Attributes as Textual Genes: Leveraging LLMs as Genetic Algorithm Simulators for Conditional Synthetic Data Generation

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel at generating synthetic data, but ensuring its quality and diversity remains challenging. We propose **Genetic Prompt**[1], a novel framework that innovatively combines genetic algorithms (GAs) with LLMs for synthetic data generation. Unlike previous GA approaches, Genetic Prompt implicitly represents text attributes as gene sequences. Additionally, We implement an active learning scheme to optimize the GA process, eliminating traditional fitness evaluation and reducing inference costs. Experiments on four relation extraction datasets show that Genetic Prompt significantly outperforms baselines including SimPrompt and AttrPrompt. Further more, analysis using metrics such as average pairwise sample similarity and Central Moment Discrepancy reveals that Genetic Prompt generates more diverse data with distributions closer to the gold dataset.

1 Introduction

Recent advances in large language models (LLMs) such as GPT-4 (OpenAI, 2024) have led to a significant paradigm change in natural language processing. However, their massive scale results in high deployment costs and low inference efficiency. Therefore, using smaller models fine-tuned on specific data remains a popular approach for downstream tasks (Zhao et al., 2024). Due to their lower cost compared to time-consuming and highly-cost process of data collections and human annotations, LLMs are seen as promising ways to meet the needs by synthetic data generation (Yu et al., 2021; Russo et al., 2020; Logeswaran et al., 2018). For these smaller models, the key factors affecting their performance are the quality and diversity of training data (Bayer et al., 2022). A number of studies (Long et al., 2024; Yu et al., 2023) have shown that the low-quality and poor-diversity of synthetic data generation by LLMs can weaken model robustness and generalizability.

However, how to promote the quality and diversity of the generated synthetic data is critical yet understudied (Ye et al., 2022). A popular direction is to find ways to better inform and guide LLMs to generate highly-qualified and -diverse data entries, conditional prompt engineering. Recent research studies have developed different conditions to instruct and guide synthetic data generations, such as clinical named entities (Xu et al., 2024; Josifoski et al., 2023), label correlation (Chung et al., 2023), and writing styles (Xu et al., 2024; Yu et al., 2023). While those approaches have improved in-context learning capabilities of LLMs for synthetic data generations, those pre-defined data conditions may miss the varying nature of data domains (e.g., news article (Xu et al., 2024) vs clinical (Yu et al., 2023)) and potentially reduce diversity and generalizability of those approaches.

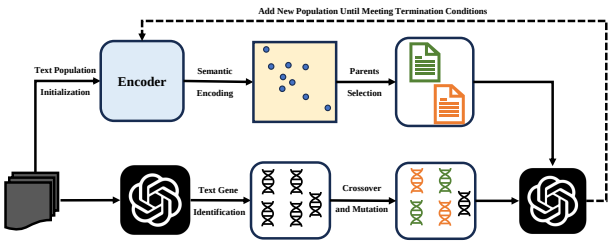

Figure 1: The Genetic Prompt framework for synthetic data generation.

Inspired by the genetic algorithm (Mitchell, 1980) (GA) of using evolution and genetics theories that captures real-world distributions, we propose a **Genetic Prompt** Framework (Figure 1) for synthetic data generation that 1) implicitly treats text attributes as gene sequences in genetic algorithms, 2) diversifies synthetic data by gene mutation process to encounter data variations, and 3) leverages LLMs' reasoning and in-context learning

[1]The data and code is available on anonymous repository: https://anonymous.4open.science/r/Genetic-Prompt4SDG-44B5/README.md

Algorithm 1 Genetic Algorithm Implemented by LLM for Synthetic Data Generation

Input: Label set L, textual genes G, target data size n, diversity coefficient α.
Initialize population $P = \{p_0, p_1, \ldots, p_{50}\}$ from gold dataset for each label l.
Build a sample pool $\{P, E\}$ for each label l for the initial population.
▷ E are semantic representations of population P.
Calculate the diversity score $APS = \{a_0, a_1, \ldots, a_l\}$ of each class in gold dataset.
▷ APS: **A**verage **P**air wise sample **S**imilarity.
for $l \in L$ **do**
  while population $< n$ and $a < \alpha \cdot a_l$ **do**
    Get semantic distances of pairs in $\{P, E\}$: $\forall e_i, e_j \in E, \quad d_{ij} = \text{dist}(e_i, e_j)$
    Select the most distant pair from **unused** samples as parents: $(p_i^*, p_j^*) = \arg\max_{e_i, e_j} d_{ij}$
    Randomly partition textual genes G into three groups: $G = G_1 \cup G_2 \cup G_3$.
    Perform LLM-based crossover on G_1 and G_2 using parents (p_i^*, p_j^*) and perform mutation on $G_3 : p_{new} = \text{LLM-GA}(p_i^*, p_j^*, G_1, G_2, G_3)$
    Add the new sample to pool: $\{P, E\} := \{P, E\} \cup \{p_{new}, e_{new}\}$
    Calculate the diversity score a of current population: $a = APS(P)$.
  end while
  Remove the initial population from the sample pool.
end for
Output: High diversity and quality synthetic data

abilities to augment a small amount of real data. The integration of GAs with LLMs has yielded promising results in domains like neural architecture search (Chen et al., 2023) and code generation (Lehman et al., 2022) by leveraging LLMs' textual capabilities and GAs' optimization strengths. However, the GA and LLMs have rarely been explored for synthetic data generation. Genetic algorithms (Wu et al., 2024) usually include an fitness evaluation process where high-fitness samples are selected as parents, and low-fitness offspring are discarded. We utilize active learning (AL) methods (Zhang et al., 2023; Ren et al., 2021) to select the semantically most distant pair of samples as parents, maximizing text diversity through LLMs' in-context learning abilities. For the generated offsprings, we do not evaluate its fitness but directly add them to the population, which avoids the increased costs typically associated with data generation process.

