ISSN 2007-9737

AstraMT: Instruction-Tuned Few-Shot Assamese-English Translation with Context-Aware Prompting and Reranking

Basab Nath[1,*], Sunita Sarkar[2], Somnath Mukhopadhyay[2]

[1] Bennett University,
School of Computer Science and Engineering,
India

[2] Assam University,
Department of Computer Science and Engineering,
India

{basabnath, sarkarsunita2601}@gmail.com, som.cse@live.com

Abstract. Developing machine translation (MT) systems for low-resource languages such as Assamese remains challenging due to limited parallel corpora and morphological complexity. Recent instruction-tuned large language models (LLMs) offer few-shot translation capabilities, but static prompt-based methods often yield suboptimal performance in real world scenarios. This paper introduces AstraMT, a modular pipeline for Assamese–English few-shot translation using LLMs. AstraMT incorporates a context-aware prompt selector (CAPS), syntactic prompt templates, multi-output reranking based on BLEU and COMET scores, and a lightweight post-editing module that corrects named entity errors and auxiliary omissions. The framework was evaluated on two datasets: the FLORES-200 devtest set and a manually aligned subset of the Samanantar corpus. AstraMT achieved BLEU improvements of up to +3.2 and COMET gains of +0.07 over static few-shot prompting. The AstraMT-Mixtral variant reached a BLEU of 23.0 on FLORES-200 and 21.3 on Samanantar, outperforming the supervised IndicTrans2 baseline. Qualitative and error analyses further highlighted AstraMT's ability to generate fluent and semantically accurate translations. These results demonstrate that AstraMT provides an effective and extensible framework for LLM based translation in low-resource settings and can generalize across different LLMs without additional fine-tuning.

Keywords. Context ware prompt selector, prompt constructor, LLM, mixtral.

1 Introduction

Machine Translation (MT) has seen significant advances in recent years, primarily driven by deep neural architectures and the availability of large-scale parallel corpora [4, 28]. However, such progress has remained largely inaccessible to low-resource languages like Assamese, which lack the volume and quality of bilingual datasets required to train traditional Neural Machine Translation (NMT) systems [11]. This presents a substantial barrier to equitable language technology development across the Indian subcontinent.

The recent emergence of instruction-tuned Large Language Models (LLMs) such as GPT-4 [22], Mixtral [15], and LLaMA2-Chat [26] has opened up new possibilities for MT in low-resource scenarios. These models, trained on broad multilingual corpora, are capable of zero-shot and few-shot translation by leveraging in-context learning. However, prior work using such LLMs has often relied on static prompt templates with a small, manually curated set of examples. While effective in high-resource settings, this approach fails to account for domain shifts, semantic variability, and stylistic nuances inherent in Assamese, limiting both translation quality and robustness.

This paper introduces **AstraMT**—a modular framework for Assamese–English few-shot translation using instruction-tuned LLMs. Unlike existing prompting pipelines, AstraMT features a dynamic, multi stage design that adapts the translation context to each input.

The framework comprises four key components: (1) a CAPS that retrieves semantically similar examples using multilingual sentence embeddings, (2) a Prompt

Constructor that dynamically formats and assembles instruction aligned prompts, (3) a Multi Prompt Reranking stage that evaluates multiple candidate translations based on BLEU (Bilingual Evaluation Understudy) and COMET (Crosslingual Optimized Metric for Evaluation of Translation) scores, and (4) a lightweight Post Editing Module that corrects frequent LLM translation errors, including named entity mismatches, dropped auxiliaries, and punctuation inconsistencies.

The main contributions of this study are as follows:

— AstraMT is proposed as a modular framework for few-shot translation between Assamese and English using instruction-tuned LLMs.

— CAPS is introduced as a context-aware retrieval mechanism that selects semantically relevant examples for dynamic prompt construction.

— Multi-prompt reranking and a lightweight post-editing module are integrated to improve translation quality without fine-tuning.

— Extensive evaluation is conducted across multiple LLMs, benchmarks, and metrics, and a new qualitative taxonomy of translation errors in low-resource Indic MT is presented.

The rest of the paper is organized as follows: Section 2 reviews related literature. Section 3 details the AstraMT architecture. Section 4 describes the experimental setup. Section 5 presents results and analysis. Section 6 concludes the paper and outlines directions for future work.

