Transplant Then Regenerate: A New Paradigm for Text Data Augmentation

Guangzhan Wang[♠] **Hongyu Zhang**[♡] **Beijun Shen**[♠] **Xiaodong Gu**[♠*]

[♠] *Shanghai Jiao Tong University* [♡] *Chongqing University*

{wangguangzhan, bjshen, xiaodong.gu}@sjtu.edu.cn
hyzhang@cqu.edu.cn

Abstract

Data augmentation is a critical technique in deep learning. Traditional methods like Back-translation typically focus on lexical-level rephrasing, which primarily produces variations with the same semantics. While large language models (LLMs) have enhanced text augmentation by their "knowledge emergence" capability, controlling the style and structure of these outputs remains challenging and requires meticulous prompt engineering. In this paper, we propose *LMTransplant*, a novel text augmentation paradigm leveraging LLMs. The core idea of LMTransplant is *transplant-then-regenerate*: incorporating seed text into a context expanded by LLM, and asking the LLM to regenerate a variant based on the expanded context. This strategy allows the model to create more diverse and creative content-level variants by fully leveraging the knowledge embedded in LLMs, while preserving the core attributes of the original text. We evaluate LMTransplant across various text-related tasks, demonstrating its superior performance over existing text augmentation methods. Moreover, LMTransplant demonstrates exceptional scalability as the size of augmented data grows.

1 Introduction

Data augmentation is a critical technique in deep learning (Khosla and Saini, 2020; Shorten et al., 2021; Ding et al., 2022). Deep learning models are data-hungry and often suffer from limited datasets. Data augmentation generates additional training samples through transforming or rephrasing the existing dataset. This process increases data diversity, reduces the risk of overfitting, and enhances the models' generalization ability.

Data augmentation has been extensively studied in NLP tasks (Feng et al., 2021; Pellicer et al., 2023; Chen et al., 2023; Bayer et al., 2022). One simple prior approach (Wei and Zou, 2019) enhances text data through word-level transformations, such as random insertion and deletion. While easy to implement, it often generates low-quality samples that severely disrupt the semantic coherence of the generated text. Later, Back-translation (Sennrich et al., 2016) has been commonly used for sentence level rephrasing. Specifically, a translation model first translates the original text into a different language and then translates it back into the original language. Although the augmented text is semantically coherent, it often exhibits high similarity to the original text, leading to poor data diversity (Pellicer et al., 2023; Edunov et al., 2018).

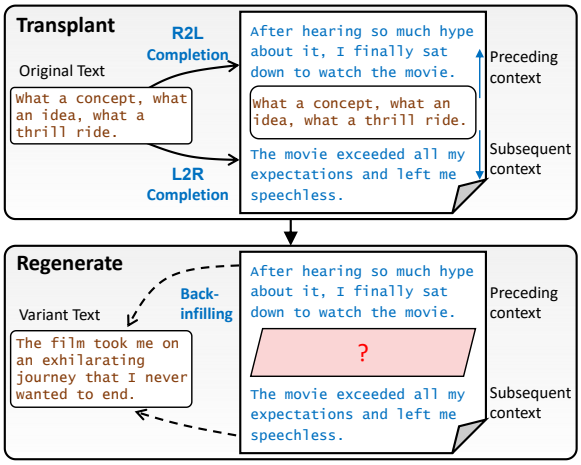


Figure 1: Illustration of LMTransplant.

Recent advancements in LLMs have spurred significant interest in LLM-based data augmentation methods (Whitehouse et al., 2023; Ding et al., 2024; Zhou et al., 2024; Qiao et al., 2024). A key characteristic of LLMs is their "knowledge emergence" capability, which stems from two factors: (1) the extensive prior knowledge acquired during pre-training, and (2) their robust language understanding and instruction-following abilities (Evuru et al., 2024; Ghosh et al., 2024). These strengths allow LLMs to generate desired outputs directly through demonstrations or natural language instruc-

*Correspondence: xiaodong.gu@sjtu.edu.cn

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13918–13932
November 4-9, 2025

tions without requiring additional training. A notable example is AugGPT (Dai et al., 2025), which instructs ChatGPT to rephrase text for improving text classification performance. However, this rephrasing-based method only generates variants with similar semantics and underutilizes the rich knowledge embedded in LLMs, limiting the diversity and creativity of the generated data. As a result, when augmented data volume reaches a certain threshold, further increasing may not yield performance improvements (Zhou et al., 2024). Although contextual augmentation methods can enhance content diversity, such as GPT3Mix (Yoo et al., 2021), which leverages the powerful few-shot learning capabilities of LLMs to generate mixed augmented text, it is highly sensitive to example quality and input order, potentially introducing uncontrolled biases that are not aligned with the original data distribution.

To address these limitations and better leverage the potential of LLMs, we propose LMTransplant, a novel text data augmentation paradigm. The core idea of LMTransplant is *transplant-then-regenerate* (TTR), namely, embedding the original text into an expanded context generated by LLMs and then instructing LLMs to regenerate a variant based on this enriched context. Specifically, LMTransplant employs bidirectional text continuation—backward (right-to-left) and forward (left-to-right)—to create the preceding and subsequent context of the original text. This original text is then masked within its expanded context, and LLMs are prompted to generate a replacement that introduces novel content diversity beyond rephrasing, while still preserving core attributes of the original text. Therefore, LMTransplant enhances both the diversity and creativity of the generated text, while maintaining alignment with the original data distribution.

We apply LMTransplant to various deep learning tasks, including text classification, question answering and named entity recognition (NER), and compare its performance with existing data augmentation methods. Experimental results demonstrate that LMTransplant can generate higher-quality augmented data. Training models with data augmented by our approach yields significant performance improvements across all tasks. Compared to non-augmentation, LMTransplant achieves accuracy gains of 28.16%, 19.96%, 7.68%, 23.66%, and 10.25% on the SST-2, TREC, SNIPS, MLQA, and CoNLL-2003 datasets, respectively.

In summary, our contributions are as follows:

- We propose a novel transplant-based paradigm for text data augmentation. Unlike existing methods which primarily focus on rephrasing, LMTransplant crafts content-level text variants, thereby crafting higher-quality augmented texts.
- We present a novel transplant and regeneration algorithm using bidirectional text continuation and masked text prediction. The algorithm allows for generating core attributes similar yet more diverse and creative text by effectively utilizing knowledge embedded in LLMs. Experiments demonstrate that, LMTransplant achieves significant performance improvements across different tasks.

2 Approach

2.1 Overview

We propose LMTransplant, a novel data augmentation paradigm. The core idea of LMTransplant is *transplant-then-regenerate*: integrating the original text into contextual scenarios generated by LLMs, and then asking LLMs to regenerate new variants given expanded contexts. This strategy allows the model to create content-level variants while preserving the core attributes of the original text. Figure 1 illustrates the entire augmentation process. For a given text, LMTransplant uses bidirectional text continuation to generate its preceding and subsequent contexts (Section 2.2). Subsequently, LMTransplant masks the original text in the transplanted text and asks the LLM to regenerate the missing parts given the crafted contexts, thereby producing new variants of the original text (Section 2.3). Each step of this process will be elaborated in the following sections.

2.2 Transplant

Given a seed text, we incorporate it into a relevant contextual scenario. Specifically, we treat the seed text as a fragment of a broader contextual passage, and then use an LLM to generate a semantically natural and logically coherent surrounding context. This process can be conceptualized as bidirectional text continuation, which involves two steps: (1) a forward (left-to-right) continuation process that continues writing the subsequent context of the seed text, and (2) a backward (right-to-left) continuation process that reconstructs the preceding context of the seed text.