We conducted our experiments on four relation extraction datasets across multiple domains (e.g., biochemistry, news, and social media), demonstrate the effectiveness and generalizability of our Genetic Prompt framework for generating high-quality synthetic data. Across multiple classifiers, Genetic Prompt consistently outperformed baseline methods SimPrompt and AttrPrompt, achieving average increases of 18.5% in accuracy and 18.3% in macro-F1 score compared to SimPrompt, and 12.1% and 11.1% respectively compared to AttrPrompt. Notably, Genetic Prompt showed superior data scalability and diversity, with performance improving as sample size increased from 1,000 to 3,000. Analysis using metrics such as average pairwise sample similarity, inter-sample N-gram frequency, and Central Moment Discrepancy revealed that Genetic Prompt generated more diverse data with distributions closer to the gold dataset. Our contributions can be summarized as follows.

- We propose **Genetic Prompt**, which for the first time extends the collaborative paradigm of LLMs and genetic algorithms (GAs) to the field of synthetic data generation.

- We optimize the genetic algorithm process using an active learning scheme, eliminating the need for traditional fitness evaluation and avoiding wasted inference costs.

- Experiments on four datasets demonstrate that Genetic Prompt not only generates data closer to the distribution of the original data but also enhances the diversity of generated training samples. Experiments with multiple different classifiers prove that the data generated by Genetic Prompt is superior in both quality and scalability compared to baseline methods.

Crossover and Mutation Operations Implemented by LLMs

You need to generate synthetic sentence for the **Protein Chemical Relation extraction** task.

Example 1: <protein> Losartan </protein> was the first, but by no means remained the only, <chemical> AT1 </chemical> receptor antagonist.

Example 2: <protein> CYP3A5 </protein> carrier status had no influence on <chemical> midazolam </chemical> oral clearance or its inhibition by ketoconazole.

The sentence should follow the requirements below:
1. Must discuss about the **Agonist** Relation between **Chemical** and **Protein.**
2. The Sentence Structure, Entity Proximity and Polarity should inherit from Example1.
3. The Length and Style should inherit from Example2.
4. The Voice and Modifier must be **different from the given 2 examples.**

Figure 2: Prompt template for ChemProt Dataset. The Blue text highlights the textual genes identified by LLM. The Green and Orange text represent the two selected parents used in the crossover operation. And the offspring generated by LLM will randomly inherit some of their genes and undergo random mutations in the remaining genes.

2 Genetic Prompt for Synthetic Data Generation

In this section, we describe the proposed **Genetic Prompt** framework in detail. The key idea of Genetic Prompt is to treat textual attributes as genes and prompting LLMs to do crossover and mutation operations over a small initial population from gold data. It consists of three major modules: 1) textual genes identification, 2) active parents selection and 3) crossover and mutation. We formalize this three modules into algorithmic instructions that guide LLMs in implementing the Genetic Prompt framework, as detailed in Algorithm 1, the complete process is illustrated in Figure 1.

2.1 Textual Genes Identification

To improve the quality of text generation or alter its style, attribute-aware conditional text generation has been proven effective. However, these methods generally rely on attribute values strictly predefined by humans (Yu et al., 2021; Russo et al., 2020; Logeswaran et al., 2018) or LLMs (Xu et al., 2024; Yu et al., 2023). In contrast, our approach views text attributes as the 'genes' of the text. This means we only need to use a single round of semi-automatic interaction to identify the text's genes, without the need to predefine specific values for each attribute.

Specifically, our solution involves inputting the task type (such as chemical protein interaction for Chemprot) and task samples into ChatGPT and posing questions like "If the attributes of the text are viewed as genes, which genes are most important in the given chemical protein interaction extraction task?" Taking the Chemprot (Taboureau et al., 2010) dataset as an example, we received responses including *length, sentence structure, entity proximity, polarity*, and so on. Subsequently, following previous work (Yu et al., 2023), we adopt a human-AI collaboration scheme to empirically select the vital genes (listed in the Appendix A).

2.2 Active Parents Selection

In genetic algorithms, the objective is to select the fittest pair of parents for crossover and mutation, aiming to produce offspring with improved characteristics. This approach aligns well with the fundamental assumption of active learning, which posits that different samples have varying levels of importance and contribute differently to performance improvement. The standard pool-based active learning (Ren et al., 2021; Zhang et al., 2023) paradigm aims to select the most beneficial samples from an unlabeled data pool for annotation. Drawing inspiration from both methods, we adopt an active learning approach in our genetic algorithm framework. Specifically, we use the Euclidean distance between sample semantic representations [2] as a diversity criterion. This is used to select a previously unused pair of samples from the data pool as parents for the current iteration.

2.3 Crossover and Mutation

In each iteration, the genetic prompt uses the LLM as a simulator for the genetic algorithm to generate a new data sample. Specifically, a crossover operation is performed between the parents, allowing the new offspring sample to randomly inherit text genes from both parents. Subsequently, the offspring undergoes mutation on genes that were

[2] We use the Sentence Transformer (Reimers and Gurevych, 2019) to get the semantic representations.

Data	Year	Train	Dev	Test	Length	Class	Domain	Imbalance Ratio	Relation Type
Chemprot	2011	1,069	670	1,041	72.7	5	Biomedicine	7.16	Directed
DDI	2013	1,482	/	339	65.7	4	Pharmacology	8.08	Undirected
Semeval	2010	6,590	/	2,263	35.9	9	Web	1.99	Undirected
Conll04	2004	922	231	288	48.7	5	News	1.51	Directed

Table 1: Data Statistics

not inherited (the prompts corresponding to these two operations are shown in Figure 2).

Afterwards, unlike traditional genetic algorithms that would evaluate the fitness of the new sample and potentially discard it, our algorithm unconditionally adds the offspring to the population, as a small amount of noisy data can enhance the model's robustness (Wang et al., 2024; Li et al., 2021) by improving generalization and enabling the model to better handle real-world variations. This setup also avoids wasting generation costs.

