	0.1	0.1
Herustic + Symbolic Filters	0.12	0.13	0.13	0.17	0.16	0.18	0.20
SD	0.04	0.03	0.01	0.02	0.03	0.02	0.01
Herustic + LLM Discriminator	0.17	0.21	0.23	0.34	0.42	0.45	0.49
SD	0.08	0.04	0.09	0.07	0.02	0.02	0.05
Herustic + Symbolic + LLM Discriminator	**0.38**	**0.39**	**0.49**	**0.47**	**0.51**	**0.53**	**0.50**
SD	0.04	0.08	0.06	0.04	0.05	0.05	0.04

Table 5: Average F1-score and SD from an ablation study with the YELP dataset on BERT model

B Ablation Study on Counterfactual Filtering Methods

We performed an ablation study to investigate the impact of the different components in our filtering pipeline. We follow the same approach as § 4.3.1 where each condition is run with different seeds 8 times. For each condition we report an average F1 score and the standard deviation (SD) in Table 5. Our approach involves generating counterexamples with a fine-tuned GPT-4o model and applying all three filters defined in § 3.2 before using the data for active learning.

In this study, we investigate the impact of different configurations by varying the filtering mechanisms used with the generator model.

The ablation study is conducted using the YELP dataset with a BERT model for the downstream active learning tasks. The configurations tested include:

- No Filters: Counterexamples generated without any filters applied
- Heuristic Filter: Applying only the heuristic filter
- Heuristic + Symbolic Filters: Applying both heuristic and symbolic filters
- All Filters: Applying all three filters defined in § 3.2

The results indicate that the use of all filters significantly improves the performance of the trained model (See Table 5). The average F1-score with all filters applied reaches 0.51 for 70 shots and peaks at 0.53 for 90 shots, demonstrating a 2X improvement over the baseline with no filters (F1-score of 0.23 for 70 shots). Using a pairwise t-test we find that this is statistically significant ($p<0.0001$), underscoring the value of carefully filtering LLM-generated counterfactuals to produce usable data for model training.

Surprisingly, we found that incorporating the symbolic filter without the LLM discriminator decreases the performance of downstream training. Further analysis of the included examples revealed that some generated sentences included the original sentence with additional parts that corresponded to the target label. While the LLM discriminator would filter these out, without its use in the pipeline, these generated counterfactuals are mistakenly treated as negative examples, when technically they are just multi-labeled positive examples. However, we observe a substantial improvement in performance when the symbolic filter is used in conjunction with the LLM discriminator, as opposed to using the LLM discriminator alone. This demonstrates the effectiveness of combining both methods to enhance the quality and accuracy of the generated counterfactuals.

The ablation study highlights the crucial role of the filtering pipeline. By systematically evaluating the impact of each component, we demonstrate that the integration of heuristic, symbolic filters, and the LLM discriminator leads to significant improvements in downstream active learning task. This validates our hypothesis that filtering LLM-generated data is essential in determining usable and useful data for achieving higher performance and reliability in model training