Towards Low-Resource Manchu Text Generation: Corpus Construction and Multi-Granularity Augmentation

Meiqi Wang[1], Zhiyong Xu[1], Xiaoxin Sun[2,4]*, Ruixin Yu [2,4], Zhiyong Guo [2,4], Zifan Wang [2,4], Shuqi You [2,4], Siyu Liu [2,4], Xiayu Qin [2,4], Suhua Wang[3]*

[1] School of Computer Engineering 、 School of Artificial Intelligence, Jiangsu Second Normal University,

[2] College of Information Science and Technology, Northeast Normal University,

[3] School of Computer, Changchun Humanities and Sciences College

[4] Northeast Normal University Manchu Intelligent Information Processing and Digital Qing History Laboratory

*Corresponding authors: sunxx772@nenu.edu.cn, wangsuhua@ccrw.edu.cn

Abstract

Manchu is a typical low-resource agglutinative language, for which existing digital resources are largely limited to character recognition and basic annotation, while sentence-level generation remains underexplored. In this paper, we study constrained sentence-level Manchu continuation. We construct the first context-target corpus for this continuation task, consisting of 3214 examples collected from Manchu Stories in Translation, Common Manchu Dialogues, and other sources, covering both narrative and conversational genres[1]. To improve robustness to rich suffixation, relatively flexible word order, and frequent OCR or transcription noise in Manchu, we propose a Multi-Granularity Data Augmentation (MGDA) strategy that expands training data through character-level noise injection, phrase rearrangement, and suffix perturbation, without relying on external dictionaries or translation models. Experiments on ByT5-small and Qwen2-1.5B show that MGDA substantially improves Manchu text generation performance: BLEU-4/ROUGE-L increases from 2.03/2.59 to 17.60/42.70 for ByT5, and from 1.98/6.42 to 14.12/41.49 for Qwen. Further ablation studies demonstrate the complementarity of the three augmentation granularities and suggest potential benefits of byte-level modeling in this experimental setting.

1 Introduction

Manchu was the "national language" of the Qing dynasty in China and was widely used in politics, military affairs, archival administration, and social communication during that period (Chao Ke, 2014). However, as its domains of use have disappeared, Manchu has become a severely endangered language (UNESCO, 2010). This sharp decline in active speakers contrasts with the large volume of extant Manchu documents and archives, many of which remain insufficiently organized, transcribed, and translated. Many Qing-era Manchu archives also contain damaged, incomplete, or noisy text, so contextual completion can support archival processing, corpus expansion, and digital teaching. Against this background, constructing sentence-level generation corpora for Manchu and developing text generation methods under low-resource conditions are important both for resource construction and for methodological exploration.

Current research on Manchu NLP is still at an early stage and faces two major challenges. First, high-quality Manchu corpora are extremely scarce. Existing datasets mainly focus on OCR-based word recognition (Wang et al., 2022b; Bi et al., 2026), while corpora for sentence-level semantic understanding and generation are nearly absent. Second, as an agglutinative language, Manchu expresses grammatical meanings such as case, tense, and aspect through rich suffixation, resulting in complex morphological variation and placing higher demands on models' sequence modeling abilities (Norman, 2003). In recent years, pretrained models, especially large language models (LLMs), have shown strong cross-lingual transfer capabilities (Brown et al., 2020), bringing new opportunities for low-resource language processing. Nevertheless, for the specific task of Manchu text generation, directly applying existing models still suffers from severe data sparsity: even with fine-tuning or in-context learning, the extremely small training scale makes it difficult for models to learn reliable lexical distributions and syntactic patterns.

Data augmentation is a common approach to mitigating low-resource problems. Existing methods include back-translation, synonym replacement,

[1] https://zenodo.org/records/20393618

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 546–559
November 6-10, 2026 ©2026 Association for Computational Linguistics

random perturbation, and generation-based augmentation with pretrained models (Li et al., 2021; Yang et al., 2025; Yang, 2024; Sennrich et al., 2016; Wei and Zou, 2019; Wang et al., 2022a). However, these methods are not fully suitable for Manchu text generation. Back-translation and synonym replacement require additional bilingual resources or dictionaries, which are also scarce for Manchu; generation-based augmentation depends on the model's prior knowledge of the target language, while general-purpose pretrained models provide limited coverage of Manchu; and ordinary character perturbation fails to capture key linguistic properties such as suffixal variation in agglutinative languages. Therefore, Manchu text generation requires a low-cost data augmentation method that does not rely on external resources and can explicitly account for morphological characteristics. To address these challenges, we study constrained sentence-level Manchu continuation. Given a Manchu sentence or dialogue turn as input, the model predicts the next sentence or turn. This setting is not open-ended creative generation, but a controlled context-to-continuation task. We construct the first sentence-level corpus for Manchu text generation and propose MGDA, a multi-granularity data augmentation strategy. MGDA expands training samples at three levels: characters, phrases, and morphological suffixes, aiming to improve model robustness to spelling noise, word order variation, and suffix variation. Our main contributions are as follows:

- We construct the first context-target corpus for constrained sentence-level Manchu continuation. We collect 59 stories and 60 dialogues from Manchu Stories in Translation, Common Manchu Dialogues, and other sources. After OCR extraction, manual correction, transcription normalization, and sample organization, we obtain 3214 high-quality examples covering both narrative stories and daily conversations.

- We propose MGDA, a multi-granularity data augmentation strategy for Manchu. Inspired by Manchu morphological, word-order, and transcription-noise characteristics, MGDA includes three augmentation operations: character-level random noise injection, phrase rearrangement, and suffix perturbation. It expands training data without relying on external dictionaries, back-translation models, or additional annotated resources.

- We evaluate the effectiveness of the proposed method on two representative base models. Using ByT5-small and Qwen2-1.5B, we systematically compare no augmentation, single-granularity augmentation, and MGDA. Experimental results show that MGDA substantially improves Manchu text generation performance. Ablation studies further verify the complementarity of the three augmentation operations and indicate that byte-level modeling may be beneficial under the current Manchu continuation setting.

2 Related Work

2.1 Current State of Manchu Natural Language Processing

Existing Manchu NLP studies mainly focus on document recognition and basic linguistic annotation. Word-level OCR datasets and recognition models have been developed for printed or handwritten Manchu texts (Wang et al., 2022b; Bi et al., 2026; Wang et al., 2024), and recent work has explored morphological annotation, named entity recognition, and part-of-speech tagging (Choi et al., 2023; Lee et al., 2024). For generation-oriented tasks, prior research is still limited. Manchu-Korean, Manchu-English and Manchu–Chinese translation studies have begun to explore parallel data and LLM-based transfer (Seo et al., 2023; Zhang et al., 2024; Pei et al., 2025; Wang et al., 2026), but sentence-level Manchu text generation remains under-resourced and lacks a dedicated corpus.