3 Experimental Settings

3.1 Datasets

We have retrieved four public available relation extraction datasets, Chemprot (Taboureau et al., 2010), DDI (Herrero-Zazo et al., 2013), Semeval2010-task8 (Hendrickx et al., 2019) and Conll04 (Roth and Yih, 2004). Chemprot (Taboureau et al., 2010) is a biomedical dataset derived from PubMed abstracts, focusing on chemical-protein interaction relation extraction, containing sentences annotated with chemical and protein entities and their various fine-grained interaction types. We follow previous work (Peng et al., 2019), use the same 5 classes: Substrate, Upregulator, Downregulator, Agonist and Antagonist. We extract DDI (Drug-Drug Interaction) (Herrero-Zazo et al., 2013) dataset from the semeval-2013-task9 challenge, we filter all sentences that containing Drug-Drug Interactions and follow (Peng et al., 2019) to evaluate performance of the four DDI types. Semeval2010-task8 (Hendrickx et al., 2019) is a multi-Way classification task of Semantic Relations, which contains nine directed relations and the `Other` relation. To balance the relation types in the selected four datasets, We merge the 9 directed relation pairs into nine undirected symmetric relations and remove the `Other` ralation. Conll04 (Roth and Yih, 2004) is a relation extraction dataset consisting of news articles with annotations for four entity types and five relation types (Lives-In, Located-In, OrgBased-In, Work-For, and Kill). We utilize the standard training and test sets for evaluating the performance of five relation types. We summarize data statistics and splits in Table 1.

3.2 Technical Details

Following (Yu et al., 2023; Xu et al., 2024), we employed `GPT-3.5-turbo` (OpenAI, 2022) as our data generator, creating 5,000 samples for the Semeval dataset and 3,000 samples for each of the other three datasets. We set the temperature to 1 and p-value to 1, ensuring a diverse output distribution. Responses that did not adhere to the instructions, such as *'I am just a large language model.'* or *'I cannot do that for you.'* , were filtered out. Unless specified otherwise, we used `RoBERTa-large` (Liu et al., 2019) as our text classifier, training it with standard cross-entropy loss function. All fine-tuning processes were conducted with a learning rate of 1e-5 for 3 epochs, without utilizing any validation data. Our evaluation metrics included accuracy and macro-averaged F1 score, providing a comprehensive assessment of our method's performance across various datasets.

3.3 Baselines

Since our Genetic Prompt requires a small amount of real data to initialize the population, for a fair comparison, we compare the genetic prompt against two approaches that can be integrated with few-shot learning: 1) SimPrompt (Ye et al., 2022; Yu et al., 2023) and 2) AttrPrompt (Yu et al., 2023). For SimPrompt, we use its original class-conditional setting, but to ensure a fair comparison, we provide it with two random examples from the same class. Since AttrPrompt is also a method based on textual attributes, we adopted the same attributes as those used in Genetic Prompt. Following the settings in the original paper, we selected specific attribute values. Additionally, we also provided two samples from the same class each time.

4 Synthetic Data Analysis

Previous research (Roelofs et al., 2019; Li et al., 2024b) has demonstrated that low-diversity data often leads to neural network overfitting, highlighting diversity as a crucial factor in determining the scalability of synthetic data generation methods. Moreover, in an ideal scenario, synthetic data should faithfully reproduce the statistical characteristics of the real-world data it aims to emulate. A significant distribution shift between synthetic and real data can result in models trained on the former failing to generalize effectively to real-world tasks.

Therefore, we employ multiple metrics to quantify the semantic diversity of data generated by Genetic Prompt and baselines, as well as their distribution shift relative to the gold dataset. These metrics include average pairwise sample similarity (APS), inter-sample N-gram frequency (INGF) (Mishra et al., 2020), and Central Moment Discrepancy (CMD) (Zellinger et al., 2022), as shown in the Table 2. The gold dataset outperforms all three synthetic datasets across all diversity indicators, with the exception of INGF of Conll04, likely due to its small sample size. Regarding distribution shift, our Genetic Prompt achieves the lowest CMD scores across all datasets, demonstrating that its generated data most closely aligns with the distribution of the gold dataset. In terms of diversity, our Genetic Prompt significantly outperforms both baselines in Intra-class APS, Inter-Class APS, and INGF, providing evidence that Genetic Prompt can generate more diverse data. Additionally, we visualized the diversity of the datasets using cosine similarity and t-SNE (see Figure 4 and Figure 5 in Appendix). The visualization revealed that data generated by Genetic Prompt exhibited cosine similarity only slightly higher than that of the gold data, indicating a high level of diversity. Moreover, in the t-SNE plot, the distribution of Genetic Prompt's generated data closely matched that of the gold data.

5 Experiments

To evaluate the effectiveness of our Genetic Prompt method, we conducted two primary sets of experiments. First, we performed few-shot synthetic data generation trials across multiple classifiers and datasets, comparing our method directly with baseline approaches SimPrompt and AttrPrompt. We then conducted synthetic-gold data fusion experiments, where we merged equal proportions of synthetic data with the original training set to assess the impact on model performance.

Data	Method	APS ↓	Intra APS ↓	Inter APS ↓	INGF	CMD ↓
Chemprot	SimPrompt	0.534	0.736	0.483	2075.0	1.146
	AttrPrompt	0.577	0.659	0.556	4213.9	1.053
	Ours	**0.432**	**0.548**	**0.395**	**5712.5**	**0.757**
	Gold	0.326	0.352	0.314	8606.9	/
DDI	SimPrompt	0.687	0.783	0.655	1487.6	1.443
	AttrPrompt	0.596	0.658	0.574	3248.9	1.002
	Ours	**0.507**	**0.596**	**0.465**	**3344.4**	**0.930**
	Gold	0.397	0.410	0.390	7022.6	/
Semeval2010	SimPrompt	0.443	0.699	0.410	4619.3	1.513
	AttrPrompt	0.273	**0.347**	0.264	**19678.6**	1.271
	Ours	**0.271**	0.377	**0.257**	14189.4	**0.993**
	Gold	0.248	0.280	0.244	9176.4	/
Conll04	SimPrompt	0.445	0.748	0.369	859.9	1.074
	AttrPrompt	0.317	0.479	0.276	**5906.9**	0.596
	Ours	**0.264**	**0.437**	**0.220**	5235.7	**0.575**
	Gold	0.177	0.231	0.162	4995.5	/

Table 2: Comparison of three quantitative metrics on diversity: the average pairwise sample similarity (APS), inter-sample N-gram frequency (INGF) (Mishra et al., 2020) and Central Moment Discrepancy (CMD) (Zellinger et al., 2022) between the synthetic and gold data of different datasets. ↓ indicates that lower values of this metric are preferable. More details of these metrics in Appendix B.3.