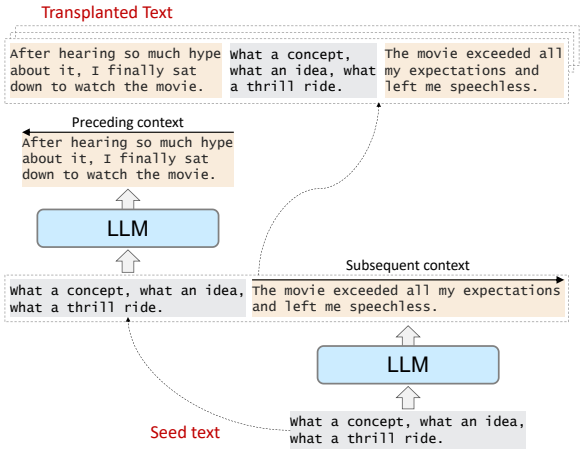


Figure 2: Illustration of text transplant.

Figure 2 illustrates the bidirectional text continuation process. First, the LLM generates a subsequent text that naturally extends the seed text, combining both to form an expanded passage. Next, the LLM generates a preceding text based on this expanded input. This process ensures that the generated context remains thematically aligned with the original text, while introducing novel information other than rephrasing through the knowledge embedded in LLMs, such as new words and expressions. Leveraging the language understanding and instruction-following capabilities of LLMs, we employ prompt engineering to guide content generation. The prompt template used for transplanting is provided in Appendix A.

2.3 Regeneration

The transplant phase generates multiple contextual scenes bearing the original text. We then introduce a *regeneration* process, where we prompt the LLM to regenerate new text variants that seamlessly integrate into the expanded context. Specifically, we provide the LLM with the crafted preceding and subsequent contexts, along with the original text, and ask it to generate a text that introduces content variation while preserving essential attributes of the original. Therefore, the regenerated text must satisfy the following criteria: (1) Fitting naturally within the surrounding context; (2) Aligning with the original text in terms of theme, length, format, and linguistic style, as mismatches in these aspects between training and testing instances are known to degrade downstream performance (Rogers et al., 2021); (3) Introducing novel elements to enrich content variation, avoiding simple rewording or direct replication.

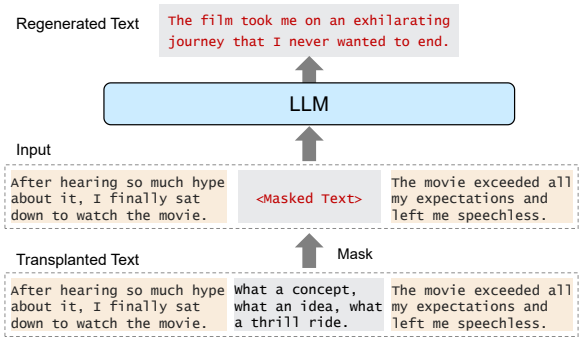


Figure 3: Illustration of regeneration.

In this step, the preceding and subsequent texts generated in Section 2.2 serve as a bridge, ensuring the newly generated text aligns with the original, such as theme and label-related information. Simultaneously, this process enriches and diversifies the content, making the regenerated text a high-quality augmentation of the original. The prompt design for this step is detailed in Appendix A.

3 Experiments

We conduct extensive experiments to evaluate the effectiveness of LMTransplant across multiple deep learning tasks by applying data augmentation to various datasets. Following established practices (Yoo et al., 2021; Dai et al., 2025; Ubani et al., 2023; Lee et al., 2024), we simulate low-resource scenarios by subsampling the training set of each dataset. Specifically, we select a subset of samples as seed data for augmentation, and then generate three augmented samples for each seed. This enables us to rigorously assess the performance of our method in data-scarce scenarios.

3.1 Implementation Details

To evaluate LMTransplant's robustness and effectiveness across a wider range of LLMs, we employ different LLMs as the base LLM for both transplant and regeneration phases, including DeepSeek-V3, GPT-3.5-Turbo and GPT-4o.

For text classification tasks, we use ModernBERT (Warner et al., 2024), a modernized bidirectional encoder-only BERT-style model available in two sizes: ModernBERT-base and ModernBERT-large. The classifier is initialized using pre-trained models from the Huggingface Transformers library (Wolf et al., 2019) and optimized with the AdamW optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2019). We set the learning rate to 4e-5 and maintain other hyperparameters consistent with

Task Type	Dataset	Classes	Train	Dev	Test
Classification	SST-2	2	6228	692	1821
	TREC	6	5406	546	500
	SNIPS	7	13084	700	700
QA	MLQA	-	1314	437	437
NER	CoNLL-2003	9	3234	748	679

Table 1: Statistics of datasets.

Fadaee et al. (2017), including 8 training epochs and a batch size of 8. During training, models are saved based on their performance on the development set, with the best-performing parameters retained for final evaluation on the test set.

For the question-answering task, we fine-tune Qwen2.5-1.5B (Team, 2024; Yang et al., 2024) using the AdamW optimizer with a learning rate of 1e-5, a batch size of 8, and 8 training epochs. And for NER task, we fine-tune ModernBERT using the same hyperparameters as in the classification task.

We repeat all experiments 10 times to mitigate the influence of randomness. Additionally, we conduct pairwise two-sample Wilcoxon Signed-Rank tests (Wilcoxon, 1992) to compare group medians and assess statistical significant differences. When $p < 0.01$, it indicates that there is a statistical significant difference between the two groups.

3.2 Datasets

We evaluate LMTransplant on five established benchmarks:

- **SST-2** (Socher et al., 2013): A widely used sentiment classification dataset of movie reviews, labeled as "positive" or "negative". We use the version provided by Wu et al. (2019)[1], which contains 6,228 training samples, 692 development samples, and 1,821 test samples.
- **TREC** (Li and Roth, 2002): A question classification dataset annotated with six types: "Abbreviation", "Entity", "Description", "Human", "Location", and "Numeric". Similar to SST-2, we use the version from Wu et al. (2019), with 5,406 training samples, 546 development samples, and 500 test samples.
- **SNIPS** (Coucke et al., 2018): A text classification dataset annotated with seven human intents: "AddToPlaylist", "BookRestaurant", "GetWeather", "PlayMusic", "RateBook", "SearchCreativeWork", and "SearchScreeningEvent". We use the version from SlotGated-SLU (Goo et al., 2018)[2], comprising 13,084 training samples, 700 development samples, and 700 test samples.
- **MLQA** (Lewis et al., 2019): A question-answering benchmark with context passages, questions, and answers. We use English samples form Huggingface[3] and filter out those exceeding 80 tokens to keep a modest length.
- **CoNLL-2003** (Tjong Kim Sang and De Meulder, 2003): A NER dataset contains four entity types: persons, organizations, locations, and miscellaneous names, tagged using the IOB scheme, resulting in nine distinct IOB labels.

Statistics of datasets are summarized in Table 1.

3.3 Metrics

We evaluate LMTransplant along two dimensions: the quality of the augmented texts (intrinsic evaluation) and their impact on deep learning tasks (extrinsic evaluation). For intrinsic evaluation, we adopt two widely employed metrics to assess the quality of the augmented samples:

Distinct-N measures the lexical diversity (Li et al., 2016). It is defined as the ratio of unique n-grams across all generated texts and their corresponding seeds.

$$\text{Distinct-N} = \frac{\#\text{ unique n-grams}}{\#\text{ all n-grams}} \quad (1)$$

A higher value indicates greater diversity. We calculate the average score across 10 experiments as the final result.

Semantic Variability measures how well the generated text extends the semantics of the seed texts. To assess this, we first adopt BERTScore (Zhang* et al., 2020) to calculate the sentence-level similarity between the generated text and its original, leveraging BERT's contextual embeddings. We then define semantic variability as:

$$\text{Semantic Variability} = 1 - \text{BERTScore} \quad (2)$$

Thus, the higher the semantic variability, the better the variability that the new text variants are.