2 Literature Review

Recent advances in instruction-tuned LLMs have revitalized interest in zero-shot and few-shot approaches to machine translation (MT), especially for languages where parallel corpora are scarce. While conventional supervised NMT methods still dominate high-resource settings, low-resource scenarios demand alternative strategies such as in-context learning, retrieval-augmented prompting, and output reranking.

In this section, we survey key developments in these areas, with a particular focus on approaches that inform the design of our proposed AstraMT framework.

2.1 Few-shot and Zero-shot Machine Translation

Traditional neural machine translation (NMT) systems rely on large-scale parallel corpora for supervised training [4, 28], which are often unavailable for low-resource languages such as Assamese. This limitation has led to the exploration of zero-shot and few-shot approaches, particularly with the advent of instruction-tuned large language models (LLMs) such as GPT-3 [5], GPT-4 [22], and Mixtral [15].

These models enable translation through in context learning, where a small set of examples is included in the prompt to guide the model. However, as shown in [12, 16], the quality of few-shot translation heavily depends on the choice and ordering of examples, which are often manually fixed and domain-insensitive.

2.2 Context-aware Prompting and Retrieval Methods

Recent work has highlighted the importance of context-aware prompting in improving in-context learning performance [16, 25, 2]. Retrieval augmented methods select semantically similar examples from a database of candidate prompts based on input similarity, often using multilingual sentence embeddings such as LASER (Language-Agnostic Sentence Representations) [3] or LaBSE (Language-agnostic BERT Sentence Embeddings) [7]. In the context of MT, this has been applied to improve few-shot translation through adaptive example selection [18, 29], but primarily in high-resource settings and for European or Chinese language pairs. To the best of our knowledge, no prior work has explored this for Assamese–English translation.

2.3 Reranking and Post Editing in LLM based Translation

In traditional MT, reranking of n-best hypotheses has been widely used to improve fluency and adequacy, particularly in statistical MT systems [21]. In LLM based settings, reranking has reemerged as a way to select the best output among multiple generations [17], using reference free metrics like COMET [9] or sentence level BLEU. Similarly, lightweight post-editing methods have been proposed to address common errors in LLM translations such as hallucination, named entity distortion, or inconsistent verb inflections [6]. This work builds on these ideas by incorporating both reranking and post editing directly into the inference pipeline for low-resource translation.

2.4 Low-resource Indic Language Translation

Despite increased interest in multilingual NLP, most prior work has focused on major Indic languages like Hindi, Bengali, and Tamil [11, 24]. Assamese remains underrepresented in public MT benchmarks, with limited availability of high-quality corpora and pretrained models. FLORES-200 [20] and the Samanantar corpus [24] have improved access to parallel data, while Laskar et al. [13] introduced a domain-specific English–Assamese corpus spanning agriculture, news, and COVID-19, which proved effective for domain-adapted NMT. Back-translation has recently shown strong results: Ahmed etal. (ICON 2023) report improvements of over +6 BLEU for English→Assamese via iterative methods [1]. Concurrently, Nath et al. (2024) demonstrate that integrating transliteration modules into NMT pipelines yields significant gains in translation accuracy, particularly for named entities [19]. Nevertheless, these methods involve additional training and pipelined components. AstraMT offers a fully inference driven alternative, relying entirely on instruction tuned LLMs enhanced by dynamic example retrieval, reranking, and post editing.

While previous works have explored context-aware prompting or reranking individually, few have unified these into an end to end pipeline for low-resource MT. To our knowledge, AstraMT is the first framework that combines adaptive prompt selection, multi output reranking, and post-editing specifically for Assamese–English translation using instruction-tuned LLMs.

3 Methodology

This section describes the design of AstraMT, the proposed modular framework for few-shot Assamese to English translation using instruction-tuned LLMs. AstraMT eliminates the need for model fine-tuning by leveraging in-context learning with dynamically selected and reranked examples. The system is designed to address challenges such as vocabulary sparsity, semantic drift, and inconsistent translation style common in low-resource setups.