2.2 Data Augmentation Methods for Low-Resource Languages

Data augmentation is commonly used to reduce overfitting in low-resource settings (Li et al., 2021). Rule-based methods such as deletion, insertion, swap, synonym replacement, and back-translation are simple and effective (Sennrich et al., 2016; Wei and Zou, 2019; Karpukhin et al., 2019), while generative methods use pretrained models to create additional examples (Wang et al., 2022a; Devlin et al., 2019). Representation-level methods such as MixUp operate in feature space rather than directly editing text (Guo et al., 2019; Yang, 2024). However, many of these approaches require dictionaries, translation systems, or strong pretrained representations, which are unavailable for Manchu.



Figure 1: Covers of Manchu Stories in Translation (left) and Common Manchu Dialogues (right).

This motivates a resource-free augmentation strategy that directly reflects Manchu's spelling noise, word-order variation, and suffixal morphology.

3 Data Construction

3.1 Data Sources

The Manchu corpus constructed in this paper is primarily derived from the following sources: Manchu Stories in Translation (Zhuang, 1993) includes 40 folk stories with original Manchu text, Chinese translation and Roman transcription, from which we select 37 complete narrative pieces covering fables and folklore (Figure 1). Common Manchu Dialogues (Zhuang, 2006) provides 40 daily conversations aligned with Chinese texts and speaker annotations, composed of short conversational utterances (Figure 1). We also collect an additional 22 stories and 20 dialogues from online resources and other published books.

3.2 Data Preprocessing Process

Data Acquisition and Processing The raw data underwent a two-stage preprocessing pipeline. First, for PDF documents, we applied OCR followed by manual proofreading to correct recognition errors, missing special characters (e.g., ū, š), and segmentation issues. Editable texts from online sources were manually collected. Second, all texts were uniformly converted to the Taiqing transcription format. Detailed procedures are described in Appendix A.

3.3 Sample Construction

We construct samples as constrained sentence-level continuation pairs. For each source text, the current Manchu sentence or dialogue turn is used as the input context and the immediately following sentence or turn is used as the target continuation. Stories are segmented into sentences by two Manchu experts, while dialogues are segmented by speaker turns. The model input contains only the preceding Manchu sentence or turn, without Chinese translations or other auxiliary information; the context is first limited to one sentence or turn and then truncated according to each model's maximum input length.

Table 1: Corpus statistics. MSL/CSL denote average Manchu/Chinese sequence length in characters; MWL denotes average Manchu word length; MWS denotes average Manchu words per sequence; TTR denotes Type-Token Ratio, defined as the number of unique word types divided by the total number of word tokens.

Metric	Stories	Dialogues
Count	59	60
MSL	668	82.3
CSL	250	22.3
MWL	4.91	5.07
MWS	198.78	17.53
TTR	0.495	0.456

3.4 Corpus Statistics

After preprocessing, the corpus contains 3214 context-target samples from 59 stories and 60 dialogues. We use a source-level constrained split: samples from the same story or dialogue are assigned to a single partition. The training, validation, and test sets contain 2572, 321, and 321 samples, respectively. Table 1 summarizes the genre-level statistics.

The two genres differ substantially in length: stories contain much longer Manchu and Chinese sequences, while dialogues are shorter and closer to daily spoken usage. The similar MWL values indicate that the main difference lies in sequence length rather than word length. TTR (Type-Token Ratio) remains around 0.46–0.50 for both genres, suggesting moderate lexical diversity under a small-data setting.

4 Method

MGDA is designed as a robustness-oriented augmentation strategy inspired by Manchu morphology, word order, and OCR or transcription noise. It does not assume that every augmented input is a fully grammatical Manchu sentence. Instead, perturbations are applied only to the input context while the target remains the original clean continuation, so the model learns to predict clean Manchu

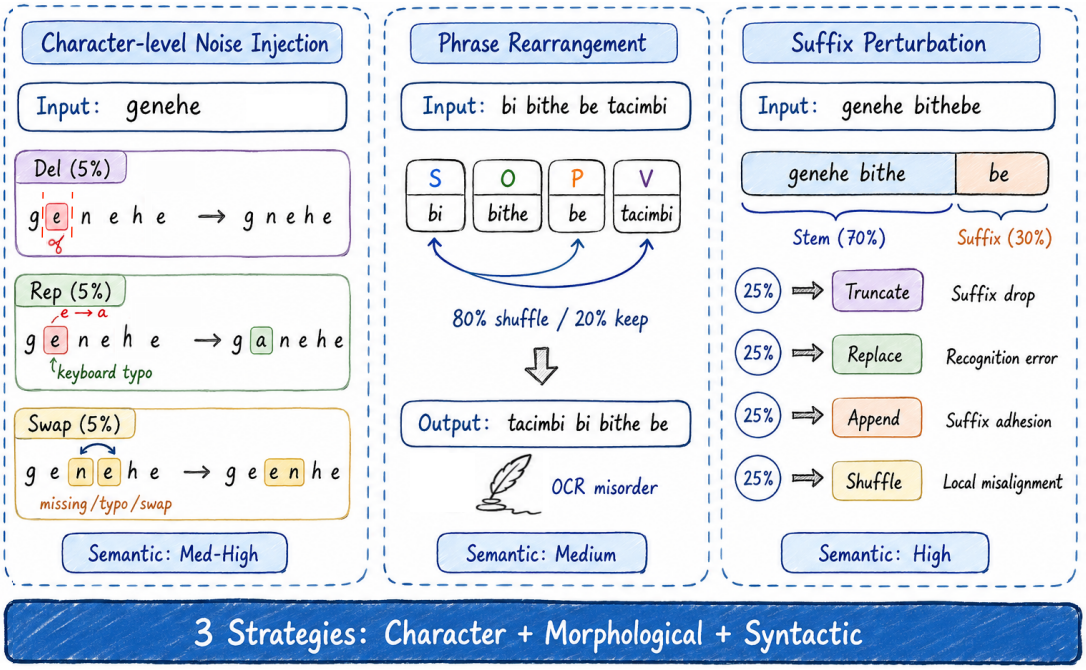


Figure 2: MGDA applies three separate input-side augmentation channels while retaining clean target continuations. The suffix channel uses a character-span heuristic, illustrated with the 70/30 split.

from controlled noisy contexts rather than to generate erroneous Manchu. As shown in Figure 2, Character-level Noise Injection applies deletion, replacement, and swap (each at 5% probability) to simulate input errors, Phrase Rearrangement shuffles phrases with 80% probability to model flexible SOV order and fragment displacement, and Suffix Perturbation perturbs a sentence-final suffix-oriented span through four operations to approximate morphological variation and word-final incompleteness.

