BanglaSTEM: A Parallel Corpus and Term-Weighted Evaluation for Technical Bangla-English Translation

Kazi Reyazul Hasan, A. B. M. Alim Al Islam, Muhammad Abdullah Adnan
Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
{kazireyazulhasan, abdullah.adnan}@gmail.com, alim_razi@cse.buet.ac.bd

Abstract

Large language models excel at technical problem solving in English but struggle when questions are posed in Bangla. While translation offers a practical solution, existing Bangla-English systems frequently mistranslate specialized terminology, altering problem semantics and degrading downstream performance. We present BanglaSTEM, a dataset of 5,000 Bangla-English sentence pairs covering computer science, mathematics, physics, chemistry, and biology. Our pipeline extracts matching passages from official bilingual curriculum textbooks using OCR, then uses LLMs to align sentences and mark technical terms. These aligned examples serve as few-shot prompts for generating over 12,000 new translation pairs from LLMs, avoiding copyright issues. Human evaluators then select the best 5,000 pairs that correctly preserve technical terminology. We also test a term-weighted BLEU metric that gives higher weight to technical words, since standard metrics treat terminology errors and common word errors equally. We show that our weighted metric correlates better with downstream accuracy in code generation and math solving, while standard BLEU gives high scores even for wrong translations.

1 Introduction

Large language models have transformed how people solve technical problems. Students and professionals now use these models to generate code, solve mathematical equations, and understand complex scientific concepts. While large proprietary models like GPT-5, Claude, Grok and Gemini handle multiple languages including Bangla reasonably well, they come with significant limitations: ongoing API costs, privacy concerns when handling sensitive data, and limited options to customize or fine-tune for specific domains.

This has led to widespread adoption of smaller open-source models such as Llama, Mistral, Gemma, and their variants. These models offer crucial advantages: they can run locally on personal hardware, be fine-tuned for specific tasks, deployed in resource-constrained environments, and used without recurring costs or data privacy concerns. Educational institutions, startups, researchers, and individual developers increasingly rely on these models for building applications and solving domain-specific problems.

However, these smaller open-source models face a critical challenge with low-resource languages. Unlike massive proprietary models trained on trillions of parameters with extensive multilingual data, open-source models are typically trained with limited computational budgets and focused datasets. They excel at technical tasks in English but struggle significantly when the same questions are asked in Bangla. For Bangla speakers who want to utilize these accessible open-source models, this creates a major barrier. Using expensive proprietary models defeats the purpose of choosing open-source solutions. Training Bangla-specific models from scratch requires computational resources and data that most users cannot access.

One practical solution is to translate Bangla queries into English, process them with English-capable open models, and translate results back if needed. This approach preserves the benefits of open-source models while making them accessible to Bangla speakers. Unfortunately, existing Bangla-English translation systems are not designed for technical content. They work reasonably well for everyday conversation but mistranslate technical terms. A physics problem about momentum might become garbled. A coding question about recursion might lose its precise meaning. These translation errors change the problem itself, leading to incorrect solutions from downstream models.

We propose a focused solution to this translation gap. Instead of training entire language models for Bangla, we improve technical translation quality

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 403–412

specifically. This paper makes the following contributions:

- We create BanglaSTEM, a dataset of 5,000 high-quality Bangla-English parallel sentences covering five STEM domains: computer science, mathematics, physics, chemistry, and biology.
- We develop a pipeline that extracts aligned passages from official bilingual textbooks using OCR, marks technical terms explicitly, and uses these as few-shot prompts to generate over 12,000 translation candidates. Human evaluators then select the best 5,000 pairs that correctly preserve technical terminology.
- We test a term-weighted BLEU metric that assigns higher weight to technical vocabulary, addressing the limitation that standard metrics penalize terminology errors and filler word errors equally.
- We fine-tune a T5-based translation model on BanglaSTEM and demonstrate its effectiveness on code generation and mathematical problem solving, showing that our weighted metric correlates better with downstream accuracy than standard BLEU.