For extrinsic evaluation, we adopt task-related metrics. Specifically, we use accuracy (Acc) and macro F1-score (Macro-F1) for classification and

[1]https://github.com/1024er/cbert_aug

[2]https://github.com/MiuLab/SlotGated-SLU/tree/master/data/snips

[3]https://huggingface.co/datasets/dkoterwa/mlqa_filtered

Method	SST-2		TREC		SNIPS		MLQA		CoNLL-2003	
	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑
Original	$0.99_{\pm 0.01}$	-	$0.94_{\pm 0.02}$	-	$0.89_{\pm 0.03}$	-	$0.97_{\pm 0.01}$	-	$0.97_{\pm 0.01}$	-
MoreData	$0.99_{\pm 0.01}$	-	$0.89_{\pm 0.01}$	-	$0.80_{\pm 0.02}$	-	$0.93_{\pm 0.02}$	-	$0.94_{\pm 0.01}$	-
EDA	$0.31_{\pm 0.02}$	$0.19_{\pm 0.02}$	$0.46_{\pm 0.02}$	$\underline{0.23}_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.40_{\pm 0.01}$	$\underline{0.21}_{\pm 0.01}$	$0.43_{\pm 0.01}$	$\underline{0.23}_{\pm 0.01}$
BackTrans.	$0.69_{\pm 0.03}$	$0.20_{\pm 0.01}$	$\underline{0.55}_{\pm 0.03}$	$0.13_{\pm 0.01}$	$\underline{0.60}_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.16_{\pm 0.06}$	$0.62_{\pm 0.03}$	$0.12_{\pm 0.01}$
GPT3Mix	$0.67_{\pm 0.03}$	-	$0.54_{\pm 0.02}$	-	$0.50_{\pm 0.04}$	-	$\underline{0.65}_{\pm 0.04}$	-	$0.18_{\pm 0.02}$	-
AugGPT	$0.54_{\pm 0.03}$	$\underline{0.24}_{\pm 0.01}$	$0.37_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.41_{\pm 0.02}$	$\underline{0.27}_{\pm 0.01}$	$0.38_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.33_{\pm 0.02}$	$0.22_{\pm 0.01}$
LLM2LLM	$\underline{0.71}_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.52_{\pm 0.03}$	$0.17_{\pm 0.01}$	$0.54_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.61_{\pm 0.01}$	$0.17_{\pm 0.01}$	$\underline{0.71}_{\pm 0.02}$	$0.20_{\pm 0.02}$
LMTransplant (ours)										
(left, right)	$\mathbf{0.88}_{\pm 0.03}$	$\mathbf{0.39}_{\pm 0.01}$	$\mathbf{0.66}_{\pm 0.03}$	$\mathbf{0.30}_{\pm 0.01}$	$\mathbf{0.63}_{\pm 0.02}$	$\mathbf{0.36}_{\pm 0.01}$	$\mathbf{0.72}_{\pm 0.03}$	$\mathbf{0.29}_{\pm 0.01}$	$\mathbf{0.78}_{\pm 0.03}$	$\mathbf{0.27}_{\pm 0.01}$
(right, left)	$\mathbf{0.88}_{\pm 0.03}$	$\mathbf{0.39}_{\pm 0.01}$	$0.64_{\pm 0.03}$	$0.29_{\pm 0.01}$	$\mathbf{0.63}_{\pm 0.02}$	$\mathbf{0.36}_{\pm 0.01}$	$0.71_{\pm 0.02}$	$0.28_{\pm 0.01}$	$0.75_{\pm 0.03}$	$\mathbf{0.27}_{\pm 0.02}$
Unidirectional	$0.82_{\pm 0.03}$	$0.37_{\pm 0.01}$	$0.54_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.50_{\pm 0.02}$	$0.31_{\pm 0.01}$	$0.63_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.66_{\pm 0.03}$	$0.22_{\pm 0.01}$

Table 2: Quality of generated samples by various methods ($p < 0.01$). Subscript numbers denote standard deviations. SV = Semantic Variability; MoreData: randomly samples additional data from the original training set as augmented data. Underlines indicate the second-best performance for each metric. We choose DeepSeek-V3 as the base LLM for data augmentation. Results based on GPT-3.5-Turbo and GPT-4o are available in Appendix B.

NER tasks, and the accuracy of answers (Acc) for QA tasks:

$$\text{Acc} = \frac{\#\text{samples answered correctly}}{\#\text{all test samples}} \quad (3)$$

$$\text{Macro-F1} = \frac{1}{N}\sum_{i=1}^{N} F1_i \quad (4)$$

where N is the total number of classes, and $F1_i$ represents the F1-score for the i-th class.

3.4 Baselines

We compare our method with both traditional and LLM-based methods:

Easy Data Augmentation (EDA) (Wei and Zou, 2019): A rule-based data augmentation method that applies lexical transformations, including synonym replacement, random insertion, random swap, and random deletion, to the original text.

Back Translation (BackTrans.) (Sennrich et al., 2016): A widely used date augmentation method that translates the original text into another language and then back-translates it into the original language to generate variants. Following ZeroShotDataAug (Ubani et al., 2023), we use *googletrans*[4] as the machine translation model, selecting different intermediate languages for multiple augmentations of the same text.

GPT3Mix (Yoo et al., 2021): This method randomly selects several examples from the seed samples, embeds them into a prompt template, and then leverages the powerful few-shot learning capabilities of LLMs to generate mixed augmented text influenced by the provided examples.

[4]Google Translate (https://pypi.org/project/googletrans/)

AugGPT (Dai et al., 2025): This method utilizes prompts to guide LLMs in rephrasing each sentence from the training samples into multiple semantically similar but linguistically different variants, thereby enhancing text classification performance.

LLM2LLM (Lee et al., 2024): An iterative data augmentation strategy that continuously employs LLMs to generate new samples from instances mispredicted by the downstream task model.

3.5 Results

3.5.1 Intrinsic Evaluation

The intrinsic evaluation results in Table 2 demonstrate that, the quality of augmented samples generated by LMTransplant significantly outperform other baselines across all benchmarks. In particular, LMTransplant achieves lexical diversity (Distinct-3) closer to original texts without augmentation (Original) and sampling additional data from original training set as augmented data (MoreData), highlighting its effectiveness in improving lexical diversity. Meanwhile, LMTransplant also exhibits superior semantic variability compared to other baselines, indicating that it meaningfully expands the semantics of the original text rather than merely relying on simple rephrasing. Additionally, LMTransplant with bidirectional text continuation outperforms its unidirectional counterpart, which serves as an ablation model for the bidirectional continuation strategy. Further ablation study details are discussed in Section 3.5.3.

3.5.2 Extrinsic Evaluation

We evaluate the effectiveness of LMTransplant in empowering deep learning tasks by training models on augmented data. To simulate low-data sce-