AstraMT is composed of four main components that operate in sequence: (i) a CAPS, (ii) a Prompt Constructor, (iii) Multi-Prompt LLM Inference with scoring based reranking, and (iv) a lightweight Postprocessing module. These components together transform a raw Assamese sentence into a high-quality English translation through a fully inference-time pipeline.

Figure 1 shows how AstraMT processes this sentence using each module.

3.1 Framework Overview

AstraMT is a modular, fully inference-time pipeline designed to translate Assamese sentences into English using few-shot prompting with instruction-tuned LLMs. The system comprises five sequential components: CAPS, a Prompt Constructor, a multi-prompt LLM Inference engine, a Scoring and Reranking module, and a final Postprocessing stage. Each component addresses key challenges in low-resource translation ranging from vocabulary sparsity to inconsistent fluency without requiring fine tuning.

The pipeline begins with the **CAPS**, which retrieves a small number ($k = 3$ or 4) of semantically similar Assamese–English translation pairs for few-shot prompting. CAPS uses LaBSE [7] embeddings to encode both the input sentence and a pool of held-out exemplars (from FLORES-200 and Samanantar), and ranks them using cosine similarity. The most relevant examples are routed to the next module. Figure 2 shows this retrieval process.

Next, the **Prompt Constructor** assembles the selected examples into a structured in-context prompt using a fixed instruction template. Each example pair is formatted with consistent tags, and the final input sentence is appended at the end with its English counterpart left blank. This structured prompt guides the LLM to perform the translation in-context. Figure 3 illustrates this formatting process.

The constructed prompt is then passed to an instruction-tuned **LLM Inference** engine. Mixtral-8×7B-Instruct [10] was primarily used as the backbone, activating 2 of 8 expert subnetworks per token for efficient inference. Additional evaluations were conducted using GPT-4 [22] and Zephyr-7B-β [27]. Each prompt generated multiple outputs (`n=3`) using top-p sampling (`temperature=0.9`, `top-p=0.95`), increasing lexical diversity. These outputs were routed to the scoring module.

The **Scoring and Reranking** module evaluates all candidate translations using both BLEU [23] and COMET [9] scores. BLEU captures n-gram overlap with a reference, while COMET estimates adequacy and fluency using reference free contextual embeddings. The top scoring translation is selected for final refinement. Table 2 illustrates how reranking improves selection quality.

Finally, the selected output is passed to the **Postprocessing Module**, which performs surface level corrections to enhance fluency and accuracy. This includes (i) detokenization and spacing fixes, (ii) transliteration of named entities (e.g., "Silchar" → শিলছাৰ), and (iii) light grammatical edits. Table 3 showcases examples

Table 1. Summary of prior work on Assamese–English machine translation

Study / Method	Language Pair	Key Technique	Limitation / Gap
Ahmed et al. [1]	English ↔ Assamese	Iterative back-translation using synthetic data	Requires model retraining; sensitive to noisy pseudo-labels
Laskar et al. [13]	English ↔ Assamese	Domain-specific corpus with supervised NMT	Limited domain coverage; does not generalize across domains
Nath et al. [19]	English ↔ Assamese	Transliteration-enhanced NMT pipeline	Needs external transliteration module; no in-context learning
AstraMT (Ours)	English ↔ Assamese	Few-shot prompting with CAPS, multi-prompt reranking, post-editing	Fully inference-time system; avoids any training; adaptable and modular

Note: This table summarizes selected representative works with methodological relevance to our approach. While other studies exist in Assamese–English MT, we focus here on those that introduce novel techniques beyond standard transformer baselines.