2 Related Work

Machine translation for Bangla has evolved significantly over the past decade, transforming from a low-resource language to a moderately resourced one with sophisticated neural models. However, this progress has been confined almost entirely to general-domain translation, leaving technical and scientific content largely unexplored.

2.1 Neural Sequence Models and Architecture Evolution

The evolution of neural machine translation began with recurrent architectures. Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) addressed the vanishing gradient problem in sequence modeling, enabling effective learning over long sequences. LSTM-based encoder-decoder models became the standard for machine translation until the introduction of the Transformer architecture (Vaswani et al., 2017), which replaced recurrence with self-attention mechanisms, enabling parallelization and improved handling of long-range dependencies.

The Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020) further advanced the field by introducing a unified framework that converts all NLP tasks into a text-to-text format. T5 demonstrated that pre-training on large corpora followed by task-specific fine-tuning achieves state-of-the-art results across diverse tasks including translation.

Recent open-source large language models have democratized access to powerful NLP capabilities. Llama 2 (Touvron et al., 2023) introduced 7B to 70B parameter models optimized for dialogue and general tasks. Mistral 7B (Jiang et al., 2023) achieved superior performance through grouped-query attention and sliding window attention, outperforming larger models on reasoning and code generation benchmarks. Gemma (Team et al., 2024) from Google DeepMind released lightweight 2B and 7B parameter models built on Gemini research, demonstrating strong performance on language understanding tasks.

2.2 General-Purpose Bangla-English Translation

Transformer-based architectures form the foundation of modern Bangla-English translation. (Hasan et al., 2020) demonstrated that transformers substantially outperformed LSTM-based models, achieving BLEU scores of 21.42 for Bangla to English and 25.44 for English to Bangla.

IndicTrans2, developed by (Gala et al., 2023), represents the most advanced open-source system with 1.1 billion parameters trained on the Bharat Parallel Corpus Collection (AI4Bharat, 2023) containing 230 million Bengali-English sentence pairs, achieving chrF++ scores in the high 50s to low 60s on FLORES-200 (Goyal et al., 2022). Meta's NLLB-200 (Team et al., 2022) provides another baseline with 54.5 billion parameters covering 200 languages. However, language-specific models consistently outperform these massively multilingual systems.

The development of parallel corpora has been

crucial for Bangla NMT. Samanantar (Ramesh et al., 2022) compiled 49.7 million sentence pairs between English and 11 Indic languages, including 8.6 million Bengali-English pairs, through web mining, document OCR, and multilingual alignment. (Hasan et al., 2020) earlier contributed 2.75 million high-quality sentence pairs through custom sentence segmentation, aligner ensembling, and batch filtering. SUPara (Al Mumin et al., 2012) provides 800,000 manually curated pairs. All these corpora focus exclusively on general-domain content from news, web crawls, government documents, and everyday conversation.

2.3 Bangla Foundation Models and Evaluation

BanglaBERT (Bhattacharjee et al., 2021) represents the first comprehensive BERT-based model specifically for Bangla with 110 million parameters pretrained on 2.18 billion tokens, outperforming mBERT by 6.8 BLUB score and XLM-R by 4.3. BanglaT5 (Bhattacharjee et al., 2023) introduced the first T5-based sequence-to-sequence transformer for Bangla, achieving 38.8 SacreBLEU (Post, 2018) on Bangla-English translation. Both models consistently outperform multilingual alternatives, establishing language-specific pretraining as superior for Bangla.

FLORES-200 (Goyal et al., 2022) serves as the gold-standard benchmark with 3,001 professionally translated sentences across 200 languages. The IN22 benchmark provides n-way parallel content covering all 22 Indian languages with India-centric content. (Ahmed et al., 2024) demonstrated that 6,193 professionally translated sentence pairs with rigorous quality control outperformed larger automatically-mined datasets. All existing benchmarks focus on general-domain content from news, Wikipedia, and everyday conversation. No evaluation benchmark exists for technical or scientific content.