4.1 Character-Level Noise Injection

During the collection and processing of Manchu texts, common noise types include character omission, mistyping, and local character swaps. We therefore treat a Manchu sentence as a character sequence and inject three low-probability perturbations: deletion, replacement, and adjacent swap.

Operation Selection For each character, we set $p_{\text{del}} = p_{\text{rep}} = p_{\text{swap}} = 0.05$. A random value determines whether the current character is deleted, replaced, swapped with the next character, or kept unchanged. Thus, each character has a total perturbation probability of $p_{\text{total}} = 0.15$, which introduces diverse noise while keeping the sentence recognizable.

Delete The current character is skipped and not added to the output sequence.

Replace The current character is replaced by a random English letter sampled from `string.ascii_letters`. This choice reflects real keyboard-input errors, where noisy characters are often Latin letters rather than valid Manchu characters.

Adjacent Swap If a next character exists, the current character and the next character are swapped and emitted together; otherwise, the current character is retained.

4.2 Phrase Rearrangement

Manchu has relatively flexible word order. Although SOV is the typical pattern, component displacement can occur in spoken language, rhetorical contexts, and noisy corpus construction. We therefore split each sentence into phrases and randomly reorder them to improve robustness to word-order variation. This operation simulates word-order variation and fragment displacement, but it does not guarantee that every reordered sentence is natural or fully grammatical Manchu.

Phrase Segmentation A configurable separator sep is used to split the sentence. In this experiment, sep = " ", so the granularity is the word or short-phrase level, consistent with Manchu Latin

Table 2: Examples of augmented samples

Augmentation Method	Augmented Input Context	Perturbation Description
Character(Deletion)	Batu taqikv de gene-he	Character a deleted
Character(Replacement)	Batu taqikv dX gene-he	Character e replaced with X
Character(Swap)	Batu taqikv de gnee-he	Characters e and n swapped
Phrase(Rearrangement)	de gene-he Batu taqikv	Word order shuffled
Suffix(Truncation)	Batu taqikv de gene-	Suffix -he truncated
Suffix(Replacement)	Batu taqikv de gene-re	Suffix he replaced with re
Suffix(Append)	Batu taqikv de gene-hehe	Extra characters he appended
Suffix(Shuffle)	Batu taqikv de ehegne-	Internal shuffle of suffix gene-he

transcription conventions. If only one phrase is obtained, reordering is skipped.

Random Reordering Phrase reordering is applied with probability $p_{\text{shuffle}} = 0.8$; otherwise, the original order is retained as an anchor. The reordered sentence is produced by joining the resulting phrase sequence with the same separator. This operation is intended to simulate order variation and fragment misplacement rather than guarantee grammaticality.

4.3 Suffix Perturbation

Manchu is an agglutinative language, where suffixes express grammatical meanings such as case, number, possession, tense, and mood. For example, the verb gene- (to go) can take the tense suffix -he to form gene-he (went), and the noun bithe (book) can take the case suffix -be to form bithe-be. Since word-final suffixes are especially prone to variation in noisy or incomplete records, we apply suffix-oriented sentence-final perturbation while preserving the earlier context.

Suffix Localization Let $S = (c_1, \ldots, c_L)$ be a Manchu sentence. We set $\alpha = 0.3$ and split it at $t = \lfloor L(1-\alpha) \rfloor$, yielding a stem prefix $S_{\text{pre}} = c_{1:t}$ and a suffix span $S_{\text{suf}} = c_{t+1:L}$. Because reliable Manchu morphological parsers and morpheme-level annotations are unavailable, this 70/30 split is an expert-informed heuristic that treats the sentence-final 30 percent character span as a suffix-oriented region; it is not strict word-level morphological parsing and does not guarantee morpheme-perfect suffix localization.

Truncation Randomly retain the first k characters of the suffix, where $k \sim \text{Uniform}(1, |S_{\text{suf}}|)$. This simulates word-final loss or OCR truncation.

Replacement For each suffix character, replace it with probability $p_{\text{rep}} = 0.3$ by a character uniformly sampled from V_{Manchu}, the character vocabulary extracted from the full corpus.

Append Append m random corpus characters to the suffix, where $m \sim \text{Uniform}(1, \lfloor |S_{\text{suf}}|/2 \rfloor)$. This simulates extra input or word-final concatenation.

Shuffle Randomly reorder the internal character sequence of the suffix. The final perturbed sentence is obtained by concatenating the unchanged stem and the perturbed suffix.

4.4 Augmentation Sample Examples

Table 2 illustrates how the three augmentation granularities modify the same Manchu sentence. The original sample is *Batu taqikv de gene-he* ("Batu went to school"). Colored spans indicate the edited parts. The examples show that Character edits create local spelling noise, Phrase rearrangement changes word order while preserving lexical content, and Suffix edits target word-final morphology. The augmented text is not treated as a gold output. The supervision signal is kept unchanged, so the model learns to generate the same clean target under controlled input-side surface variation.

4.5 Scale of Augmented Data

For each training sample, we generate two augmented variants, so the unfiltered augmented set has $|D_{\text{aug}}| = N \times (k+1)$, where $k = 2$. Starting from 2572 training samples, each single-granularity method produces 7716 instances before filtering. Table 3 summarizes data size (DS), total characters (TC), average sentence length (ASL), and unique samples (US).

Character augmentation yields the highest number of unique samples, while Phrase and Suffix generate more repeated forms because word order and suffix spans are more constrained. We further apply Sentence-BERT filtering with a cosine-similarity

Table 3: Statistics of three augmentation methods. DS denotes data size, TC denotes total characters, ASL denotes average sentence length, and US denotes unique samples.

Metric	Character	Phrase	Suffix
DS	7716	7716	7716
TC	585903	591806	605475
ASL	60.77	60.82	62.80
US	7372	5233	5020

threshold of 0.8 to remove redundant or semantically divergent samples, leaving 6819 training samples for the final MGDA setting.