Method	SST-2		TREC		SNIPS		MLQA	CoNLL-2003	
	Acc↑	Macro-F1↑	Acc↑	Macro-F1↑	Acc↑	Macro-F1↑	Acc↑	Acc↑	Macro-F1↑
Original	$52.34_{\pm 3.19}$	$48.88_{\pm 6.59}$	$50.80_{\pm 10.60}$	$47.66_{\pm 9.06}$	$78.10_{\pm 2.77}$	$78.30_{\pm 2.68}$	$32.08_{\pm 1.02}$	$82.41_{\pm 1.31}$	$82.44_{\pm 1.14}$
MoreData	$65.41_{\pm 5.48}$	$65.41_{\pm 5.48}$	$74.30_{\pm 6.55}$	$70.75_{\pm 6.47}$	$88.23_{\pm 3.14}$	$88.30_{\pm 3.24}$	$43.30_{\pm 1.13}$	$91.56_{\pm 0.65}$	$91.50_{\pm 0.89}$
EDA	$56.78_{\pm 5.04}$	$55.79_{\pm 5.86}$	$53.50_{\pm 10.59}$	$50.58_{\pm 8.76}$	$81.86_{\pm 4.03}$	$81.80_{\pm 4.31}$	$36.92_{\pm 1.67}$	$85.08_{\pm 0.91}$	$83.89_{\pm 0.80}$
BackTrans.	$60.09_{\pm 6.25}$	$58.80_{\pm 8.16}$	$56.52_{\pm 6.57}$	$52.49_{\pm 6.37}$	$81.20_{\pm 3.86}$	$81.40_{\pm 3.49}$	$35.32_{\pm 1.24}$	$85.88_{\pm 1.01}$	$84.99_{\pm 1.07}$
GPT3Mix	$63.75_{\pm 6.76}$	$\underline{63.36}_{\pm 7.05}$	$57.68_{\pm 7.12}$	$53.39_{\pm 7.61}$	$81.80_{\pm 5.63}$	$82.15_{\pm 5.24}$	$33.18_{\pm 1.80}$	$87.92_{\pm 0.86}$	$86.75_{\pm 0.79}$
AugGPT	$61.11_{\pm 6.50}$	$60.55_{\pm 6.94}$	$\underline{58.94}_{\pm 9.43}$	$\underline{55.89}_{\pm 9.95}$	$82.73_{\pm 3.53}$	$82.98_{\pm 3.35}$	$36.04_{\pm 1.45}$	$87.27_{\pm 0.82}$	$86.68_{\pm 0.90}$
LLM2LLM	$\underline{63.90}_{\pm 5.34}$	$63.28_{\pm 5.44}$	$56.58_{\pm 7.17}$	$52.70_{\pm 6.76}$	$\underline{83.04}_{\pm 4.17}$	$\underline{83.33}_{\pm 4.14}$	$\underline{37.51}_{\pm 1.67}$	$\underline{88.04}_{\pm 1.08}$	$\underline{87.47}_{\pm 1.16}$
LMTransplant (ours)									
(left, right)	$\mathbf{67.08}_{\pm 6.92}$	$\mathbf{66.77}_{\pm 6.92}$	$60.88_{\pm 7.06}$	$56.38_{\pm 6.73}$	$84.06_{\pm 2.97}$	$84.24_{\pm 2.82}$	$39.44_{\pm 1.54}$	$90.28_{\pm 0.95}$	$88.82_{\pm 1.00}$
(right, left)	$66.21_{\pm 6.19}$	$65.34_{\pm 6.43}$	$\mathbf{60.94}_{\pm 8.59}$	$\mathbf{57.32}_{\pm 7.60}$	$\mathbf{84.10}_{\pm 2.82}$	$\mathbf{84.30}_{\pm 2.65}$	$\mathbf{39.67}_{\pm 1.78}$	$\mathbf{90.86}_{\pm 0.90}$	$\mathbf{89.07}_{\pm 1.09}$
Unidirectional	$64.34_{\pm 5.38}$	$63.52_{\pm 5.45}$	$58.42_{\pm 7.84}$	$55.06_{\pm 7.05}$	$82.53_{\pm 4.58}$	$82.78_{\pm 4.65}$	$35.69_{\pm 1.67}$	$87.04_{\pm 0.92}$	$85.96_{\pm 1.06}$

Table 3: Effectiveness of LMTransplant in empowering deep learning tasks (p <0.01). Subscript numbers denote standard deviations. We choose ModernBERT-base for classification and NER tasks and Qwen2.5-1.5B for QA task. Detailed results based on ModernBERT-large are available in Appendix C, Table 11.

narios, we adopt a subsampling strategy aligned with prior studies (Kumar et al., 2020; Ubani et al., 2023). Specifically, we randomly sample subsets (10 samples per class for classification tasks and 50 samples for QA and NER tasks) from the original training and development sets for each task. These training subsets are expanded using various augmentation methods, including LMTransplant. The augmented data, combined with the original sub-training set, are used to train deep learning models, and models' performance is evaluated on the original test set.

Results in Table 3 demonstrate that the augmented samples generated by LMTransplant significantly improve task performance. For instance, on the SST-2 dataset, LMTransplant increases classification accuracy from 52.34% to 67.08% and Macro F1-score from 48.88% to 66.77%, outperforming even MoreData. In the QA task on MLQA, LMTransplant achieves a 23.66% improvement in accuracy, significantly surpassing other LLM-based baselines. These findings indicate that, unlike methods that merely prompt LLMs, the transplanting mechanism in LMTransplant harnesses knowledge embedded in LLMs more effectively, thereby generating higher-quality augmented samples.

Time Efficiency: We also compare the time efficiency of different methods, which is defined as the average time required to generate a new sample (Table 4). To ensure fairness, all comparisons are conducted on the same hardware. EDA significantly outperforms other methods, due to its extremely simple rule-based operations. Among LLM-based methods, AugGPT is the fastest, attributed to its shortest prompts. LMTransplant is slower but still surpasses GPT3Mix and LLM2LLM. LLM2LLM exhibits the lowest efficiency, as its iterative generation and retraining process significantly increases time consumption.

Method	Time
EDA	$0.02_{\pm 0.84}$
BackTrans.	$3.44_{\pm 5.53}$
GPT3Mix	$22.78_{\pm 12.73}$
AugGPT	$4.93_{\pm 8.27}$
LLM2LLM	$30.55_{\pm 11.19}$
LMTransplant (ours)	$15.09_{\pm 13.87}$

Table 4: Time efficiency for various methods. We report the average processing time (in seconds) required to generate a new sample.

3.5.3 Ablation Study

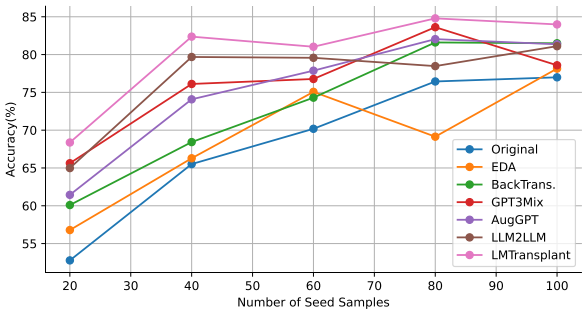
We hereby conduct ablation experiments to explore the key step in our approach—bidirectional text continuation. First, we alter the bidirectional text continuation strategy in the transplant step. Specifically, we remove backward continuation and retain only forward continuation (Unidirectional). We also experiment with altering the order of backward and forward continuations, testing both backward-first (left, right) and forward-first (right, left) approaches.

As shown in Table 3, unidirectional continuation significantly reduces effectiveness compared to bidirectional continuation, likely due to the limited context information it provides, resulting in less variation and diversity in the generated text. This is further supported by the obviously lower Distinct-3 score for unidirectional continuation in Table 2. Additionally, the order of bidirectional continuation also affects performance: forward-first continuation yields better results on most datasets. This is likely because, compared to backward continua-

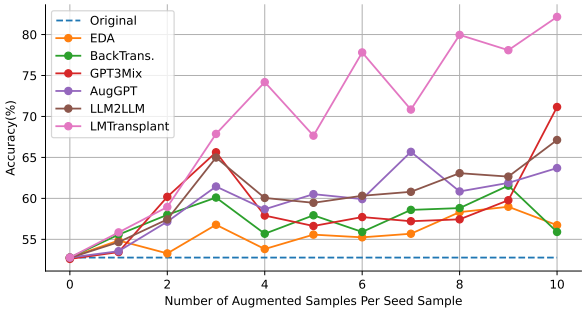
tion, LLMs are more proficient in forward continuation. Specifically, generating subsequent context first provides more information for LLMs to generate a higher-quality preceding context and thereby improve the quality of the augmented data.

3.5.4 Scaling Analysis

We further investigate the scaling effect of various sample sizes, including the number of seed samples and augmented samples. Experiments are conducted on the SST-2 dataset. First, with the number of augmented samples fixed at 3 per seed, we gradually vary the number of seed samples from 20 to 100 (10 to 50 per category). Next, with the total number of seed samples fixed at 20 (10 per category), we adjust the number of augmented samples per seed from 0 to 10. After each adjustment, models are retrained and evaluated. Results illustrated in Figures 4 demonstrate that, LMTransplant consistently achieves the highest accuracy across varying seed sample sizes (Figure 4a), showcasing its robust generalization under data scarcity. Notably, while other methods plateau after a few rounds of augmentation, LMTransplant continues to improve with additional augmented samples (Figure 4b), attributed to its effective utilization of knowledge embedded in LLMs. This underscores the exceptional scalability of LMTransplant and its efficacy in tackling data scarcity challenges.



(a) Performance under different numbers of seed samples.



(b) Performance under different numbers of augmented samples per seed sample.

Figure 4: Results of scaling analysis under various sample sizes.