Domain adaptation techniques exist in general neural machine translation literature (Chu and Wang, 2018; Hu et al., 2019) but have not been applied to Bangla technical translation. These approaches typically involve fine-tuning on domain-specific data, curriculum learning, or multi-domain training strategies. Limited work exists on biomedical terminology for Bangla, but this focuses on named entity recognition rather than translation. (Sazzed, 2022) introduced BanglaBioMed with 12,000 annotated tokens for biomedical NER.

2.4 Translation Evaluation Metrics

Standard evaluation metrics for machine translation include BLEU (Papineni et al., 2002), which measures n-gram precision between candidate and reference translations. However, BLEU has known limitations, particularly for morphologically rich languages and for distinguishing semantically critical errors from minor variations. SacreBLEU (Post, 2018) addressed reproducibility concerns by standardizing tokenization and providing consistent evaluation protocols.

Character-level metrics like chrF (Popović, 2015) and chrF++ (Popović, 2017) offer advantages for morphologically complex languages by computing F-scores over character n-grams rather than word n-grams. These metrics correlate better with human judgments for languages with rich morphology. However, none of these metrics distinguish between errors in critical technical terminology versus common vocabulary, treating all word-level errors equally regardless of their semantic importance.

2.5 The Technical Translation Gap

Despite comprehensive searching across major NLP venues, we did not find substantial research from 2019 to 2025 specifically addressing technical, scientific, or STEM domain Bangla-English machine translation. Our work directly addresses this absence of technical domain translation resources. The closest prior work, the WMT24 Bangla Seed Dataset (Ahmed et al., 2024), shares our emphasis on human-curated quality but focuses on general-domain content.

3 Methodology

Our methodology comprises five key stages (see Figure 1): few-shot prompt construction from bilingual textbooks, multi-model translation generation, rigorous human curation,

Figure 1: Overview of the BanglaSTEM pipeline. The dataset construction phase extracts aligned passages from bilingual NCTB textbooks using OCR, marks technical terms, and uses these as few-shot prompts to generate 12,711 translation candidates via three LLMs.

quality-based selection, and model fine-tuning with term-weighted evaluation. We established three core principles. First, **technical domain coverage** must span STEM disciplines where open-source LLMs demonstrate strong English performance but struggle with Bangla. Second, **terminology preservation** requires maintaining technical precision, as mistranslated terms fundamentally alter problem semantics. Third, **authentic Bangla usage** demands both transliterated English terms (ডিএনএ (DNA), এপিআই (API)) and native equivalents (অণু (molecule), সংকলন (compilation)) reflecting real-world technical discourse where both coexist naturally.

3.1 Domain Selection and Coverage

We systematically selected domains to maximize coverage of technical problem-solving scenarios. Our domain taxonomy covers computer science (7 programming languages, software engineering, systems architecture, data engineering), bioinformatics (genomics, proteomics, systems biology), computational chemistry (molecular simulation, drug discovery, quantum chemistry), medical applications, mathematics (all major subfields), and physics.

3.2 Few-Shot Prompt Construction from Bilingual Textbooks

To generate high-quality few-shot prompts for translation, we extracted aligned Bangla-English sentence pairs from official curriculum textbooks published by the National Curriculum and Textbook Board (NCTB) of Bangladesh. These textbooks exist in both Bangla and English versions, providing naturally aligned technical content that reflects authentic educational discourse.

3.2.1 Text Extraction via OCR

We used dots.ocr (Li et al., 2025), a unified Vision-Language Model that jointly performs layout detection, text recognition, and relational understanding within an end-to-end framework. Unlike traditional multi-stage OCR pipelines that suffer from error propagation, dots.ocr leverages joint training across tasks, achieving state-of-the-art performance on multilingual document parsing benchmarks including OmniDocBench.