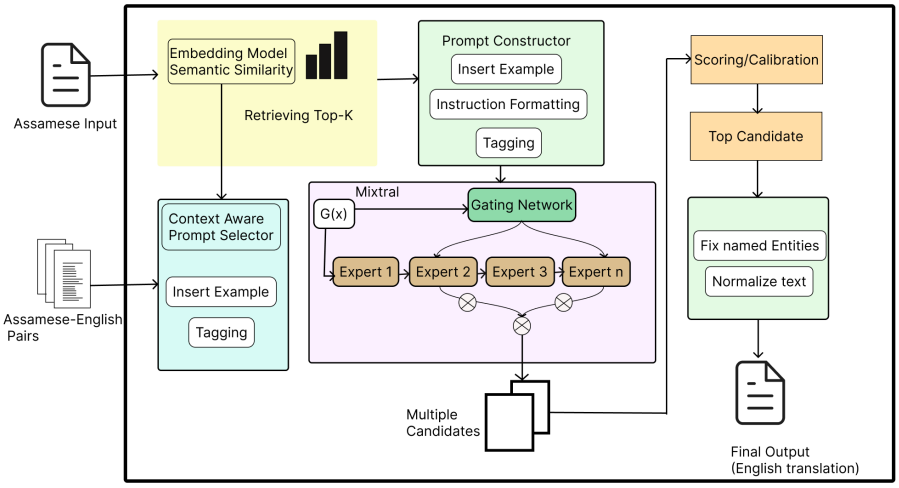

Fig. 1. Overview of the proposed AstraMT framework. The pipeline includes dynamic prompt selection, instruction-tuned LLM inference, reranking, and postprocessing to enhance Assamese–English translation

Table 2. Candidate translations for the sentence "আমি আজি চিলছার গৈছিলু" (We went to Silchar today)

Candidate Translation	COMET Score
We had gone to Silchar today.	0.71
Today we went to Silchar.	0.74
We went to Silchar today.	**0.78**

where postprocessing improves the output's readability and fidelity.

AstraMT-Mixtral is defined as the primary system, integrating all four modular components—*CAPS*, *Prompt Constructor*, *Scoring and Reranking*, and *Postprocessing*—with the Mixtral LLM. To assess model agnostic effectiveness, variants such as AstraMT-GPT and AstraMT-Zephyr are instantiated AstraMT-GPT and AstraMT-Zephyr, where Mixtral is replaced by GPT-4 and Zephyr-7B respectively.

Unlike static few-shot prompting, AstraMT dynamically retrieves, structures, evaluates, and refines translations at inference time without requiring additional model fine-tuning. Experiments demonstrate that AstraMT consistently improves translation accuracy across backbones that AstraMT consistently improves translation accuracy across backbones, particularly in low-resource settings such as Assamese to English translation.

To better illustrate how AstraMT operates, an example input sentence is presented an example input sentence:

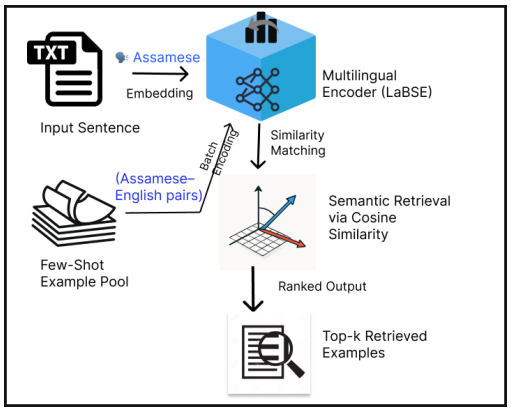

Fig. 2. CAPS uses LaBSE embeddings to retrieve the top-k most semantically similar Assamese–English example pairs based on cosine similarity

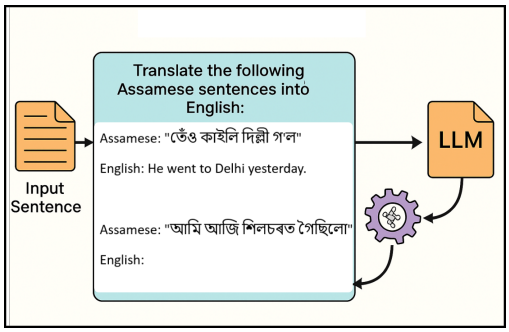

Fig. 3. Prompt Constructor module in AstraMT. The retrieved examples are formatted into an instruction-aligned prompt, which is passed to the LLM to generate translation candidates

আমি আজি চিলছার গৈছিলু (We went to Silchar today.)

Step 1: Semantic Retrieval

Given an input sentence for translation, the CAPS computes its dense representation using LaBSE embeddings. In parallel, each Assamese sentence in a curated bank of Assamese–English sentence pairs—sourced from held out portions of FLORES-200 and Samanantar is pre-encoded into the same embedding space. CAPS then compares the input embedding with these stored embeddings using cosine similarity and retrieves the top-k most contextually relevant examples. For this travel related sentence, CAPS may select examples such as: "তেও কালি দিল্লি গল" (He went to Delhi yesterday) or "আমি ঘুৰনীয়া ভ্ৰমনত আছিলু" (We were on a round trip) are selected based on cosine similarity in embedding space.