3.5.5 Case Study

We further conduct a qualitative analysis of the augmented samples generated by LMTransplant. We highlight two cases in Table 5, where LMTransplant extends the diversity and creativity of the original samples while preserving the usability of the augmented samples.

In the first case, the original text is a question asking about the size of our galaxy. We can observe that baseline methods simply rephrase the original text with the same semantics. For example, EDA swaps the words "in" and "diameter", which even disrupts the linguistic integrity of the text. Back-translation and AugGPT rephrase the original text by employing machine translation models and LLMs, respectively. While they introduce changes in sentence structure, the new text remains semantically similar to the original. In contrast, LMTransplant generates more diverse and creative text centered on the topic of "our galaxy", by leveraging the knowledge embedded in LLMs. It expands the phrase "our galaxy in diameter" to "the size of our galaxy" in the preceding context. More surprisingly, it introduces a new term "Milky Way" to represent "our galaxy" within the same context. This example demonstrates the extraordinary creativity of LMTransplant in text augmentation.

In the second case, the original text is a movie review expressing the audience's appreciation on specific aspects of a movie. Despite the monotonous semantics, we note that the texts generated by baselines exhibit sentiment divergence from the original. For example, EDA replaces "idea" with "thought", slightly weakening the emotional intensity of the original text. Back-translation introduces the expression "alarming ride" during the translation process, shifting the sentiment from positive to negative or neutral. And AugGPT uses a more abstract expression, "exhilarating experience". In contrast, LMTransplant takes the gist of the original text and regenerates a novel review, expressing the same feelings on the theme of "movie". The subsequent text retains the positive sentiment with the phrase "exceeded all my expectations and left me speechless". The regenerated movie review aligns with the original text both thematically and sentimentally while bringing new knowledge about sentiment classification.

Overall, LMTransplant effectively harnesses the knowledge embedded in LLMs through its TTR

Original Text:	How big is our galaxy in diameter?	Numeric
EDA:	How big is our galaxy diameter in	Numeric
BackTrans.:	How great is our galaxy in diameter?	Numeric
AugGPT:	What is the diameter of our galaxy?	Numeric
LMTransplant:	Astronomers have long been fascinated by the size of our galaxy. What is the approximate number of stars within the Milky Way? The diameter of the Milky Way is estimated to be about 100,000 light-years.	Numeric
Original Text:	What a concept, what an idea, what a thrill ride.	Positive
EDA:	What a concept what an thought what a thrill ride	Positive
BackTrans.:	What a concept, what an idea, what alarming ride.	Positive
AugGPT:	What a concept, what an idea, what an exhilarating experience.	Positive
LMTransplant:	After hearing so much hype about it, I finally sat down to watch the movie. The film took me on an exhilarating journey that I never wanted to end. The movie exceeded all my expectations and left me speechless.	Positive

Table 5: Cases of augmented samples generated by various methods. The last column shows the classification label.

strategy, generating text with enhanced diversity and exceptional creativity that surpasses other augmentation methods. Simultaneously, by utilizing bidirectional context as a bridge, LMTransplant ensures the new text accurately retains the core characteristics of the original, guaranteeing its usability for downstream tasks.

4 Discussion

One concern is that LMTransplant might generate samples with semantic variations that differ from the original data (as illustrated in Table 5). This could potentially disrupt the performance of deep learning models, raising questions about the practical utility of the augmented samples.

However, our approach adheres to the fundamental principles of data augmentation: producing diverse, high-quality samples while preserving the original data distribution. By leveraging the bidirectional text continuation process, our method harnesses the knowledge embedded in LLMs to generate more creative texts. This not only enhances the diversity of the data but also improves the generalization capabilities of deep learning models. This hypothesis is supported by the results presented in Section 3.5.2, which demonstrate that the texts generated by LMTransplant significantly boost the performance of deep learning tasks.

Additionally, while the generated texts may exhibit semantic differences from the original, the transplanting mechanism in our approach ensures that the generated samples preserve the core attributes of the seed text—such as theme, linguistic style, and sentiment polarity. This mechanism effectively mitigates the risk of producing nonsensical or irrelevant text samples, thereby maintaining the integrity and usefulness of the augmented data.

5 Related Work

Data augmentation has been extensively explored in NLP tasks (Feng et al., 2021; Pellicer et al., 2023; Chen et al., 2023). Popular approaches such as word-level transformations (Wei and Zou, 2019) often disrupt semantic coherence, while Back-translation (Sennrich et al., 2016) offers limited diversity due to high similarity with the source text. In contrast, LMTransplant introduces a novel paradigm for text data augmentation, generating content-level variants that are more diverse and creative in content.

Recently, LLM-based data augmentation has gained wider attention (Whitehouse et al., 2023; Ding et al., 2024; Zhou et al., 2024; Qiao et al., 2024). Leveraging their "knowledge emergence" capability, LLMs can generate desired content directly through demonstrations or natural language instructions. For instance, AugGPT (Dai et al., 2025) uses natural language instructions to guide ChatGPT in rephrasing the original text. However, this approach applies relatively simple operations and underutilizes the extensive knowledge embedded in LLMs, resulting in limited content creativity and diversity. In comparison, LMTransplant simulates realistic contextual scenarios, enabling LLMs to better leverage their knowledge and generate more diverse and creative augmented samples.

Similarly, GPT3Mix (Yoo et al., 2021) leverages LLMs' few-shot learning capabilities by providing a set of examples to generate new samples. However, it is highly sensitive to example quality, selection strategy, and example order, potentially introducing uncontrolled biases that compromise

augmentation stability. More recently, LLM2LLM (Lee et al., 2024) iteratively augments instances misclassified by downstream-task model. While this approach produces more targeted samples, its iterative generation and retraining process incurs high computational costs and lacks adaptability to new datasets. In contrast, LMTransplant utilizes LLMs' powerful language understanding and instruction-following abilities through carefully designed prompts, eliminating the need for examples. This approach offers greater flexibility for adapting to different downstream tasks while mitigating issues related to data selection sensitivity and excessive computational resource consumption.

6 Conclusion

In this paper, we propose LMTransplant, a novel text data augmentation paradigm based on transplanting strategy. By leveraging bidirectional text continuation and masked text prediction, LMTransplant generates high-quality and diverse augmented text. It constructs contextually coherent scenarios aligned with the original text, fully utilizing the knowledge embedded in LLMs. Experimental results demonstrate that the augmented text generated by LMTransplant excels in diversity and creativity while significantly improving the performance of downstream tasks.

Our replication package is available at: https://github.com/W-GZ/LMTransplant.

Limitations

Although LMTransplant demonstrates strong experimental results, we acknowledge the following limitations and challenges that warrant further investigation: First, our experiments are conducted based on DeepSeek-V3, GPT-3.5-Turbo, and GPT-4o. We plan to extend our evaluation across a wider range of LLMs to further assess the robustness and effectiveness of LMTransplant. Second, in line with prior LLM-based augmentation methods (e.g., GPT3Mix, LLM2LLM), we evaluate LMTransplant on classification, question-answering and named entity recognition tasks, which we believe are representative tasks to demonstrate the generalization capability of LMTransplant. However, when applying LMTransplant to other task types, appropriate adjustments to its prompts may be necessary to ensure adaptability and effectiveness. As such, we plan to investigate its broader applicability in future work.

Acknowledgment

This research is funded by the National Key Research and Development Program of China (Grant No. 2023YFB4503802) and the Natural Science Foundation of Shanghai (Grant No. 25ZR1401175).

References

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

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces. *arXiv e-prints*, arXiv:1805.10190.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, et al. 2025. AugGPT: Leveraging ChatGPT for text data augmentation. *IEEE Transactions on Big Data*, pages 1–12.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1679–1705, Bangkok, Thailand. Association for Computational Linguistics.

Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. 2022. Data augmentation for deep graph learning: A survey. *SIGKDD Explor. Newsl.*, 24(2):61–77.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Chandra Kiran Evuru, Sreyan Ghosh, Sonal Kumar, Ramaneswaran S, Utkarsh Tyagi, and Dinesh Manocha. 2024. CoDa: Constrained generation based data augmentation for low-resource NLP. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3754–3769, Mexico City, Mexico. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Sreyan Ghosh, Utkarsh Tyagi, Sonal Kumar, Chandra Kiran Evuru, Ramaneswaran S, S Sakshi, and Dinesh Manocha. 2024. ABEX: Data augmentation for low-resource NLU via expanding abstract descriptions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–748, Bangkok, Thailand. Association for Computational Linguistics.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Cherry Khosla and Baljit Singh Saini. 2020. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, pages 79–85. IEEE.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. LLM2LLM: Boosting LLMs with novel iterative data enhancement. *arXiv preprint arXiv:2403.15042*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. MLQA: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, page 1–7, USA. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Lucas Francisco Amaral Orosco Pellicer, Taynan Maier Ferreira, and Anna Helena Reali Costa. 2023. Data augmentation techniques in natural language processing. *Appl. Soft Comput.*, 132(C).

Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, chengfei lv, and Huajun Chen. 2024. AutoAct: Automatic agent learning from scratch via self-planning. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the association for computational linguistics*, 8:842–866.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big data*, 8(1):101.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

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

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. ZeroShotDataAug: Generating and augmenting training data with ChatGPT. *arXiv preprint arXiv:2304.14334*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

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

Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in statistics: Methodology and distribution*, pages 196–202. Springer.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, and Morgan Funtowicz. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. In *Computational Science – ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part IV*, page 84–95, Berlin, Heidelberg. Springer-Verlag.

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

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yue Zhou, Chenlu Guo, Xu Wang, Yi Chang, and Yuan Wu. 2024. A survey on data augmentation in large model era. *arXiv preprint arXiv:2401.15422*.

Appendix

A Prompt Design

Here, we provide the prompts used for transplant and regeneration.

Prompt for Transplant: In the transplant process, we treat the original text as a fragment of a broader contextual passage and use LLMs to generate the relevant preceding and subsequent contexts. This process can be conceptualized as bidirectional text continuation, which consists of two steps: (1) forward continuation (left-to-right), where the model continues writing the subsequent context of the seed text, and (2) backward continuation (right-to-left) , where the model reconstructs the preceding context of the seed text.

Utilizing the powerful language understanding and instruction-following capabilities of LLMs, we employ prompt engineering to guide bidirectional text continuation. Below is the prompt template for our bidirectional text continuation process. First, we instruct LLMs to generate a subsequent text that seamlessly connects to the seed text while maintaining logical coherence. Then, the generated text is combined with the seed text to form an extended passage. Following this, we guide LLMs to generate a preceding text using this expanded input. Through this process, the generated context remains thematically aligned with the original text, while benefiting from enhanced expressiveness and enriched information through the knowledge embedded in LLMs.

Prompt for Transplant

```
Given the original <text_type>, generate a subsequent
sentence and a preceding sentence as follows:
Subsequent Sentence: Generate a sentence that can
naturally follow the original text, ensuring a smooth
transition and logical continuation.
New Text: Place the subsequent sentence behind the
```

Method	SST-2		TREC		SNIPS		MLQA		CoNLL-2003	
	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑
Original	$0.99_{\pm 0.01}$	-	$0.94_{\pm 0.02}$	-	$0.89_{\pm 0.03}$	-	$0.97_{\pm 0.01}$	-	$0.97_{\pm 0.01}$	-
MoreData	$0.99_{\pm 0.01}$	-	$0.89_{\pm 0.01}$	-	$0.80_{\pm 0.02}$	-	$0.93_{\pm 0.02}$	-	$0.94_{\pm 0.01}$	-
EDA	$0.31_{\pm 0.02}$	$0.19_{\pm 0.02}$	$0.46_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.23_{\pm 0.01}$
BackTrans.	$0.69_{\pm 0.03}$	$0.20_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.16_{\pm 0.06}$	$0.62_{\pm 0.03}$	$0.12_{\pm 0.01}$
GPT3Mix	$0.65_{\pm 0.02}$	-	$0.43_{\pm 0.02}$	-	$0.45_{\pm 0.02}$	-	$0.61_{\pm 0.02}$	-	$0.45_{\pm 0.03}$	-
AugGPT	$0.43_{\pm 0.02}$	$0.20_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.28_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.32_{\pm 0.01}$	$0.15_{\pm 0.01}$	$0.32_{\pm 0.02}$	$0.17_{\pm 0.01}$
LLM2LLM	$0.67_{\pm 0.02}$	$0.22_{\pm 0.02}$	$0.50_{\pm 0.02}$	$0.17_{\pm 0.01}$	$0.49_{\pm 0.01}$	$0.20_{\pm 0.01}$	$0.57_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.68_{\pm 0.03}$	$0.19_{\pm 0.02}$
LMTransplant (ours)										
(left, right)	$\mathbf{0.86}_{\pm 0.02}$	$\mathbf{0.37}_{\pm 0.01}$	$\mathbf{0.67}_{\pm 0.02}$	$\mathbf{0.28}_{\pm 0.01}$	$\mathbf{0.66}_{\pm 0.02}$	$\mathbf{0.35}_{\pm 0.01}$	$\mathbf{0.70}_{\pm 0.01}$	$\mathbf{0.27}_{\pm 0.01}$	$\mathbf{0.73}_{\pm 0.02}$	$\mathbf{0.24}_{\pm 0.01}$
(right, left)	$0.85_{\pm 0.02}$	$\mathbf{0.37}_{\pm 0.01}$	$0.66_{\pm 0.02}$	$0.27_{\pm 0.01}$	$\mathbf{0.66}_{\pm 0.02}$	$0.34_{\pm 0.01}$	$0.69_{\pm 0.02}$	$\mathbf{0.27}_{\pm 0.01}$	$0.72_{\pm 0.02}$	$\mathbf{0.24}_{\pm 0.01}$
Unidirectional	$0.81_{\pm 0.03}$	$0.35_{\pm 0.01}$	$0.58_{\pm 0.03}$	$0.25_{\pm 0.01}$	$0.57_{\pm 0.02}$	$0.31_{\pm 0.01}$	$0.60_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.59_{\pm 0.04}$	$0.22_{\pm 0.01}$

Table 6: Intrinsic evaluation results based on GPT-3.5-Turbo. Subscript numbers denote standard deviations.