5.1 Few-shot Synthetic Data Generation

To most directly compare the quality of data generated by Genetic Prompt and baselines, we ran three separate trials using different random seeds on multiple classifiers. The results are presented in the Table 3.

Overall, our proposed method consistently outperforms both SimPrompt and AttrPrompt across nearly all scenarios, showcasing its superiority in both accuracy (Acc.) and macro-average F1 score (ma-F1). This trend is evident across different model sizes, from `Distil-Roberta` (Sanh et al., 2019) to `Roberta-Base` and `Roberta-Large`, with performance generally improving as model size increases. Across all datasets and model sizes, our method achieves an average increase of 18.5% in accuracy and 18.3% in macro-F1 score compared to SimPrompt, and 12.1% and 11.1% respectively when compared to AttrPrompt.

5.2 Synthetic-Gold Data Fusion

To verify synthetic data's effectiveness in enhancing model performance, we conducted synthetic-gold data fusion experiments. We merged the synthetic dataset with the original training set in equal proportions and compared model performance when trained on this merged dataset versus the original set alone. The results of `Roberta-Large` are presented in the Table 4. Across all datasets, we observe that the fusion of synthetic data and gold data is beneficial for model performance in

Method	Chemprot		DDI		Semeval		Conll04	
	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1
Distil-Roberta								
SimPrompt	$49.0_{3.0}$	$38.5_{3.0}$	$53.1_{2.6}$	$46.8_{1.8}$	$45.3_{0.4}$	$39.4_{0.9}$	$43.5_{1.1}$	$40.3_{0.6}$
AttrPrompt	$61.5_{2.7}$	$50.0_{2.0}$	$50.1_{0.9}$	$49.8_{0.7}$	$53.3_{0.2}$	$50.7_{0.4}$	$52.9_{2.5}$	$49.6_{2.9}$
Ours	$\mathbf{69.5}_{1.8}$	$\mathbf{52.5}_{1.6}$	$\mathbf{61.1}_{1.3}$	$\mathbf{53.2}_{3.9}$	$\mathbf{62.7}_{0.5}$	$\mathbf{61.2}_{0.6}$	$\mathbf{70.5}_{3.2}$	$\mathbf{70.9}_{2.3}$
Roberta$_{Base}$								
SimPrompt	$53.0_{2.3}$	$41.2_{1.8}$	$56.7_{2.6}$	$52.1_{3.0}$	$43.7_{0.8}$	$38.5_{0.9}$	$44.5_{6.1}$	$41.5_{5.1}$
AttrPrompt	$67.4_{2.8}$	$\mathbf{54.2}_{1.1}$	$49.2_{5.6}$	$50.4_{5.5}$	$53.2_{0.5}$	$50.8_{0.9}$	$48.4_{2.6}$	$42.6_{3.2}$
Ours	$\mathbf{71.6}_{2.0}$	$53.6_{1.5}$	$\mathbf{65.1}_{1.7}$	$\mathbf{63.5}_{2.1}$	$\mathbf{64.4}_{0.2}$	$\mathbf{63.0}_{0.1}$	$\mathbf{70.4}_{1.6}$	$\mathbf{70.1}_{1.5}$
Roberta$_{Large}$								
SimPrompt	$65.4_{1.6}$	$55.3_{7.3}$	$56.1_{1.9}$	$60.6_{1.3}$	$48.0_{2.2}$	$43.0_{2.5}$	$42.4_{3.1}$	$38.4_{3.9}$
AttrPrompt	$74.1_{1.2}$	$\mathbf{62.2}_{1.2}$	$56.2_{2.3}$	$56.8_{2.0}$	$57.9_{0.9}$	$55.3_{0.8}$	$53.4_{0.2}$	$49.3_{1.2}$
Ours	$\mathbf{79.1}_{0.7}$	$59.7_{1.8}$	$\mathbf{71.6}_{2.0}$	$\mathbf{71.2}_{2.5}$	$\mathbf{67.2}_{0.4}$	$\mathbf{66.6}_{0.8}$	$\mathbf{69.9}_{1.6}$	$\mathbf{69.2}_{1.8}$

Table 3: Evaluation results on three different classifiers (standard deviation calculated using three random seeds)

most cases, particularly our Genetic Prompt, which achieved the highest improvement among the three methods, with an average increase of 1.85% in accuracy and 3.55% in macro-F1. Additionally, we observed that the macro-F1 score for ChemProt showed the most significant improvement. We hypothesize that this is due to its original dataset being the most imbalanced, while the synthetic data is completely balanced. This balanced augmentation likely enhanced the model's performance on minority classes, addressing the challenge of class imbalance in the original dataset. Therefore, we consider the fusion of data generated by Genetic Prompt framework with real data to be a simple and effective data augmentation scheme.

For the results of other classifiers, please refer to the Table 7 in Appendix.

6 Case Study and Ablation Study

6.1 Case Study: Performance over Different Synthetic Data Size

As mentioned in Section 4, the diversity of synthetic data is key to data scalability. Therefore, to examine the data scalability on different methods, we conducted experiments with one thousand, two thousand, and three thousand samples respectively, and visualized the results in Figure 3.

The depicted results reveal several noteworthy trends. Firstly, Genetic Prompt consistently outperforms both baselines across all three sample sizes in terms of accuracy and macro-F1 score. Secondly, we observe a consistent positive correlation between sample size and performance for Genetic Prompt, with performance metrics improving as the number of samples increases from 1,000 to 3,000. In contrast, both AttrPrompt and SimPrompt exhibit a counterintuitive trend, where performance occasionally declines with increased sample size. This phenomenon aligns with our findings in Section 4, which posited that samples generated by Genetic Prompt possess greater diversity, consequently leading to enhanced scalability.

6.2 Case Study: Genetic Prompt with GPT-4o

In line with prior research (Yu et al., 2023; Xu et al., 2024), our primary experiments were conducted using GPT-3.5-Turbo. However, to explore the potential impact of more advanced LLMs on the Genetic Prompt framework, we extended our investigation to include GPT-4o (OpenAI, 2024), the current state-of-the-art model. To ensure experimental consistency and facilitate direct comparison, we maintained identical hyperparameter configurations across both models in our experimental setup.