Table 3. Examples illustrating postprocessing corrections in AstraMT

Raw LLM Output	Issue	Postprocessed Output
We went to Silchar today	Named entity in Latin script	আজি দিনা আমি চিলছার গৈছিলু
He has went to market	Verb agreement error	He has gone to the market
She is reading a book .	Tokenization and punctuation spacing	She is reading a book.
We had gone to Silchar today .	Mixed tense / unnatural phrasing	We went to চিলছার today.

Step 2: Prompt Construction: These examples are formatted into a few-shot prompt following a fixed instruction template:

```
Translate the following Assamese
sentences into English:
Assamese:  ‘‘তেও কালি দিল্লি গল"
English:  He went to Delhi yesterday.
Assamese:  ‘‘আমি আজি শিলছার গৈছিলু"
English:
```

Step 3: LLM Inference: This prompt is then fed to an instruction-tuned model such as Mixtral, which produces a candidate translation such as "We went to Silchar today." Multiple prompt variants are used to obtain diverse outputs, enabling better coverage of syntactic and lexical options.

Step 4: Scoring and Reranking Each candidate translation is scored using the COMET metric, which estimates translation adequacy without requiring references. When reference translations are available, sentence-level BLEU is also computed. The reranker selects the top candidate with the highest predicted quality based on these scores. For example, among candidates such as "Today we went to Silchar," and "We had gone to Silchar today," the system selects the one with the highest COMET score.

Step 5: Postprocessing: The final output undergoes cleanup ensuring named entities (e.g., "Silchar") are transliterated properly, punctuation is restored, and tokenization artifacts are removed. The polished translation is: *"We went to Silchar today."*

This end to end process, executed without model fine-tuning, enables AstraMT to produce reliable translations in a fully modular, inference only setting.

4 Experimental Setup

This section describes the datasets, evaluation metrics, and model configuration details used to assess the performance of AstraMT and its baseline counterparts. The experiments were conducted in a fully inference time setup, without any parameter updates or fine-tuning.

4.1 Datasets

AstraMT was evaluate on two Assamese–English parallel datasets representing both clean benchmarks and real world scenarios. The first is the FLORES-200 devtest set [20], containing 1,012 high quality sentence pairs. As a standardized benchmark, it supports consistent evaluation using BLEU and COMET.

The second is a 1,000-sentence subset of the Samanantar corpus [24], sampled to include 4–25 token sentences and manually verified for alignment. Unlike FLORES, Samanantar includes noisier web and newswire content, offering a more realistic evaluation setting.

4.2 Model Configuration

Four model backends were evaluated in this study, encompassing both traditional supervised Neural Machine Translation (NMT) systems and instruction tuned LLMs. All LLMs were used in few-shot settings without any additional fine-tuning, leveraging in-context learning for translation. a small number of example Assamese–English sentence pairs (typically 3–4) are included within the prompt to demonstrate the translation task. These examples, retrieved from a curated bank using the CAPS module, precede the actual input sentence, whose English translation is left blank. The LLM is then expected to complete the output based on the demonstrated patterns, allowing it to generalize without modifying model weights.

AstraMT-Mixtral, the primary system, uses Mixtral-8$\times$7B-Instruct as its backbone. Mixtral is a sparse Mixture-of-Experts (MoE) Transformer model developed by Mistral AI, where two of eight expert subnetworks are activated per layer during inference. The HuggingFace Transformers interface was used with nucleus sampling (`temperature`=0.9, `top-p`=0.95) and generated three completions per prompt (`num_return_sequences`=3). Inference was performed on an NVIDIA RTX A6000 GPU with 47GB VRAM. The vanilla Mixtral model without AstraMT components was also evaluated using a static 4-shot prompt configuration as a baseline.

GPT-4 was accessed via the OpenAI API using a similar 4-shot prompt structure, with `temperature`=0.7 and three completions per input. Zephyr-7B-β, a smaller instruction aligned open source model, was tested with the same settings but limited to 3-shot prompts due to its shorter context window.