Method	SST-2		TREC		SNIPS		MLQA		CoNLL-2003	
	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑	Dist-3↑	SV*↑
Original	$0.99_{\pm 0.01}$	-	$0.94_{\pm 0.02}$	-	$0.89_{\pm 0.03}$	-	$0.97_{\pm 0.01}$	-	$0.97_{\pm 0.01}$	-
MoreData	$0.99_{\pm 0.01}$	-	$0.89_{\pm 0.01}$	-	$0.80_{\pm 0.02}$	-	$0.93_{\pm 0.02}$	-	$0.94_{\pm 0.01}$	-
EDA	$0.31_{\pm 0.02}$	$0.19_{\pm 0.02}$	$0.46_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.30_{\pm 0.01}$	$0.16_{\pm 0.01}$	$0.40_{\pm 0.01}$	$0.21_{\pm 0.01}$	$0.43_{\pm 0.01}$	$0.23_{\pm 0.01}$
BackTrans.	$0.69_{\pm 0.03}$	$0.20_{\pm 0.01}$	$0.55_{\pm 0.03}$	$0.13_{\pm 0.01}$	$0.60_{\pm 0.02}$	$0.19_{\pm 0.01}$	$0.53_{\pm 0.01}$	$0.16_{\pm 0.06}$	$0.62_{\pm 0.03}$	$0.12_{\pm 0.01}$
GPT3Mix	$0.81_{\pm 0.02}$	-	$0.50_{\pm 0.03}$	-	$0.53_{\pm 0.01}$	-	$0.70_{\pm 0.02}$	-	$0.57_{\pm 0.02}$	-
AugGPT	$0.63_{\pm 0.03}$	$0.23_{\pm 0.01}$	$0.40_{\pm 0.03}$	$0.17_{\pm 0.01}$	$0.41_{\pm 0.02}$	$0.26_{\pm 0.01}$	$0.54_{\pm 0.03}$	$0.26_{\pm 0.01}$	$0.40_{\pm 0.02}$	$0.20_{\pm 0.01}$
LLM2LLM	$0.79_{\pm 0.02}$	$0.25_{\pm 0.01}$	$0.56_{\pm 0.02}$	$0.20_{\pm 0.01}$	$0.58_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.68_{\pm 0.01}$	$0.22_{\pm 0.01}$	$0.79_{\pm 0.02}$	$0.24_{\pm 0.01}$
LMTransplant (ours)										
(left, right)	$\mathbf{0.92}_{\pm 0.02}$	$\mathbf{0.46}_{\pm 0.01}$	$\mathbf{0.79}_{\pm 0.02}$	$\mathbf{0.35}_{\pm 0.01}$	$\mathbf{0.78}_{\pm 0.01}$	$\mathbf{0.43}_{\pm 0.01}$	$\mathbf{0.85}_{\pm 0.01}$	$\mathbf{0.38}_{\pm 0.01}$	$\mathbf{0.86}_{\pm 0.02}$	$\mathbf{0.27}_{\pm 0.01}$
(right, left)	$\mathbf{0.92}_{\pm 0.02}$	$0.45_{\pm 0.02}$	$0.77_{\pm 0.01}$	$0.32_{\pm 0.01}$	$\mathbf{0.78}_{\pm 0.02}$	$0.39_{\pm 0.01}$	$0.83_{\pm 0.01}$	$0.37_{\pm 0.01}$	$\mathbf{0.86}_{\pm 0.03}$	$\mathbf{0.27}_{\pm 0.01}$
Unidirectional	$0.86_{\pm 0.02}$	$0.45_{\pm 0.01}$	$0.67_{\pm 0.02}$	$0.29_{\pm 0.01}$	$0.71_{\pm 0.01}$	$0.36_{\pm 0.01}$	$0.76_{\pm 0.02}$	$0.35_{\pm 0.01}$	$0.76_{\pm 0.02}$	$0.25_{\pm 0.01}$

Table 7: Intrinsic evaluation results based on GPT-4o. Subscript numbers denote standard deviations.

```
original <text_type> to form a new text.
Preceding Sentence: Create a sentence that can
naturally precede the new <text_type>, making the
transition smooth and logical.

The original <text_type> is: <original text>

Now please return the generated subsequent sentence
and preceding sentence in the following format:
Preceding Sentence: [The generated preceding sentence]
Original Text: [The original <text_type>]
Subsequent Sentence: [The generated subsequent
sentence]
```

Prompt for Regeneration: The bidirectional text continuation phase generates multiple contextual scenes bearing the original text. However, these transplanted texts cannot be directly used as augmented data. Firstly, they all contain the same original text, which sacrifices the diversity of the augmented data. Secondly, these generated texts exhibit significant differences from the original text in various aspects, such as text length and format, making their differences from the original text too obvious. This is not ideal for data augmentation.

Therefore, we introduce a regeneration process, which uses LLMs to regenerate new text variants that seamlessly integrate into the expanded context. Specifically, we provide LLMs with the crafted preceding and subsequent contexts, along with the original text, and ask it to generate a new text. We implement this process using a prompt. The specific prompt template is as follows. In the prompt, we also specify some requirements for the new text generation: first, the new text should connect smoothly with the surrounding context, ensuring semantic consistency and logical flow; second, the new text should align with the original text in length, format, and linguistic style; and finally, the new text must not be a simple modification or direct copy of the original text. Instead, it should introduce novel elements, enhancing the richness and variation of the content. The prompt also explicitly instructs the LLM to adhere strictly to the original text's label category (denoted as "<label_type>").

Prompt for Regeneration

```
You are provided with three pieces of text:
1. Preceding Sentence: <preceding sentence>
2. Original <text_type>: <original text>
3. Subsequent Sentence: <subsequent sentence>

You are an expert in text data augmentation. Your task
is to generate a new <text_type> that can replace the
original <text_type> while meeting these requirements:
1. Fits naturally between the preceding sentence and
subsequent sentence, maintaining logical flow and
coherence.
2. Similar in text length, format (sentence pair,
etc.), and language style to the original <text_type>.
```

Method	SST-2	TREC	SNIPS	MLQA	CoNLL
Original	1.00	0.97	0.94	0.98	0.98
MoreData	0.99	0.94	0.87	0.95	0.96
EDA	0.35	0.53	0.35	0.43	0.47
BackTrans.	0.71	0.57	0.66	0.55	0.63
AugGPT	0.55	0.40	0.45	0.39	0.35
LLM2LLM	0.73	0.56	0.59	0.63	0.72
LMTransplant (ours)					
(left, right)	**0.89**	**0.70**	**0.74**	**0.75**	**0.82**
(right, left)	**0.89**	0.68	**0.74**	0.73	0.80
Unidirectional	0.83	0.58	0.59	0.65	0.67

Table 8: Compute Distinct-N per seed and average across seeds.

Method	SST-2	TREC	SNIPS	MLQA	CoNLL
Original	342.60	457.80	438.50	283.30	593.20
MoreData	1375.50	1687.90	1554.10	1117.40	2279.30
EDA	389.50	803.60	587.50	538.40	892.80
BackTrans.	867.30	935.10	1150.80	550.40	1340.80
GPT3Mix	927.80	1074.20	980.20	739.00	962.30
AugGPT	717.90	800.80	962.30	498.70	806.90
LLM2LLM	964.40	998.60	1070.50	791.50	1022.21
LMTransplant (ours)					
(left, right)	**1144.00**	**1384.10**	**1490.60**	**927.70**	1471.30
(right, left)	1129.80	1343.20	1466.90	896.60	**1534.00**
Unidirectional	1113.10	1105.10	1154.90	759.80	1172.60

Table 9: Numbers of unique n-grams for various methods.

```
3. Similar '<label_type>' to the original <text_type>,
which is '<original label>'.
4. The new <text_type> should not simply repeat the
original <text_type>.

Now please return the generated new <text_type> as
'Middle Sentence' in the following format:
Preceding Sentence: [The provided preceding sentence]
Middle Sentence: [The generated new <text_type>]
Subsequent Sentence: [The provided subsequent sentence]
```

B Intrinsic Evaluation

The intrinsic evaluation results based on GPT-3.5-Turbo and GPT-4o are provided in Table 6 and Table 7, respectively. These results closely align with those obtained based on DeepSeek-V3 (Table 2), demonstrating that LMTransplant consistently generates high-quality augmentations across different LLM architectures, highlighting its robustness and effectiveness.

We also explore alternative ways of computing Distinct-N and Semantic Variability to further validate the high quality of samples generated by LMTransplant.

Distinct-N: We calculate Distinct-N across all seeds and their corresponding augmentations. However, repeated n-grams across different seeds may affect the results. To address this, we compute Distinct-N per seed and then average across all seeds. As shown in the Table 8, this method generally yields higher scores for all augmentation methods, as it avoids counting repeated n-grams across different seeds. However, since the seeds themselves already exhibit low redundancy in n-grams (with Original's Distinct-N close to 1.0), the overall improvement is limited. This indicates that when the seeds already have low n-gram redundancy, computing Distinct-N per seed or globally makes little difference. Additionally, this method is not applicable to GPT3Mix, as each generated text corresponds to multiple seeds, so GPT3Mix was excluded from this comparison.

The ratio-based Distinct-N metric also has limitations—when seed texts already have high diversity, and augmented samples are diverse but larger in scale, the ratio-based metric may fail to distinguish which augmentation method has more advantages. To better capture this, we additionally compute the number of unique n-grams in the seeds, and in the combined set of seeds and augmented texts. As shown in the Table 9, LMTransplant consistently achieves significantly higher scores than other methods, approaching those of MoreData, highlighting its ability to introduce a substantial number of novel words and expressions.