As illustrated in Table 5, the quality of data generated by Genetic Prompt using GPT-4o shows a substantial improvement compared to that produced by GPT-3.5-Turbo. This significant enhancement suggests that the full potential of Genetic Prompt has not yet been fully unleashed. On average, the use of GPT-4o resulted in an improvement of 8.2% in accuracy and 9.7% in macro-F1 score across all datasets. These notable gains indicate that the effectiveness of Genetic Prompt can be further amplified with advancements in Large Language Models (LLMs), pointing to a promising trajectory for future developments in this field.

Method	chemprot		DDI		semeval		conll04	
	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1
Gold	86.9	68.7	89.9	86.5	93.8	93.8	92.8	93.0
Gold + SimPrompt	88.5 (+1.6)	76.7 (+8.0)	89.2 (-0.7)	87.0 (+0.5)	94.8 (+1.0)	94.6 (+0.8)	95.6 (+2.8)	95.4 (+2.4)
Gold + AttrPrompt	87.2 (+0.3)	75.2 (+6.5)	90.4 (+0.5)	88.1 (+1.6)	94.3 (+0.5)	94.2 (+0.4)	95.9 (+3.1)	95.8 (+2.8)
Gold + Ours	89.1 (+2.2)	77.1 (+8.4)	90.7 (+0.8)	88.4 (+1.9)	94.2 (+0.4)	94.0 (+0.2)	96.8 (+4.0)	96.7 (+3.7)

Table 4: Performance comparison of four methods on various datasets.

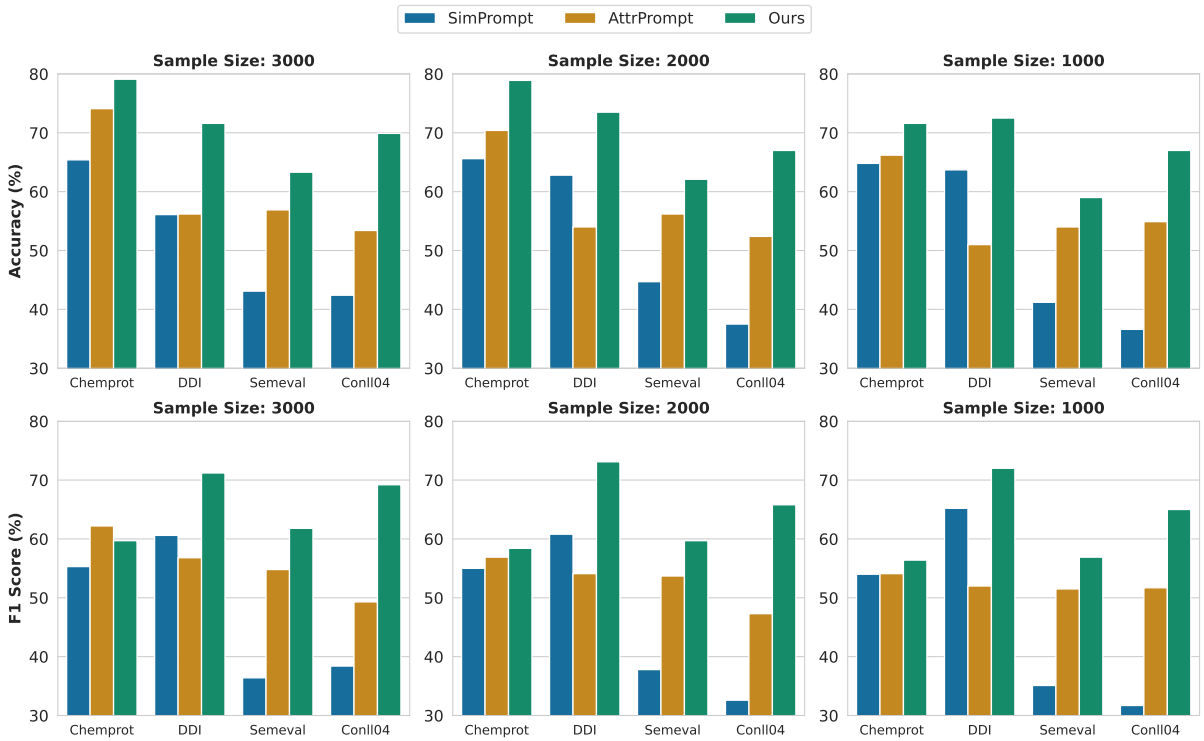

Figure 3: Impact of Sample Size on Model Performance.

Method	Chemprot		DDI		Semeval		Conll04	
	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1
GPT-3.5-Turbo	79.1	59.7	71.6	71.2	67.2	**66.6**	69.9	69.2
GPT-4O	**79.7**	**65.5**	**76.7**	**74.6**	**68.3**	66.1	**86.2**	**86.5**

Table 5: Comparison of Genetic Prompt performance using GPT-3.5-Turbo and GPT-4o.

6.3 Ablation Study

In order to prove the effectiveness of used active learning module, we conducted an ablation study that replace the AL-based parents selection algorithm with random parents selection.

As shown in Table 6, the AL-based method consistently outperforms the random selection in both accuracy and macro-F1 scores. Notably, for the DDI dataset, the AL-based method achieves a substantial improvement, increasing accuracy from 65.1% to 71.6% and macro-F1 from 64.2% to 71.2%. Similarly, for Conll04, the accuracy improves from 65.1% to 69.9%, and macro-F1 from 64.2% to 69.2%. These results demonstrate that incorporating active learning into parent selection significantly enhances data quality across all datasets.

Method	Chemprot		DDI		Semeval		Conll04	
	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1
Random	75.4	58.6	65.1	64.2	65.6	62.3	65.1	64.2
Ours	**79.1**	**59.7**	**71.6**	**71.2**	**67.2**	**66.6**	**69.9**	**69.2**

Table 6: Ablation of Active Learning module of Genetic Prompt framework, shown in percentages.