Each AstraMT variant: Mixtral, GPT-4, and Zephyr was evaluated within our complete modular pipeline, which includes CAPS based prompt selection, structured prompt construction, reranking via COMET and BLEU, and a final postprocessing step. For ablation purposes, all LLMs were also tested under vanilla few-shot prompting using static exemplars without any AstraMT enhancements. Table 4 provides a summary of the configurations, prompting strategies, and inference environments used in the experiments.

5 Results and Analysis

This section presents both quantitative and qualitative evaluations of AstraMT and its baselines. BLEU and COMET scores are reported on two test sets: FLORES-200 and Samanantar and the impact of key components such as CAPS and prompt size is analyzed. Finally, AstraMT is compared against existing Assamese–English NMT systems.

5.1 Main Results: FLORES-200 and Samanantar

Table 5 and Table 6 present BLEU and COMET scores for baseline LLMs and AstraMT variants on the FLORES-200 devtest set and a curated subset of the Samanantar corpus. AstraMT variants represent the full pipeline, incorporating context-aware prompt selection, multi-prompt reranking with COMET and BLEU, and final postprocessing to enhance fluency and named entity consistency.

AstraMT-Mixtral achieved the highest performance on both datasets, with BLEU 23.0 and COMET 0.71 on FLORES-200, and BLEU 21.3 and COMET 0.67 on Samanantar.

AstraMT-GPT and AstraMT-Zephyr also showed strong performance, consistently outperforming their static prompting counterparts. Notably, even the smaller Zephyr-7B-β benefited significantly from the modular pipeline.

Table 4. Model configurations and inference settings. Mixtral is the default backbone used in AstraMT unless otherwise specified. AstraMT variants apply the full pipeline: CAPS, reranking, and postprocessing

Model	**Model Type**	**Few-Shot Strategy**	**Inference Settings**
Hardware			
Mixtral-8×7B-Instruct (vanilla)	LLM (MoE, instruct)	Static 4-shot	`temperature`=0.9, `top-p`=0.95, `n`=3
NVIDIA RTX A6000			
AstraMT-Mixtral	**AstraMT (LLM + modules)**	**Dynamic via CAPS**	**Mixtral + CAPS + rerank + postprocess**
NVIDIA RTX A6000			
GPT-4	LLM (instruction-tuned)	Static 4-shot	`temperature`=0.7, `n`=3
OpenAI API			
Zephyr-7B-β	LLM (lightweight, instruct)	Static 3-shot	Same as Mixtral
NVIDIA RTX A6000			
AstraMT-Mixtral	AstraMT (LLM + modules)	Dynamic via CAPS	Mixtral + CAPS + rerank + postprocess
NVIDIA RTX A6000			
AstraMT-GPT	AstraMT (LLM + modules)	Dynamic via CAPS	GPT-4 + CAPS + rerank + postprocess
OpenAI API			
AstraMT-Zephyr	AstraMT (LLM + modules)	Dynamic via CAPS	Zephyr + CAPS + rerank + postprocess
NVIDIA RTX A6000			

Table 5. Performance on the FLORES-200 Assamese–English devtest set. All AstraMT variants use CAPS, reranking, and postprocessing

System	**LLM Type**	**CAPS**	**Rerank**	**Postproc**	**BLEU**	**COMET**
GPT-4	API	✗	✗	✗	17.2	0.61
Mixtral-8x7B	Open-source	✗	✗	✗	19.8	0.64
Zephyr-7B-β	Open-source	✗	✗	✗	18.4	0.62
AstraMT-GPT	API	✓	✓	✓	**22.5**	**0.69**
AstraMT-Mixtral	Open-source	✓	✓	✓	**23.0**	**0.71**
AstraMT-Zephyr	Open-source	✓	✓	✓	**21.7**	**0.68**

5.2 Ablation: CAPS and Prompt Size

To assess the contribution of the CAPS module, it was replaced with static prompts in AstraMT-Mixtral. As shown in Table 7, removing CAPS led to a 2.4 BLEU and 0.06 COMET drop on FLORES-200, underscoring the value of context-aware prompt selection.