The three augmentation operations are independent channels rather than a fixed sequential pipeline. Each operation is applied separately to the original input contexts to produce candidate augmented samples. After filtering, the retained candidates are merged with the original training samples to construct the final MGDA training set.

5 Experimental Setup

5.1 Base Models

We evaluate MGDA on two representative architectures: ByT5-small (Xue et al., 2022), a byte-level sequence-to-sequence model suitable for morphologically rich low-resource text, and Qwen2-1.5B (Yang et al., 2024), a general large language model adapted with LoRA (Hu et al., 2022). ByT5 is fully fine-tuned, while Qwen is trained with parameter-efficient LoRA adapters.

5.2 Evaluation Metrics

We report BLEU-1 to BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004). BLEU measures n-gram precision between generated and reference Manchu sentences, while ROUGE-L measures longest-common-subsequence overlap. All scores range from 0 to 100 and are computed on Taiqing transcription strings after whitespace tokenization.

5.3 Hyperparameter Settings

Both models are trained for 3 epochs unless a fixed update budget is specified below, using AdamW with a learning rate of 5e-5 and batch size 4. ByT5 uses maximum input/target lengths of 256 and beam size 5; Qwen uses maximum input/target lengths of 128, LoRA rank 8, LoRA alpha 32, and beam size 3. Full settings and hardware details are given in Appendix B.

5.4 Comparative Method Settings

We compare fine-tuning without augmentation, single-granularity augmentation using only Character, Phrase, or Suffix operations, and the full MGDA strategy combining all three. The final MGDA training set contains 6819 samples after semantic filtering. The original comparisons use three epochs, so update counts depend on training-set size. We add four controls targeting the nominal MGDA update budget: NoAug-SameSteps, Duplicate-SameSize, RandomOversample, and EDA-SameSize. All experiments use the same source-level train/validation/test split described in Section 3.4. Validation and test data are neither augmented nor resampled.

Training-budget controls The NoAug-SameSteps protocol repeats the original 2572 training examples to target the nominal MGDA update count. Duplicate-SameSize contains two complete copies of the original set plus 1675 original examples, giving 6819 entries. RandomOversample draws 6819 examples uniformly with replacement from the original training set. EDA-SameSize retains the 2572 originals and adds 4247 input-side variants using generic token deletion, swap, replacement, or insertion, with clean targets unchanged. The three size-matched controls use three epochs; other model and evaluation settings follow the original experiments.

The controls target the nominal MGDA update budget; counting assumptions and implementation details are given in Appendix D.

6 Experimental Results and Analysis

6.1 Main Experimental Results

Table 4 compares vanilla, fine-tuned, and MGDA-enhanced models on the test set.

Vanilla models perform poorly, confirming that pretrained models alone do not cover Manchu well. Fine-tuning improves BLEU-1 but still yields weak BLEU-4 and ROUGE-L. MGDA brings the largest gains for both backbones: ByT5 reaches 17.60 BLEU-4 and 42.70 ROUGE-L, while Qwen reaches 14.12 BLEU-4 and 41.49 ROUGE-L. ByT5 is slightly stronger under MGDA in our current setup, suggesting that byte-level modeling may be useful for character- and suffix-oriented perturbations in this task.

Table 4: BLEU-1 to BLEU-4 and ROUGE-L metrics for ByT5 and Qwen on the test set

Model & Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
ByT5 (Vanilla)	4.13	4.04	2.16	0.98	1.71
ByT5 (Fine-tuned)	23.28	11.09	8.33	2.03	2.59
ByT5-MGDA (Ours)	51.27	38.72	29.08	17.60	42.70
Qwen (Vanilla)	4.58	2.43	2.61	0.62	5.67
Qwen (Fine-tuned)	28.04	16.11	12.31	1.98	6.42
Qwen-MGDA (Ours)	49.78	37.37	29.22	14.12	41.49

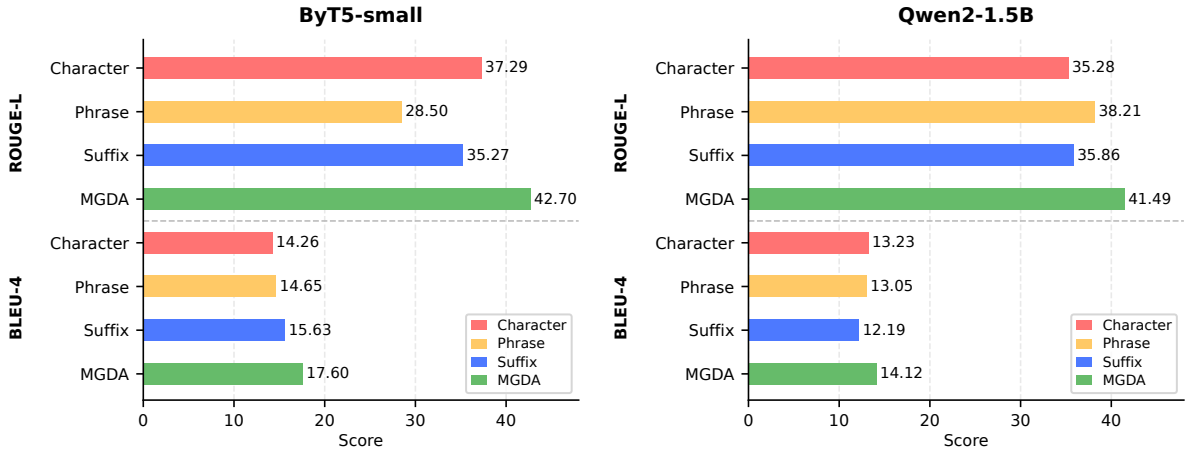


Figure 3: Single-granularity ablation results for Character, Phrase, and Suffix augmentation. Detailed scores are reported in Appendix C.

6.2 Training-Budget Controls

Results under the nominal-budget protocol suggest that longer training and repeated exposure explain part of the improvement over the original three-epoch baseline. NoAug-SameSteps increases BLEU-4/ROUGE-L from 2.03/2.59 to 10.91/30.36 for ByT5 and from 1.98/6.42 to 9.16/27.58 for Qwen. Thus, the original baseline-to-MGDA difference cannot be attributed entirely to augmentation. Table 5 reports all scores.