We extracted text from scanned PDF versions of NCTB textbooks covering mathematics, physics, chemistry, and information and communication technology (ICT) for grades 9-12. We processed 2,533 pages across 8 textbook pairs (Bangla and English versions), extracting approximately 16,500 text segments.

3.2.2 Sentence Alignment and Term Marking

After extraction, we employed GPT-4o to perform sentence-level alignment between Bangla and English versions of the same passages. The alignment prompt instructed the model to:

1. Match semantically equivalent sentences across language versions

2. Identify and mark technical terms with indexed tags
3. Preserve the natural sentence boundaries from each language

Technical terms were marked using indexed tags in both languages to ensure terminology correspondence. For example:

> **English:** The process of <1>integration</1> is the reverse of <2>differentiation</2>.
> **Bangla:** <1>সমাকলন</1> প্রক্রিয়াটি <2>অন্তরকলন</2> এর বিপরীত।

This explicit marking serves two purposes: (1) it ensures the LLM learns correct technical term correspondences during few-shot prompting, and (2) it enables automatic identification of technical terms for our weighted evaluation metric.

3.2.3 Copyright Considerations

Since NCTB textbook content is copyrighted, we used these extracted pairs exclusively as few-shot prompt examples to guide LLM generation rather than including them directly in our dataset. This approach transfers the translation style and terminology patterns without reproducing copyrighted material in the final corpus.

3.3 Multi-Model Translation Candidate Generation

We employed three frontier language models: GPT-4o, Claude Sonnet 4, and Gemini 2.5 Pro. Each model received domain-specific few-shot prompts containing 8-12 examples from our textbook-extracted pairs, demonstrating correct terminology usage with explicit term marking.

This process generated 12,711 translation candidates balancing creativity and consistency across models.

3.4 Human Curation Process

We recruited 7 native Bengali speakers with STEM backgrounds, each assigned to a specific domain matching their expertise to ensure technically informed evaluation. All 5,000 candidate translation pairs underwent human verification; each pair was independently evaluated by two annotators from the same domain pool. Annotators were not shown the model identities during evaluation to avoid bias. Each translation received evaluation on three 5-point Likert scales:

Translation Accuracy (TA): Semantic fidelity to source (5=perfect equivalence, 1=severe mistranslation). **Technical Terminology (TT):** Correct domain-specific term usage (5=all terms accurate, 1=critical errors). **Linguistic Naturalness (LN):** Grammatical correctness and fluency (5=completely natural Bangla, 1=broken grammar).

Annotators followed explicit guidelines requiring them to assess whether technical terms were rendered using standard Bengali STEM conventions and whether the translated sentence preserved the precise meaning of the source without paraphrase or omission. Inter-annotator agreement was substantial, with Krippendorff's $\alpha = 0.78$ for TA, $\alpha = 0.81$ for TT, and $\alpha = 0.67$ for LN. The relatively lower agreement on LN reflects natural variation in stylistic preferences rather than disagreement on correctness. Disagreements exceeding 1.5 points were resolved through discussion between annotators to reach consensus.

3.5 Quality-Based Selection

We computed composite quality scores using weighted combination prioritizing technical correctness:

$$Q = 0.4 \cdot \text{TA} + 0.4 \cdot \text{TT} + 0.2 \cdot \text{LN} \quad (1)$$

For each English sentence, we first selected the highest-scoring translation:

$$t^* = _{t \in \{t_1, t_2, t_3\}} Q(t) \quad (2)$$

This yielded 4,237 unique pairs. To capture valid translation diversity where technical terms admit multiple correct renderings, we included 763 additional candidates with $Q \geq 4.0$ and $\Delta Q < 0.5$ from the top translation, reaching 5,000 pairs.

Table 1 shows around 83.2% achieve scores above 4.0, with mean TA of 4.41 and mean TT of 4.52.