7 Related Work

7.1 Genetic Algorithm with Large Language Models

Large Language Models (LLMs) possess rich domain knowledge and text processing capabilities, while genetic algorithms excel at global search and iterative optimization. To leverage these complementary characteristics, the main collaboration paradigms include using LLMs to enhance genetic

algorithms or using genetic algorithms to enhance LLM's outputs (Wu et al., 2024; Petke et al., 2018). Specifically, these collaborative paradigms have been widely applied in areas such as code generation (Lehman et al., 2022), summary generation, and neural network architecture search (Chen et al., 2023). For example, LLMs can be evolution strategies (Lange et al., 2024) or evolutionary operators (Guo et al., 2024; Yang et al., 2024) for black-box prompt optimization. Further more, if sentences or paragraphs are considered as gene sequences(Liu et al., 2024), one can use a hierarchical genetic algorithm on a set of prompt prefixes to generate the optimal prompt prefix for specific task. Additionally, several applications have proved that LLM-GA collaboration paradigm can solve more complex problems in the real world, demonstrating its potential to address challenging issues beyond academic settings. For example, in the field of software development, it has been used for Python code generation (Lehman et al., 2022), while in creative industries, it has facilitated game design with user feedback serving as fitness evaluation (Lanzi and Loiacono, 2023). The paradigm has also shown promise in AI system optimization through its application in neural architecture search (Chen et al., 2023), and in scientific modeling by generating symbolic expressions(Meyerson et al., 2024).

However, these works share common limitations: they primarily use explicit token sequences and focus on finding a single optimal solution. In contrast, our Genetic Prompt diverges from these approaches by implicitly using textual attributes as genes, better leveraging the reasoning capabilities of LLMs. This also allows us to generate a large number of high-quality samples, successfully extending the LLM-GA collaboration paradigm to the field of synthetic data generation.

7.2 Synthetic Data Generation with Large Language Models

Recent advancements in the field of natural language processing have seen large language models (LLMs) play a pivotal role in synthetic text generation (Long et al., 2024). However, simply using LLM-synthesized data does not guarantee quality and diversity (Yu et al., 2023), and may potentially introduce biases inherent to the LLMs. To address these limitations, researchers have employed various techniques to enhance LLMs, including multi-step generation (Cui and Wang, 2024), in-context learning (Li et al., 2024a; Xu et al., 2024), label augmentation (Xiao et al., 2023), and conditional prompting (Yu et al., 2021; Russo et al., 2020; Logeswaran et al., 2018). For example, studies have explored methods to extract and predefine text attributes (Yu et al., 2023) or incorporate external knowledge (Xu et al., 2024) through semi-automatic interactive processes. These approaches utilize the extracted information as constraints to guide the data generation process. While such conditional prompt methods effectively raise the lower bound of text data diversity, they inadvertently introduce limitations. Specifically, the imposition of numerous constraints can potentially hinder LLMs' inherent ability to learn from context and accurately mimic the patterns and distributions found in real-world data.

In contrast, to alleviate these limitations of conditional prompt method, we propose the Genetic Prompt Framework for synthetic data generation, which takes a fundamentally different approach. Our method implicitly treats text attributes as gene sequences in genetic algorithms, leveraging LLMs' reasoning and in-context learning abilities to augment a small amount of real data. By applying crossover and mutation operations to this data, our framework can produce a large volume of highly diverse synthetic data that closely approximates real data distributions.

8 Conclusion

In this work, we introduced the Genetic Prompt Framework, a novel approach combining Large Language Models (LLMs) and Genetic Algorithms (GAs) for synthetic data generation. Our method treats text attributes as gene sequences and leverages LLMs' reasoning and in-context learning abilities to augment small real datasets. By using active learning to select semantically distant parent samples and adding offspring directly without fitness evaluation, we maximize text diversity while keeping generation costs low.

Experiments on four relation extraction datasets demonstrated the effectiveness of Genetic Prompt. It consistently outperformed baseline methods SimPrompt and AttrPrompt across multiple classifiers. Notably, Genetic Prompt showed superior data scalability and diversity, with performance improving as sample size increased. Analysis using metrics like APS, INGF, and CMD revealed that Genetic Prompt generated more diverse data with distributions closer to the gold dataset.

9 Limitations

Our study has two notable limitations. Firstly, like any data generation method, Genetic Prompt carries the risk of amplifying or introducing biases present in the seed data or the LLM's training data. This could potentially lead to the propagation or exacerbation of existing biases in the synthetic datasets. Secondly, due to budget constraints, we were unable to conduct experiments on a wider range of datasets. Our experiments were limited to four relation extraction datasets, which may restrict the generalizability of our findings across different domains or task types. Future work should address these limitations by investigating bias mitigation strategies and expanding the scope of experiments to diverse datasets and domains.

References

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Comput. Surv.*, 55(7).

Angelica Chen, David Dohan, and David So. 2023. Evoprompting: Language models for code-level neural architecture search. In *Advances in Neural Information Processing Systems*, volume 36, pages 7787–7817. Curran Associates, Inc.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 575–593, Toronto, Canada. Association for Computational Linguistics.

Wanyun Cui and Qianle Wang. 2024. Ada-instruct: Adapting instruction generators for complex reasoning. *Preprint*, arXiv:2310.04484.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *Preprint*, arXiv:1911.10422.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1555–1574, Singapore. Association for Computational Linguistics.

Robert Lange, Yingtao Tian, and Yujin Tang. 2024. Large language models as evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '24 Companion, page 579–582, New York, NY, USA. Association for Computing Machinery.

Pier Luca Lanzi and Daniele Loiacono. 2023. Chatgpt and other large language models as evolutionary engines for online interactive collaborative game design. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 1383–1390, New York, NY, USA. Association for Computing Machinery.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. 2022. Evolution through large models. *Preprint*, arXiv:2206.08896.

Junlong Li, Jinyuan Wang, Zhuosheng Zhang, and Hai Zhao. 2024a. Self-prompting large language models for zero-shot open-domain QA. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 296–310, Mexico City, Mexico. Association for Computational Linguistics.