MGDA nevertheless outperforms all four budget controls on every reported metric for both models. Relative to NoAug-SameSteps, it gains 6.69 BLEU-4 and 12.34 ROUGE-L for ByT5, and 4.96 BLEU-4 and 13.91 ROUGE-L for Qwen. It also exceeds both duplication and random oversampling, suggesting that repeated exposure alone does not explain its performance.

EDA-SameSize is the strongest control on BLEU-4 and ROUGE-L. MGDA exceeds it by 3.74 BLEU-4 and 3.89 ROUGE-L for ByT5, and 0.97 BLEU-4 and 4.45 ROUGE-L for Qwen. These results support a benefit of MGDA beyond the generic token perturbations tested here, although the smaller Qwen BLEU-4 margin calls for cautious interpretation. Run-level variance and significance tests are unavailable for the controls, so these differences are descriptive.

6.3 Ablation Experiments

To understand the respective contributions and synergistic effects of the three granularities of augmentation methods, we designed three ablation experiments: comparison of single-granularity augmentation effects, ablation of internal operations within character-level augmentation, and the effect of training data size.

Comparison of Single-Granularity Augmentation Effects. Figure 3 compares character, phrase, suffix, and joint MGDA augmentation for both base models; detailed scores are listed in Appendix C.

As shown in Figure 3, all three single-granularity variants improve over the fine-tuned baseline, while MGDA achieves the best overall performance. Character augmentation mainly improves robustness to local spelling errors, Phrase augmentation exposes the model to word-order variation, and Suffix augmentation benefits morphological modeling; their combination achieves the highest BLEU-4 and ROUGE-L for both models.

Table 5: Nominal-budget controls on the test set. Fine-tuned uses three epochs; the added controls target the nominal MGDA update budget. Fine-tuned and MGDA scores are repeated for comparison.

Model & Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
ByT5 (Fine-tuned)	23.28	11.09	8.33	2.03	2.59
ByT5-NoAug-SameSteps	36.72	27.94	18.85	10.91	30.36
ByT5-Duplicate-SameSize	42.12	32.85	22.48	12.63	35.72
ByT5-RandomOversample	42.64	32.30	24.54	13.02	29.43
ByT5-EDA-SameSize	43.90	34.63	25.47	13.86	38.81
ByT5-MGDA	51.27	38.72	29.08	17.60	42.70
Qwen (Fine-tuned)	28.04	16.11	12.31	1.98	6.42
Qwen-NoAug-SameSteps	35.54	25.89	18.54	9.16	27.58
Qwen-Duplicate-SameSize	40.92	31.45	22.06	12.75	36.60
Qwen-RandomOversample	41.05	32.10	24.96	11.33	31.88
Qwen-EDA-SameSize	42.73	33.89	24.35	13.15	37.04
Qwen-MGDA	49.78	37.37	29.22	14.12	41.49

Ablation of Internal Operations of Character-Level Augmentation Character-level noise injection includes random deletion, random replacement, and adjacent character swap.

Figure 4 reports the effect of each operation and their combination; detailed scores are listed in Appendix C. The combined character augmentation variant achieves higher BLEU-4 and ROUGE-L than any single operation for both models, suggesting complementary benefits from deletion, replacement, and adjacent swap.

Effect of Training Data Size We compare the fine-tuned and MGDA settings at augmented training-set sizes of 500, 1000, 1500, and 2000 examples. These counts refer to training examples after augmentation, not to the number of original unaugmented examples. The Fine-tuned setting in this size ablation also uses augmented data and is distinct from the unaugmented full-training-set baseline in Table 4.

Figure 5 in Appendix C shows that performance generally improves as the augmented training set grows. MGDA improves ROUGE-L at every tested size for both models, while BLEU gains vary with the metric and data size. At 1000 examples, MGDA increases ROUGE-L from 12.30 to 19.23 for ByT5 and from 13.55 to 19.39 for Qwen. ByT5-MGDA has lower BLEU-4 than its fine-tuned counterpart at 500 and 2000 examples, whereas Qwen-MGDA improves BLEU-4 at all four tested sizes. These comparisons concern augmented training sets and should not be interpreted as an unaugmented data-scaling experiment. Detailed scores are reported in Appendix C.

Appendix E presents four representative Qwen-MGDA outputs and analyzes their successful continuations and remaining errors.

6.4 Discussion

The above experimental results validate the effectiveness of MGDA for the Manchu text generation task, while also revealing several issues worth further exploration.

Why does the byte-level model (ByT5) achieve a greater improvement under MGDA? In our current setup, ByT5 obtains slightly larger gains under MGDA, which may indicate that byte-level modeling is useful for character- and suffix-oriented perturbations. However, this comparison does not isolate tokenization from model architecture, parameter scale, pretraining data, or fine-tuning strategy, because ByT5-small and Qwen2-1.5B differ along all of these dimensions. Therefore, we treat the ByT5 result as a setting-specific observation rather than a general causal conclusion about byte-level models. Controlled comparisons are needed before drawing broader claims for morphologically rich low-resource languages.

How are the three granularities of MGDA complementary? Character-level augmentation primarily improves robustness to spelling errors; Phrase Rearrangement encourages invariance to word-order variation; and Suffix Perturbation encourages robustness to sentence-final morphological variation. Ablation experiments show that using any single granularity alone is inferior to the combination of all three, confirming that they address different difficulties of low-resource languages from different perspectives.

Comparison with existing low-resource data augmentation methods: Our method does not

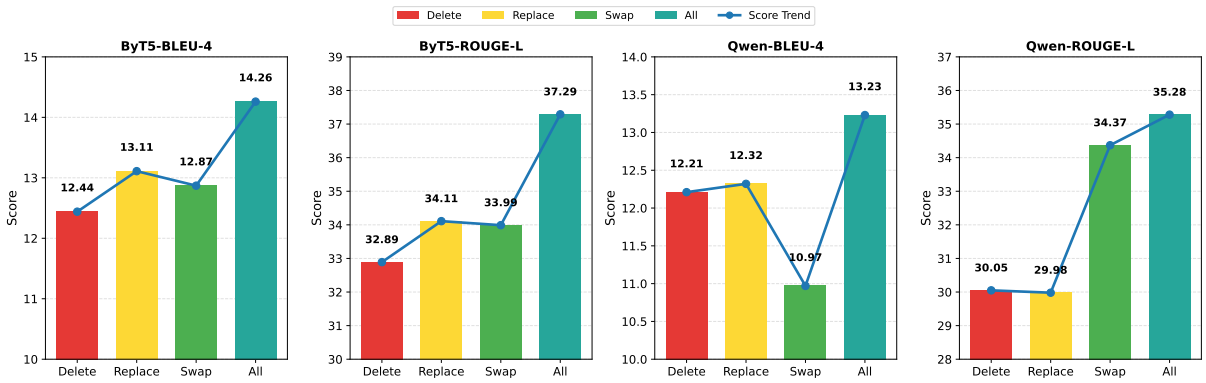


Figure 4: Ablation results for internal operations of character-level augmentation.

rely on bilingual dictionaries, back-translation models, or generative large models. It achieves augmentation solely through perturbations based on rules and morphological knowledge, offering advantages such as low computational cost, strong interpretability, and easy reproducibility. With only 3214 original samples, MGDA increases BLEU-4 from around 2 to 14–17 relative to the original three-epoch baseline. These gains reflect both augmentation and increased training exposure; the nominal-budget controls help assess their contributions.