Table 1: Quality score distribution in BanglaSTEM.

Score Range	Count	%	Mean TA	Mean TT
4.5 – 5.0	1,847	36.9	4.82	4.91
4.0 – 4.5	2,316	46.3	4.37	4.48
3.5 – 4.0	837	16.7	3.91	4.02
Overall	**5,000**	**100.0**	**4.41**	**4.52**

3.6 Dataset Composition

Table 2 presents domain distribution. Totals exceed 5,000 as sentences may address multiple domains.

Table 2: Domain distribution in BanglaSTEM. Sentences may belong to multiple categories.

Domain	Count	% of Dataset
Programming	2,601	52.0
Information Technology	1,184	23.7
Mathematics	1,274	25.5
Physics	491	9.8
Chemistry	367	7.3
Biology & Bioinformatics	278	5.6

Table 3 presents linguistic characteristics, showing Bangla sentences average 12.4 words versus 14.5 for English reflecting morphological differences.

Technical Bangla strategically mixes transliterated English terms and native vocabulary. Table 4 shows programming uses 78% transliteration (ফাংশন [function], অ্যারে [array]) while mathematics uses only 42% with well-established Bengali terms (সমীকরণ [equation], অন্তরকলন [differentiation]).

3.7 Term-Weighted BLEU Metric

Standard BLEU (Papineni et al., 2002) treats all tokens equally, which fails to capture the outsized importance of technical terminology in STEM translation. A translation that renders "integration" as "addition" receives similar penalty to one that uses "the" instead of "a"—yet the former fundamentally changes problem semantics while the latter is stylistic.

3.7.1 Standard BLEU Formulation

BLEU computes modified n-gram precision by comparing candidate translation c against reference r:

$$p_n = \frac{\sum_{\text{n-gram} \in c} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{\text{n-gram} \in c} \text{Count}(\text{n-gram})} \quad (3)$$

Table 3: Linguistic statistics for BanglaSTEM parallel sentences.

Metric	Bangla	English
Mean sentence length (words)	12.4	14.5
Median sentence length (words)	13	14
Std deviation (words)	5.4	6.6
Mean sentence length (characters)	87.1	102.7
Total vocabulary size	8,799	9,407
Total Words	62,001	72,499

Table 4: Transliterated versus native terminology usage across domains.

Domain	Transliterated	Native Bangla
Programming	78%	22%
Information Technology	65%	35%
Mathematics	42%	58%
Physics	55%	45%
Chemistry	61%	39%
Biology	58%	42%
Overall	**63%**	**37%**

where $\text{Count}_{\text{clip}}$ clips the count of each n-gram to its maximum occurrence in the reference. The final BLEU score combines precisions across n-gram orders with a brevity penalty:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (4)$$

where $w_n = 1/N$ (typically $N = 4$) and $\text{BP} = \min(1, e^{1-|r|/|c|})$.

3.7.2 Term-Weighted Modification

We introduce Term-Weighted BLEU (TW-BLEU) by assigning weight $w_t > 1$ to technical terms and weight 1 to common tokens. Let T denote the set of technical terms identified via our term-marking process. For unigram precision, instead of counting each token once, we count technical terms with weight w_t:

$$\text{Count}_w(\text{token}) = \begin{cases} w_t & \text{if token} \in T \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

The weighted unigram precision becomes:

$$p_1^w = \frac{\sum_{u \in c} \text{Count}_w(u) \cdot \mathbb{1}[u \in r]}{\sum_{u \in c} \text{Count}_w(u)} \quad (6)$$

where $\mathbb{1}[\cdot]$ is the indicator function. This formulation ensures that missing a technical term incurs w_t times the penalty of missing a common word.