Junnan Li, Caiming Xiong, and Steven C.H. Hoi. 2021. Learning from noisy data with robust representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9485–9494.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. 2024b. Entropic distribution matching in supervised fine-tuning of llms: Less overfitting and better diversity. *Preprint*, arXiv:2408.16673.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

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

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. *Advances in Neural Information Processing Systems*, 31.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11065–11082, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Elliot Meyerson, Mark J. Nelson, Herbie Bradley, Adam Gaier, Arash Moradi, Amy K. Hoover, and Joel Lehman. 2024. Language model crossover: Variation through few-shot prompting. *Preprint*, arXiv:2302.12170.

Swaroop Mishra, Anjana Arunkumar, Bhavdeep Sachdeva, Chris Bryan, and Chitta Baral. 2020. Dqi: Measuring data quality in nlp. *arXiv e-prints*, pages arXiv–2005.

Tom M Mitchell. 1980. The need for biases in learning generalizations.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt/. Accessed: 2023-07-24.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-10-15.

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

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65, Florence, Italy. Association for Computational Linguistics.

Justyna Petke, Saemundur O. Haraldsson, Mark Harman, William B. Langdon, David R. White, and John R. Woodward. 2018. Genetic improvement of software: A comprehensive survey. *IEEE Transactions on Evolutionary Computation*, 22(3):415–432.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM Comput. Surv.*, 54(9).

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. 2019. A meta-analysis of overfitting in machine learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Giuseppe Russo, Nora Hollenstein, Claudiu Cristian Musat, and Ce Zhang. 2020. Control, generate, augment: A scalable framework for multi-attribute text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 351–366.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Olivier Taboureau, Sonny Kim Nielsen, Karine Audouze, Nils Weinhold, Daniel Edsgärd, Francisco S. Roque, Irene Kouskoumvekaki, Alina Bora, Ramona Curpan, Thomas Skøt Jensen, Søren Brunak, and Tudor I. Oprea. 2010. ChemProt: a disease chemical biology database. *Nucleic Acids Research*, 39:D367–D372.

Jie Wang, Zili Wu, Minyan Lu, and Jun Ai. 2024. An empirical study on the effect of training data perturbations on neural network robustness. *Sensors (Basel, Switzerland)*, 24(15).

Xingyu Wu, Sheng hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Evolutionary computation in the era of large language model: Survey and roadmap. *Preprint*, arXiv:2401.10034.

Ruixuan Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. FreeAL: Towards human-free active learning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14520–14535, Singapore. Association for Computational Linguistics.

Ran Xu, Hejie Cui, Yue Yu, Xuan Kan, Wenqi Shi, Yuchen Zhuang, May Dongmei Wang, Wei Jin, Joyce Ho, and Carl Yang. 2024. Knowledge-infused

prompting: Assessing and advancing clinical text data generation with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15496–15523, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2251–2268.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Advances in Neural Information Processing Systems*, volume 36, pages 55734–55784. Curran Associates, Inc.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. 2022. Central moment discrepancy (cmd) for domain-invariant representation learning. In *International Conference on Learning Representations*.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023. LLMaAA: Making large language models as active annotators. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13088–13103, Singapore. Association for Computational Linguistics.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Comput. Surv.*, 56(11).

A Selected Textual Genes

The selection of genes for each dataset was tailored to capture the unique characteristics and requirements of their respective domains. Across all four datasets, common genes such as **sentence length** and **voice** were consistently included, reflecting their universal importance in text analysis. However, each dataset incorporated specific genes to address its particular focus. The Chemprot dataset, dealing with protein-chemical relations, included genes like **interaction verb** and **entity proximity** to capture the nuances of biochemical interactions. The DDI (Drug-Drug Interaction) dataset emphasized pharmacological aspects with genes such as **drug mentions** and **interaction verb**. For the more general relation classification tasks, both the SemEval2010 and CoNLL04 datasets shared a set of genes including **sentence structure**, **readability**, and **entity distance**, which are crucial for understanding the contextual relationships between entities.

B Diversity and Distribution Metrics

In this work, we use two metrics to evaluate the diversity and similarity within the datasets: Average Pairwise Similarity (APS) and Inter-Sample N-gram Frequency (INGF) (Mishra et al., 2020). We also use the Central Moment Discrepancy (CMD) (Zellinger et al., 2022) between the synthetic and gold data as the measure of distribution shift.

B.1 Average Pairwise Similarity (APS)

The APS measures the overall similarity between samples by averaging the cosine similarity between all embeddings. Given embeddings e_i and e_j for samples i and j, APS is defined as:

$$\text{APS} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \cos(e_i, e_j)$$

where N is the total number of samples and $cos(e_i, e_j)$ represent the cosine similarity between e_i and e_j.

We also compute Intra-Class and Inter-Class APS:

Intra-Class APS measures the similarity between embeddings within the same class:

$$\text{Intra-Class APS} = \frac{1}{|S|} \sum_{(i,j) \in S} \cos(e_i, e_j)$$

where S is the set of sample pairs with the same class label, $label_i = label_j$.

Inter-Class APS measures the similarity between embeddings from different classes:

$$\text{Inter-Class APS} = \frac{1}{|D|} \sum_{(i,j) \in D} \cos(e_i, e_j)$$

where D is the set of sample pairs with different class labels, $label_i \neq label_j$.

B.2 Inter-Sample N-gram Frequency (INGF)

The INGF quantifies the distribution of n-grams across samples. In our work, we include 1-gram, 2-gram and 3-gram in INGF computation:

$$\text{INGF} = \sum_i \frac{1}{\sigma\left(\frac{i(\nu)}{\text{size}(i)}\right)} \frac{(\nu_i - c)(d - \nu_i)}{\text{size}(i)}$$

where $i \in words, bigram, trigram$, ν represent the frequency, c and d are the minimum and maximum frequency bounds, and σ represents the standard deviation of normalized frequencies. size(i) refers to the total number of n-grams.