7 Conclusion

This paper tackles constrained sentence-level Manchu continuation, a task with limited systematic resource support. We present the first sentence-level corpus for Manchu text generation and propose MGDA, a multi-granularity data augmentation strategy that operates without external dictionaries or translation models. Experimental results demonstrate that augmenting training data at three levels—character noise injection, phrase rearrangement, and suffix perturbation—effectively mitigates data sparsity. Ablation studies further confirm the complementarity of the three augmentation operations. Under the nominal-budget protocol, MGDA outperforms extended training, repeated sampling, and generic EDA. These results suggest benefits beyond repeated exposure, subject to verification of actual update counts. Together, this work provides a foundational corpus and a reproducible augmentation approach for Manchu text generation, while offering a reference for modeling other low-resource agglutinative languages. Future directions include corpus expansion, incorporation of finer-grained morphological information, controlled comparisons of tokenization and model architecture, and extension to Manchu-Chinese translation and other low-resource language tasks.

Acknowledgments

This work was supported by: (1) National Social Science Foundation in 2024 (No. 24BMZ101); (2) Jilin Provincial Science and Technology Development Program Project(No. 20250203109SF);(3) Major Project of Social Sciences Fund of Jilin Province (No.2023Z015).

Limitations

Despite the promising results, this study has several limitations. First, the corpus is relatively small (3214 samples) and limited to two genres (stories and dialogues), which may affect generalization to other text types. Second, our MGDA strategy relies on rule-based and morphology-inspired perturbations; while effective, it may not cover all possible noise patterns in real-world Manchu texts, and the 70/30 sentence-final suffix heuristic is not a substitute for reliable morpheme-level analysis. Third, the base models (ByT5-small, Qwen2-1.5B) are not specifically pre-trained for Manchu; larger or Manchu-adapted models could yield better performance. The comparison between ByT5 and Qwen also mixes tokenization with architecture, parameter scale, pretraining data, and fine-tuning strategy, so broader controlled experiments are required. Finally, systematic human evaluation of generated continuations is lacking; automatic overlap metrics may miss grammatical and semantic errors. For the new budget controls, actual update logs and run-level variability are unavailable, and matching optimizer updates does not ensure identical token-level compute. Future work will address these limitations by expanding the corpus, exploring gen-

erative augmentation, and designing task-specific model architectures.

Ethics/Data Statement

The corpus constructed in this paper is derived from publicly available published books and online sources. All data are used for non-commercial academic research purposes. No personal or sensitive information is involved. We encourage researchers using this corpus to respect the cultural heritage of the Manchu language and to apply the data responsibly.

References

Xiaojun Bi, Wenhao Tao, Zheng Chen, and Haipeng Sun. 2026. SCC3: A novel structure-connected cognition cube network for Manchu word recognition. *Expert Systems with Applications*, 297:129374.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Chao Ke. 2014. *Research History of Manchu Tungusic Language*. China Social Sciences Press, Beijing. In Chinese.

Woonho Choi, Sunghoon Jung, and Jeong Up Do. 2023. Construction of the Manchu corpus: focusing on Manwen laodang Taidzu. *ALTAI HAKPO*, (33):67–87.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with MixUp for sentence classification: An empirical study. Technical Report 1905.08941, arXiv.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Sangah Lee, Sungjoo Byun, Jean Seo, and Minha Kang. 2024. ManNER and ManPOS: Pioneering NLP for endangered Manchu language. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 11030–11039.

Hongzheng Li, Chong Feng, and Heyan Huang. 2021. A survey on low-resource neural machine translation. *Acta Automatica Sinica*, 47(6):1217–1231. In Chinese.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74–81.

Jerry Norman. 2003. The Manchus and their language (presidential address). *Journal of the American Oriental Society*, 123(3):483–491.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Renhao Pei, Yihong Liu, Peiqin Lin, François Yvon, and Hinrich Schuetze. 2025. Understanding in-context machine translation for low-resource languages: a case study on Manchu. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 8767–8788.

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*, pages 86–96.

Jean Seo, Sungjoo Byun, Minha Kang, and Sangah Lee. 2023. Mergen: The first Manchu-Korean machine translation model trained on augmented data. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning*, pages 118–124.

UNESCO. 2010. *Atlas of the World's Languages in Danger*, 3 edition. UNESCO Publishing, Paris.

Meiqi Wang, Xiaoxin Sun, Dongjie Wang, Ruixin Yu, Xiantao Heng, Shuo Wang, Zhen Huang, Peng Zhao, Suhua Wang, and Minghao Yin. 2026. ManCC: A task-anchored benchmark for Manchu–classical Chinese cross-lingual modeling. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 27271–27292, San Diego, California, United States. Association for Computational Linguistics.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022a. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 4242–4255.

Zhiwei Wang, Siyang Lu, Mingquan Wang, Xiang Wei, and Yingjun Qi. 2022b. AMRE: An attention-based CRNN for Manchu word recognition on a woodblock-printed dataset. In *Neural Information Processing: 29th International Conference, ICONIP 2022*, pages 267–278. Springer.

Zhiwei Wang, Siyang Lu, Xiang Wei, Run Su, Yingjun Qi, and Wei Lu. 2024. Learn more Manchu words with a new visual-language framework. *ACM Transactions on Asian and Low-Resource Language Information Processing*, page 3652992.

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*, pages 6382–6388.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

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, and 43 others. 2024. Qwen2 technical report. Technical Report 2407.10671, arXiv.

Jian Yang, Liu Sun, and Lifang Zhang. 2025. Survey on data processing and data augmentation in low-resource language automatic speech recognition. *Computer Science*, 52(8):86–99. In Chinese.