3.7.3 Illustrative Example

Consider translating: "*Compute the derivative of the polynomial function.*"

With $w_t = 3$, the denominator for unigram precision is:

$$1 + 1 + 3 + 1 + 1 + 3 + 1 = 11$$

If a candidate correctly translates all common words but mistranslates "derivative" as "calculation":

- **Standard BLEU:** Loses 1/7 $\approx$ 14.3% precision
- **TW-BLEU ($w_t = 3$):** Loses 3/11 $\approx$ 27.3% precision

This amplified penalty better reflects the semantic impact of terminology errors on downstream task performance.

3.7.4 Extension to Higher-Order N-grams

For n-grams containing at least one technical term, we apply the maximum weight among constituent tokens:

$$\text{Count}_w(\text{n-gram}) = \max_{t \in \text{n-gram}} \text{Count}_w(t) \quad (7)$$

This ensures that technical bigrams like "binary search" or "quantum entanglement" receive appropriate weighting. The final TW-BLEU score uses these weighted precisions:

$$\text{TW-BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n^w\right) \quad (8)$$

We set $w_t = 3$ based on preliminary experiments showing this value maximizes correlation with downstream task accuracy while maintaining interpretability.

3.8 Fine-tuning Model

We fine-tuned BanglaT5, a state-of-the-art sequence-to-sequence model pre-trained on Bangla-English translation tasks. Starting from the `banglat5` (Bhattacharjee et al., 2023) checkpoint (247M parameters), we applied supervised fine-tuning on our curated BanglaSTEM dataset. No explicit terminology annotations were provided during fine-tuning; the model learns domain-specific STEM terminology implicitly through exposure to the curated parallel sentence pairs. Our training configuration balanced convergence speed with stability: learning rate of 5×10^{-4} with 25 warmup steps, batch size of 16 with gradient accumulation over 4 steps (effective batch size 64), and 8 training epochs. Maximum sequence length was set to 512 tokens.

4 Experiments

We evaluated BanglaSTEM's impact on downstream STEM reasoning tasks through two benchmarks: Bangla code generation and mathematical problem-solving. Beyond measuring task accuracy, we demonstrate that our term-weighted BLEU metric correlates more strongly with downstream performance than standard BLEU, validating its utility for technical translation evaluation.

4.1 Experimental Setup

All experiments were conducted on NVIDIA A100 GPUs. We compared four translation approaches:

1. **Direct Bangla**
2. **Google Translation API**
3. **BanglaT5-Base** (Bhattacharjee et al., 2023)
4. **BanglaSTEM** (Our fine-tuned model)

Each translation approach was evaluated with multiple state-of-the-art LLMs to ensure robustness across model architectures and sizes.

4.2 Task 1: Bangla Code Generation

We evaluated on 400 programming problems from mHumanEval (Raihan et al., 2025) written in Bangla, covering algorithms, data structures, and software engineering concepts. Problems required generating Python functions with correct syntax and logic.

Table 5 shows BanglaSTEM consistently outperforms all baselines across model sizes. On average, BanglaSTEM achieves 81.7% accuracy, representing a 21.8 percentage point improvement over BanglaT5-Base and 6.2 points

Table 5: Code generation accuracy (%) on 400 Bangla programming problems across multiple LLMs. Best results per model in **bold**.

LLM	Direct	Google	BanglaT5	Ours
Gemma-27B	35.3	76.5	59.8	**82.5**
Llama-3.1-70B	37.5	78.0	62.3	**84.0**
Llama-3.1-8B	28.0	68.3	51.5	**74.8**
Mistral-7B	24.5	64.0	47.8	**71.3**
Average	35.5	75.5	59.9	**81.7**

over Google Translate. Notably, even smaller models like Mistral-7B with BanglaSTEM (71.3%) outperform larger models like Llama-3.1-70B with BanglaT5-Base (62.3%), demonstrating that translation quality can compensate for model capacity.

4.3 Task 2: Bangla Mathematical Problem-Solving

We evaluated on 100 problems from the Bangla Math Olympiad training set (Sushmit et al., 2024), spanning arithmetic, algebra, number theory, and combinatorics. These problems require multi-step reasoning and precise understanding of mathematical concepts.