B.3 Central Moment Discrepancy (CMD)

Central Moment Discrepancy (Zellinger et al., 2022) measures the difference of probability distributions on a compact interval that considers higher order central moments. For our computations, we define the compact interval as $[-1, 1]^N$ and consider the first five central moments. The CMD between two distributions p and q, with corresponding samples $X = (X_1, \ldots, X_n)$ from p and $Y = (Y_1, \ldots, Y_n)$ from q, is calculated as:

$$CMD(p, q) = \frac{1}{2}\|\mathbb{E}(X) - \mathbb{E}(Y)\|_2 + \sum_{k=2}^{5} \frac{1}{2^k}\|\mathbf{c}_k(X) - \mathbf{c}_k(Y)\|_2$$

where $\mathbb{E}(X)$ and $\mathbb{E}(Y)$ represent the expectations (means) of the samples, and $\mathbf{c}_k(X)$ and $\mathbf{c}_k(Y)$ are the central moments of order k for X and Y, respectively. Each term $\mathbf{c}_k(X)$ is defined as:

$$\mathbf{c}_k(X) = \left(\mathbb{E}\left(\prod_{i=1}^{N}(X_i - \mathbb{E}(X_i))^{r_i}\right)\right)$$

where $r_1 + \cdots + r_N = k$, and $r_1, \ldots, r_N \geq 0$.

C Additional Results

C.1 Augmenting Gold Dataset with Synthetic Data

To better compare the performance of several methods, we present some experimental results on other classifiers on Table 7.

C.2 Synthetic Data Analysis

We visualized the synthetic data and real data using t-SNE and calculated the cosine similarity distribution of the data samples, which are shown in Figure 5 and Figure 4 respectively.

D Hardware and Software

All experiments are conducted on a workstation with an AMD Ryzen 9 7950X 16-core CPU, an NVIDIA RTX 4090 GPU with 24GB memory, 128GB of system RAM, running Ubuntu and utilizing the PyTorch (Paszke et al., 2019) framework. Besides, for our proposed Genetic Prompt framework, we utilize the FAISS library (Johnson et al., 2021) for faster calculation. For the Sentence Transformer used in our experiments, we utilize the 'stsb-roberta-large' checkpoint.

Method	chemprot		DDI		semeval		conll04	
	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1	Acc.	ma-F1
Roberta$_{Base}$								
Gold	82.3	61.2	83.0	77.6	90.5	90.0	86.9	87.2
Gold + SimPrompt	83.3 (+1.0)	65.6 (+4.4)	83.2 (+0.2)	78.9 (+1.3)	92.4 (+1.9)	92.0 (+2.0)	91.0 (+4.1)	91.3 (+4.1)
Gold + AttrPrompt	83.1 (+0.8)	70.2 (+9.0)	81.8 (-1.2)	77.1 (-0.5)	92.3 (+1.8)	91.8 (+1.8)	90.9 (+4.0)	91.2 (+4.0)
Gold + Ours	83.7 (+1.4)	70.0 (+8.8)	84.5 (+1.5)	80.3 (+2.7)	91.8 (+1.3)	91.5 (+1.5)	91.5 (+4.6)	91.8 (+4.6)
Distil-Roberta								
Gold	77.3	52.8	80.4	74.4	89.9	89.2	85.7	86.2
Gold + SimPrompt	83.9 (+6.6)	69.7 (+16.9)	81.9 (+1.5)	78.3 (+3.9)	89.9 (+0)	88.7 (-0.5)	88.9 (+3.2)	89.3 (+3.1)
Gold + AttrPrompt	84.6 (+7.3)	70.2 (+17.4)	81.7 (+1.3)	77.5 (+3.1)	89.6 (-0.3)	89.0 (-0.2)	89.0 (+3.3)	89.2 (+3.0)
Gold + Ours	85.4 (+8.1)	72.7 (+19.9)	83.7 (+3.3)	80.0 (+5.6)	90.3 (+0.4)	89.2 (+0)	89.9 (+4.2)	90.3 (+4.1)

Table 7: Performance comparison with growth (in green) and reduction (in red).

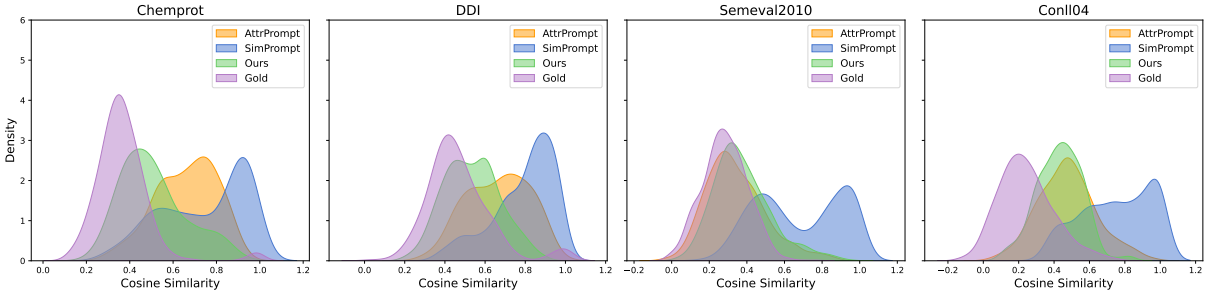

Figure 4: The distribution of cosine similarity of text pairs sampled from the same class.

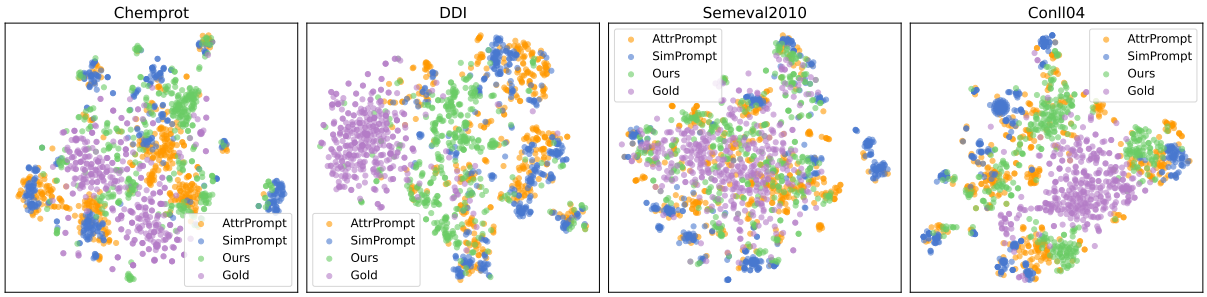

Figure 5: T-SNE of the four datasets