Lixia Yang. 2024. *Research on Data Augmentation and Feature Fusion for Low-Resource Languages Based on MixUp*. Ph.D. thesis, Yunnan Normal University, Kunming. In Chinese.

Kexun Zhang, Yee Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15654–15669.

Jifa Zhuang. 1993. *Manchu Stories in Translation*. Wen Shi Zhe Publishing House, Taipei.

Jifa Zhuang. 2006. *Common Manchu Dialogues*. Wen Shi Zhe Publishing House, Taipei.

A Data Preprocessing Details

Data Acquisition and Processing For PDF-format non-editable documents (such as the two books Manchu Stories in Translation and Common Manchu Dialogues), Umi-OCR_Paddle v2.1.5 is used for optical character recognition to convert image-based text into editable text. The OCR recognition parameters are set as follows: recognition language is traditional Chinese, confidence threshold is 0.7, and layout analysis is enabled to preserve paragraph structure and formatting. The OCR output is manually proofread sentence by sentence, with corrections focused on missing Manchu special characters (such as ū, š), character confusion, paragraph misalignment, sentence segmentation errors, and punctuation anomalies, ensuring text accuracy. Proofreading was conducted by three computer-science undergraduates, who compared OCR outputs sentence by sentence with the original PDF pages. Each proofreader completed one pass, and the identified issues were consolidated and corrected in a unified revision. For editable Manchu text directly obtainable from online sources, manual selection and item-by-item collection are used to supplement the corpus, ensuring diverse sources and content standards.

Conversion Manchu texts collected in different writing forms are uniformly converted to the Taiqing transcription format, completing character normalization and format unification. The cleaned and proofread corpus is then segmented into ordered sentences or dialogue turns and organized into context-target samples, where each input is the immediately preceding Manchu sentence or turn and each target is the following sentence or turn.

B Experimental Setup Details

This appendix reports the training configurations omitted from the main text for space. Table 6 lists the full fine-tuning setup for ByT5-small, while Table 7 lists the LoRA setup for Qwen2-1.5B. The two models share the same optimizer, learning rate, batch size, and default number of epochs (with the fixed-step exception described in the main text), but use different maximum sequence lengths and beam sizes because of their different model architectures and memory costs.

Table 6 uses full-parameter fine-tuning because ByT5-small is compact enough for the available V100 GPU. The relatively longer input and target

Table 6: ByT5-small hyperparameter settings

Hyperparameter	Setting
Optimizer	AdamW
Learning Rate	5e-5
Batch Size	4
Training Epochs	3
LR Scheduler	Linear decay (no warmup)
Max Input/Target Length	256 / 256
Decoding Method	Beam Search (beam=5)
Early Stopping	Validation loss, patience=5

length of 256 is used to preserve more context for byte-level modeling.

Table 7: Qwen2-1.5B LoRA hyperparameter settings

Hyperparameter	Setting
Optimizer	AdamW
LoRA Learning Rate	5e-5
Batch Size	4
Training Epochs	3
LoRA Rank (r)	8
LoRA Alpha	32
Max Input/Target Length	128 / 128
Decoding Method	Beam Search (beam=3)
Random Sampling	No (beam search)

Table 7 uses LoRA to reduce the number of trainable parameters for Qwen2-1.5B. The maximum input and target lengths are set to 128 to keep GPU memory usage stable, and beam size is reduced to 3 for decoding efficiency.

All experiments are conducted on a single NVIDIA Tesla V100 GPU with 32GB memory. The models use PyTorch and Hugging Face Transformers; exact software versions could not be verified from the available experiment records. In this setup, ByT5 full fine-tuning takes about 2 hours, while Qwen LoRA fine-tuning takes about 5 hours.

For evaluation, we use the NLTK library to compute automatic evaluation metrics including BLEU-1 to BLEU-4. This usage is consistent with the intended purpose of NLTK, which is designed for natural language processing research and evaluation. The original experiments report the mean across multiple independent runs. For the added budget controls, the available aggregate results do not specify the number of runs or their dispersion.

C Experimental Results

This appendix provides the numerical values behind the ablation figures discussed in Section 6.3. Table 8 corresponds to Figure 3, Table 9 corresponds to Figure 4, and Table 10 corresponds to Figure 5.

Table 8 shows that each single-granularity augmentation improves over the fine-tuned baseline in Table 4, while the joint MGDA setting achieves the strongest overall performance for both ByT5 and Qwen. This supports the claim that character-, phrase-, and suffix-level perturbations are complementary rather than interchangeable.

Table 9 further decomposes Character-level Noise Injection into deletion, replacement, and adjacent swap operations. The combined setting achieves the highest BLEU-4 and ROUGE-L for both models, suggesting complementary benefits from deletion, replacement, and adjacent swap.

Table 10 reports the scores for the augmented-data size ablation. All listed sizes count training examples after augmentation. The Fine-tuned rows therefore do not represent the unaugmented baseline in Table 4. MGDA improves ROUGE-L over the fine-tuned comparison at each size, but gains are not uniform across every metric.

D Training-Budget Control Details

The four controls use the original training split and the same model-specific settings: full fine-tuning for ByT5-small and LoRA for Qwen2-1.5B. The corresponding training files contain 2572 entries for NoAug-SameSteps and 6819 each for Duplicate-SameSize, RandomOversample, and EDA-SameSize. Inspection against original sample identifiers confirms that duplication and oversampling preserve both input and target, and that all EDA entries preserve the original target.

Duplicate-SameSize gives each original example two copies and gives 1675 examples a third copy. RandomOversample samples with replacement, so exposure frequencies vary across examples. The EDA file consists of the original 2572 examples and 4247 augmentation entries, including 32 whose input is unchanged. EDA uses generic whitespace-token operations without the sentence-final 70/30 heuristic or MGDA's three-channel design.