Table 6: Mathematical problem-solving accuracy (%) on 100 Bangla Math Olympiad problems. Best results per model in **bold**.

LLM	Direct	Google	BanglaT5	Ours
Gemma-27B	33.0	74.0	61.0	**81.0**
Llama-3.1-70B	34.0	75.0	62.0	**82.0**
Llama-3.1-8B	31.0	72.0	59.0	**79.0**
Mistral-7B	27.0	67.0	54.0	**74.0**
Average	33.6	74.0	61.0	**81.0**

As shown in Table 6, BanglaSTEM achieves an average of 81.0% accuracy across all models, with consistent improvements of 7-10 points over Google Translate and 18-20 points over BanglaT5-Base.

4.4 TW-BLEU Correlation with Downstream Accuracy

We analyzed 500 translation-task pairs, computing both BLEU and TW-BLEU scores against reference translations and recording downstream task success.

Correct solutions correlate with higher TW-BLEU: For problems solved correctly, TW-BLEU scores average 0.72 compared to 0.68 for standard BLEU. The 4-point gap indicates TW-BLEU better captures the translation quality required for successful reasoning.

Incorrect solutions show larger TW-BLEU penalties: For failed problems, TW-BLEU averages 0.41 while standard BLEU averages 0.52. This 11-point difference demonstrates that TW-BLEU more severely penalizes the terminology errors prevalent in downstream failures.

A natural question is whether this gap reflects TW-BLEU's sensitivity to terminology specifically, or simply that failed problems involve harder sentences with lower translation quality overall. We observe that standard BLEU scores are also lower for failed problems (0.52 vs. 0.68), confirming that sentence difficulty is indeed a contributing factor. However, the *differential* between TW-BLEU and BLEU penalties is markedly larger for incorrect solutions (11 points) than for correct ones (4 points). If general sentence difficulty were the sole driver, we would expect both metrics to degrade proportionally. The disproportionate TW-BLEU drop instead suggests that failed problems disproportionately involve mistranslated technical terms—errors that standard BLEU, by treating all tokens equally, systematically underweights. This is consistent with manual inspection of failure cases, where a correctly structured but terminologically incorrect translation (e.g., a mistranslated function name or operator) produced fluent but semantically wrong reasoning that BLEU scored leniently.

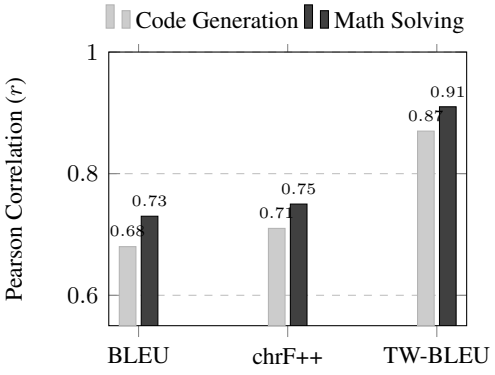

Figure 2: Pearson correlation between translation metrics and downstream task accuracy. TW-BLEU substantially outperforms BLEU and chrF++ on both tasks.

Figure 2 quantifies these observations through Pearson correlation coefficients. TW-BLEU achieves $r = 0.87$ for code generation and $r = 0.91$ for mathematical problem-solving, substantially outperforming both BLEU ($r = 0.68$, $r = 0.73$) and chrF++ ($r = 0.71$, $r = 0.75$). This validates our hypothesis that weighting technical terminology provides a more reliable signal for translation quality in STEM domains, beyond what general translation difficulty alone can explain.