Method	SST-2	TREC	SNIPS	MLQA	CoNLL
EDA	0.17	0.13	0.20	0.14	0.14
BackTrans.	0.09	0.10	0.11	0.07	0.15
AugGPT	0.12	0.09	0.12	0.09	0.05
LLM2LLM	0.15	0.11	0.14	0.10	0.12
LMTransplant (ours)					
(left, right)	**0.29**	**0.21**	**0.25**	**0.18**	**0.18**
(right, left)	**0.29**	**0.21**	**0.25**	0.16	0.17
Unidirectional	0.26	0.16	0.17	0.14	0.11

Table 10: The differences between samples generated per seed.

Semantic Variability: To capture differences between samples generated per single seed, we conduct an additional experiment: for each seed, we compute pairwise BERTScore among its three augmented texts. We average these pairwise scores per seed and then compute the mean across all seeds. To ensure reliability, we repeat this process over 10 independent trials and report the mean score. Finally, we define 1 - BERTScore as the metric to quantify the difference among augmented texts generated for the same seed. The results can be found in the Table 10.

Results show that LMTransplant consistently

Method	SST-2		TREC		SNIPS	
	Acc↑	Macro-F1↑	Acc↑	Macro-F1↑	Acc↑	Macro-F1↑
Original	$54.77_{\pm 3.57}$	$53.97_{\pm 3.97}$	$51.80_{\pm 7.44}$	$48.52_{\pm 6.17}$	$86.17_{\pm 2.88}$	$86.23_{\pm 2.91}$
MoreData	$71.57_{\pm 3.90}$	$71.12_{\pm 4.07}$	$72.08_{\pm 8.11}$	$68.90_{\pm 8.19}$	$91.53_{\pm 2.04}$	$91.67_{\pm 2.08}$
EDA	$59.69_{\pm 3.22}$	$58.72_{\pm 3.56}$	$57.22_{\pm 7.33}$	$54.08_{\pm 8.10}$	$\underline{88.87}_{\pm 2.31}$	$\underline{88.88}_{\pm 2.36}$
BackTrans.	$61.76_{\pm 4.44}$	$60.78_{\pm 5.41}$	$55.38_{\pm 8.72}$	$52.17_{\pm 8.03}$	$87.97_{\pm 2.93}$	$88.13_{\pm 2.78}$
GPT3Mix	$66.91_{\pm 2.67}$	$66.64_{\pm 2.68}$	$58.52_{\pm 7.49}$	$\underline{56.68}_{\pm 6.32}$	$88.19_{\pm 3.12}$	$88.11_{\pm 3.32}$
AugGPT	$63.62_{\pm 4.04}$	$62.79_{\pm 3.98}$	$55.82_{\pm 5.57}$	$53.86_{\pm 5.09}$	$87.10_{\pm 4.35}$	$87.36_{\pm 4.20}$
LLM2LLM	$\underline{67.38}_{\pm 3.07}$	$\underline{67.33}_{\pm 5.60}$	$\underline{59.56}_{\pm 8.90}$	$55.95_{\pm 8.45}$	$88.84_{\pm 1.94}$	$88.86_{\pm 1.91}$
LMTransplant (ours)						
(left, right)	$\mathbf{71.89}_{\pm 3.80}$	$\mathbf{71.47}_{\pm 4.29}$	$61.18_{\pm 5.70}$	$59.88_{\pm 5.65}$	$89.07_{\pm 4.25}$	$89.08_{\pm 4.66}$
(right, left)	$70.46_{\pm 4.89}$	$69.56_{\pm 5.40}$	$\mathbf{62.22}_{\pm 8.74}$	$\mathbf{60.37}_{\pm 8.52}$	$\mathbf{89.60}_{\pm 3.24}$	$\mathbf{89.70}_{\pm 3.38}$
Unidirectional	$68.19_{\pm 3.98}$	$67.66_{\pm 3.96}$	$56.46_{\pm 6.07}$	$54.19_{\pm 7.10}$	$88.03_{\pm 2.84}$	$88.23_{\pm 2.87}$

Table 11: Extrinsic evaluation results based on ModernBERT-large. Subscript numbers denote standard deviations.

achieves higher variability than other methods, further demonstrating its ability to generate diverse augmentations.

C Extrinsic Evaluation

The extrinsic evaluation results of text classification tasks based on ModernBERT-large are provided in Table 11. Similar to the results based on ModernBERT-base (provided in Table 3), the augmented samples generated by LMTransplant significantly improve task performance, surpassing other methods across all datasets. Meanwhile, bidirectional continuation outperforms unidirectional continuation, primarily because unidirectional continuation provides more limited contextual information, resulting in less variation and diversity in the generated text.

D Label Consistency

Ensuring the augmented text maintains label consistency is a key factor in determining the quality of augmented samples. LMTransplant incorporates two key mechanisms to preserve label consistency: (1) The bidirectional context continuation during the transplant phase serves as a bridge between the original and augmented text. This ensures that the augmented text accurately preserves essential attributes of the original, including label-related information. (2) Prompt constraints: in the regeneration phase, we employ explicit prompt constraints (as seen in Appendix A) to instruct the LLM to adhere strictly to the original text's label category (denoted as "<label_type>").

To further validate label consistency, we employ an LLM to predict the labels of augmented texts and compute the ratio of generated labels that match the original labels. The prompt is as follows:

Method	SST-2	TREC	SNIPS
Original	0.9500	0.6750	0.9543
MoreData	0.9363	0.7042	0.9325
EDA	0.9275	0.6192	0.9100
BackTrans.	0.8975	0.6333	0.9118
GPT3Mix	<u>0.9708</u>	0.6414	**0.9691**
AugGPT	0.9312	0.6771	0.9429
LLM2LLM	0.9425	<u>0.6916</u>	0.9214
LMTransplant (ours)			
(left, right)	0.9663	**0.7283**	0.9161
(right, left)	0.9688	0.7154	0.9354
Unidirectional	**0.9750**	0.7125	<u>0.9439</u>

Table 12: Label consistency for various methods.

Prompt for Label Prediction

```
You are an expert text classifier. Your task is to
analyze the given text and select exactly ONE most
appropriate label from the provided candidate label
list.

Text: {text}
Candidate label list: {self.label_enum_str(label_set)}

Please return ONLY the selected label name without
explanations, punctuation or additional text.
```

The results are shown in Table 12. We notice that LMTransplant effectively maintains label consistency in augmented texts. This demonstrates that LMTransplant can produce diverse and creative samples while preserving the core attributes of the original text, thereby maintaining the integrity and usefulness of the augmented data.

E Generate IOB Label for NER Task

For named entity recognition (NER) tasks, the IOB label sequence of each augmented text typically differs from that of the original, as the positions of named entities are likely to change during augmentation. Consequently, it is necessary to regenerate label sequences for the augmented texts. To achieve this, we employ a prompt-based approach:

we provide the original text and its corresponding label sequence to LLMs, and then prompt LLMs to generate the appropriate IOB label sequence for the augmented text. The prompt template is shown below:

Prompt for IOB Label Generation

```
You are a professional named entity recognition (NER)
annotation expert. Your task is to tokenize the given
sentence, identify the named entities, and assign a
corresponding BIO-format label and label ID to each
token.

This task includes only the following four types
of entities: persons (PER), organizations (ORG),
locations (LOC), and miscellaneous names (MISC).

Use the following BIO labels and their corresponding
label IDs: {'O': 0, 'B-ORG': 1, 'B-MISC': 2, 'B-PER':
3, 'I-PER': 4, 'B-LOC': 5, 'I-ORG': 6, 'I-MISC': 7,
'I-LOC': 8}.

Please output the result strictly in the following
format (only include these four lines):
sentence: original sentence
entities: ['token1', 'token2', ..., 'tokenN']
labels: [BIO_label1, BIO_label2, ..., BIO_labelN]
IDs: [label_id1, label_id2, ..., label_idN]

Here is an example:
sentence: {example_sentence}
entities: {example_entities}
labels: {example_labels}
IDs: {example_ids}

Now, please perform named entity recognition and
annotation for the following sentence:
{input_text}

Return only the result. Do not include any explanation
or additional content.
```