The implementation protocol specifies token deletion with probability 0.1, at least one token swap with the number otherwise set to ten percent of input length rounded down, and token replacement or insertion at a suggested rate of 0.1 using training-input vocabulary. It also calls for the same Sentence-BERT filtering rule as MGDA, with further candidates generated until the target size is reached. These are protocol specifications;

Table 8: Ablation results of individual modules

Model & Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
ByT5-Character	47.93	35.53	27.23	14.26	37.29
ByT5-Phrase	49.66	36.31	28.47	14.65	28.50
ByT5-Suffix	49.12	38.57	28.05	15.63	35.27
ByT5-MGDA (Ours)	51.27	38.72	29.08	17.60	42.70
Qwen-Character	46.71	32.70	21.59	13.23	35.28
Qwen-Phrase	48.12	36.83	25.88	13.05	38.21
Qwen-Suffix	48.06	35.94	29.54	12.19	35.86
Qwen-MGDA (Ours)	49.78	37.37	29.22	14.12	41.49

Table 9: Ablation results of character-level data augmentation

Model & Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
ByT5-Delete	46.10	32.06	20.78	12.44	32.89
ByT5-Replace	45.89	33.78	20.49	13.11	34.11
ByT5-Swap	44.57	30.02	21.27	12.87	33.99
ByT5-All	47.93	35.53	27.23	14.26	37.29
Qwen-Delete	42.13	32.82	21.16	12.21	30.05
Qwen-Replace	43.94	29.35	19.00	12.32	29.98
Qwen-Swap	40.37	29.31	19.96	10.97	34.37
Qwen-All	46.71	32.70	21.59	13.23	35.28

Table 10: Detailed results for different augmented training-set sizes. Aug. size is the number of examples after augmentation. Fine-tuned also uses augmented data here and differs from the unaugmented baseline in Table 4.

Aug. size	Model & Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L
500	ByT5 (Fine-tuned)	12.36	3.94	3.53	2.24	2.81
500	ByT5-MGDA (Ours)	12.77	7.58	3.96	2.15	3.13
1000	ByT5 (Fine-tuned)	20.68	14.40	11.40	7.47	12.30
1000	ByT5-MGDA (Ours)	23.41	18.39	8.01	9.73	19.23
1500	ByT5 (Fine-tuned)	34.01	24.82	14.29	10.35	23.92
1500	ByT5-MGDA (Ours)	36.00	27.41	15.34	10.87	25.34
2000	ByT5 (Fine-tuned)	43.32	31.39	25.45	13.86	28.82
2000	ByT5-MGDA (Ours)	43.82	32.74	26.61	12.50	31.03
500	Qwen (Fine-tuned)	10.83	7.42	2.12	1.08	4.25
500	Qwen-MGDA (Ours)	11.26	5.13	2.93	2.49	4.79
1000	Qwen (Fine-tuned)	20.58	11.37	6.16	4.43	13.55
1000	Qwen-MGDA (Ours)	20.07	13.30	7.43	4.70	19.39
1500	Qwen (Fine-tuned)	25.01	23.56	15.66	7.78	20.90
1500	Qwen-MGDA (Ours)	27.16	26.53	19.02	9.53	23.01
2000	Qwen (Fine-tuned)	37.34	31.52	22.04	12.26	26.78
2000	Qwen-MGDA (Ours)	39.39	35.91	25.85	12.77	29.11

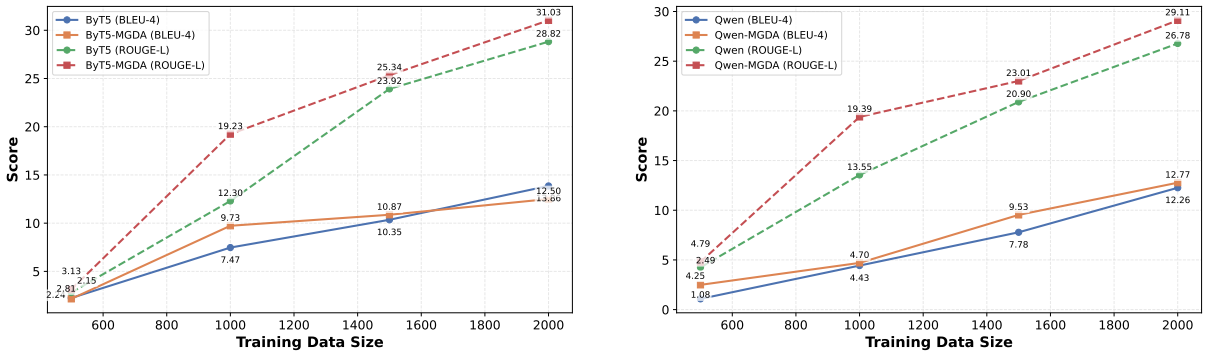


Figure 5: Ablation results for different augmented training-set sizes. Training Data Size counts examples after augmentation; both fine-tuned and MGDA settings use augmented data.

Context	Reference	Generated
...Batu aibide bi. 巴图在哪? Where is Batu?	Batu taqikv de gene-he 巴图去学校了。 Batu went to school.	Batu taqihv de gene-he-o
...kemuni unde. udu se ohobi. 尚未……。已经几岁了? Not yet... How old are you now?	ere aniya juwan ilan se. 今年十三岁。 Thirteen years old this year.	juwan ninggun se ohobi.
...tehe boude inu emu adali, fajiran de gemu muduri niruhabi, tura de gemu muduri folohobi. 像住在家里一样。墙壁上都会绘制了龙。在柱子上也雕刻了龙。 Just like at home. Dragons were painted all over the walls and carved on the pillars.	ere jergi muduri weihe be tuqibume. 把这些龙的牙齿（制作）出来， (Make) these dragons' teeth,	ere jergi muduri weihe be tuqibume.
...haqin haqin i turi hvda mujakv ja , utala aniya inu ere gese akv bihe. yargiyvn ? 各种豆子的价格都很低。这些年也不曾有过这样低的价格。真的吗? The prices of all kinds of beans are very low. Prices have not been this low in years. Really?	si jai aika geneqi. 你若再去， If you go again,	si jai aika geneqi.

Table 11: Four typical cases of text generation from Qwen-MGDA. Chinese and English translations are provided for the contexts and references.

the available artifacts do not include filtering logs or training logs that independently verify all implementation settings or actual optimizer update counts.

The nominal three-epoch budgets are 1929 and 5115 updates for 2572 and 6819 examples, respectively, assuming batch size four, no accumulation, no dropped final batch, and no early stopping. The fixed-step control targets 5115 updates. Actual update counts are unverified; even equal updates would not establish equal token counts, FLOPs, or wall-clock time. Run-level scores, seed variability, and statistical significance are unavailable.

E Case Study

Table 11 gives four representative outputs from Qwen-MGDA. The cases show both strengths and remaining errors. Case 1 contains a word-form error and an extra suffix, while Case 2 confuses a numeral. Cases 3 and 4 exactly match the references, indicating that the model handles frequent expressions and short answers better than spelling-sensitive or numerically precise generation.