5 Conclusion and Future Work

We presented BanglaSTEM, the first parallel corpus for technical domain Bangla-English translation, accompanied by a simple term-weighted evaluation metric that better predicts downstream task performance. Our fine-tuned model achieves 81.7% accuracy on code generation and 81.0% on mathematical problem-solving, representing improvements of 6-7 percentage points over open-source LLMs and 20+ points over general-purpose NMT models. More broadly, we contribute a reproducible pipeline for creating technical translation resources from bilingual educational materials. This methodology of OCR extraction, LLM-assisted alignment with term marking, multi-model candidate generation, and human curation can be adapted to any low-resource language with access to bilingual STEM curricula.

Several directions remain for future work. First, scaling to 20,000-50,000 pairs through community annotation efforts would improve coverage of long-tail terminology. Second, extending the pipeline to additional languages with similar educational infrastructure would validate cross-lingual generalizability. Third, exploring domain adaptation techniques that utilize our curated data as seed examples for continued pre-training could further improve translation quality.

Ethics Statement

All annotators were compensated above local minimum wage standards and provided informed consent. The dataset contains no personal or sensitive content. Example sentences were generated by LLMs or adapted from publicly available educational materials. We acknowledge that technical content may reflect biases present in the training data of LLMs used for generation.

Limitations

Our dataset of 5,000 pairs, while carefully curated, represents a foundation rather than comprehensive coverage. Domain distribution is uneven, with programming heavily represented (52%) compared to biology (5.6%). We recommend using BanglaSTEM for fine-tuning pre-trained models rather than training from scratch.

Evaluation focused on two downstream tasks with seven LLMs; generalization to other tasks (e.g., technical summarization, documentation generation) and domains (e.g., engineering, statistics) remains to be verified. The multi-model generation approach using proprietary LLMs limits full reproducibility of the generation phase, though we release all curated pairs and fine-tuned models to enable downstream research. The term-weighted BLEU metric requires manual verification of technical terms which are detected by LLMs, that may not scale to new domains without additional annotation effort.

References

Firoz Ahmed, Nitin Venkateswaran, and Sarah Moeller. 2024. The bangla/bengali seed dataset submission to the wmt24 open language data initiative shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 556–566.

AI4Bharat. 2023. Bharat parallel corpus collection. https://github.com/AI4Bharat/IndicTrans2.

Md Abdullah Al Mumin, Abu Awal Md Shoeb, Md Reza Selim, and Muhammed Zafar Iqbal. 2012. Supara: A balanced english-bengali parallel corpus. *SUST Journal of Science and Technology*, 16(2):46–51.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, and Rifat Shahriyar. 2023. Banglanlg and banglat5: Benchmarks and resources for evaluating low-resource natural language generation in bangla. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 726–735.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining

and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. corr abs/1806.00258 (2018). *arXiv preprint arXiv:1806.00258*.

Jay Gala, Pranjal A Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, and 1 others. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *arXiv preprint arXiv:2305.16307*.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation. *arXiv preprint arXiv:2009.09359*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. *arXiv preprint arXiv:1906.00376*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. 2025. dots.ocr: Multilingual document layout parsing in a single vision-language model. *Preprint*, arXiv:2512.02498.

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.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the third conference on machine translation: Research papers*, pages 186–191.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nishat Raihan, Antonios Anastasopoulos, and Marcos Zampieri. 2025. mhumaneval-a multilingual benchmark to evaluate large language models for code generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11432–11461.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan Ak, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, and 1 others. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Salim Sazzed. 2022. Banglabiomed: A biomedical named-entity annotated corpus for bangla (bengali). In *Proceedings of the 21st workshop on biomedical language processing*, pages 323–329.

Sushmit, Asib Rahman, Asif Azad, Ashrafur Rahman, Mohammad Sadat Hossain, Nafis Tahmid, Shattik Islam, Fahad Ahmed Akash, Mohaiminul Islam, Abir Muhtasim, Tahsin, and Tawsif Tashwar Dipto. 2024. Dl sprint 3.0 | bengali ai math olympiad. https://kaggle.com/competitions/dlsprint3. Kaggle.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

NLLB Team, Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.