The effect of prompt size (i.e., number of few-shot examples) on BLEU and COMET was also analyzed. Table 8 shows that performance improved with more shots, peaking around 4-shot settings. However, longer prompts increased input token length and latency, suggesting a trade-off between accuracy and efficiency.

5.3 Qualitative Examples

Table 9 provides representative translations from the FLORES-200 set, comparing baseline Mixtral with AstraMT-Mixtral. The latter produced more fluent, contextually accurate translations, demonstrating the value of dynamic prompting and reranking.

5.4 Comparison with Prior Assamese–English NMT Systems

AstraMT was compared with prior Assamese–English NMT systems (Table 11). Laskar et al. explored multiple Transformer-based approaches: their alignment aware model [14] integrated prior alignment and pre-trained

Table 6. Performance on the Samanantar Assamese–English subset. All AstraMT variants use CAPS, reranking, and postprocessing

System	LLM Type	CAPS	Rerank	Postproc	BLEU	COMET
GPT-4	API	✗	✗	✗	16.8	0.58
Mixtral-8x7B	Open-source	✗	✗	✗	18.9	0.61
Zephyr-7B-β	Open-source	✗	✗	✗	17.5	0.59
AstraMT-GPT	API	✓	✓	✓	**21.0**	**0.66**
AstraMT-Mixtral	Open-source	✓	✓	✓	**21.3**	**0.67**
AstraMT-Zephyr	Open-source	✓	✓	✓	**20.1**	**0.64**

Table 7. Ablation study of the CAPS module using AstraMT-Mixtral

Configuration	BLEU	COMET
AstraMT-Mixtral (no CAPS)	20.6	0.65
AstraMT-Mixtral (full)	**23.0**	**0.71**

Table 8. Effect of prompt size on AstraMT-Mixtral (FLORES-200)

Shots	BLEU	COMET	Tokens/Input	Latency (s)
1	20.1	0.65	120	1.5
2	21.4	0.67	190	2.2
3	22.8	0.70	260	3.0
4	**23.0**	**0.71**	320	3.6

Table 9. Qualitative translations comparing baseline Mixtral and AstraMT-Mixtral

Input (Assamese)	Mixtral	AstraMT-Mixtral
আমি আজি শিলছার গৈছিলু	Today went Silchar.	We went to Silchar today.
তিনি বজাত বজারটু কেনেকুয়া আসিল?	Three market how was?	How was the market at three o'clock?
মই কালি তুমাক লগ পাইছিলু।	I met you tomorrow.	I met you yesterday.
মই "শিলছার" পালু	I found " Silchar ".	I found "Silchar."

language models, achieving a BLEU of 18.44, while a domain-adapted version [13] combined curated corpora with monolingual augmentation, reaching 20.04. IndicTrans2 [8], trained on the Samanantar corpus [24], served as a multilingual baseline with BLEU scores between 12 and 20. In contrast, AstraMT-Mixtral, without any supervised fine-tuning, achieved a BLEU of 23.0, demonstrating the efficacy of LLM based few-shot translation augmented with CAPS and reranking.

5.5 Human Evaluation

A human evaluation was conducted on 100 randomly selected Assamese–English sentence pairs from the FLORES-200 devtest set to assess translation quality. Three bilingual annotators independently rated the outputs of AstraMT variants (Mixtral, GPT, Zephyr), static few-shot prompting, and IndicTrans2 across three dimensions adapted from Laskar et al. [13]: adequacy (faithfulness to source meaning), fluency (grammatical and natural expression in English), and overall quality (a composite judgment of adequacy and fluency). Each criterion was rated on a 1–5 Likert scale. Table 12 reports the average scores across systems. It is observed that AstraMT-Mixtral obtained the highest human ratings across all dimensions.

5.6 Translation Error Analysis

Finally, 50 outputs from AstraMT-Mixtral were manually annotated to identify common translation errors. As shown in Figure 4, the most frequent issues involved tense mismatches and literal phrasing. AstraMT occasionally produced incorrect verb tense, such as translating past events using the present form for example, "মই কালি আঁহিলু" was rendered as "I come yesterday" instead of the correct "I came yesterday." Named entity handling errors were also prevalent, where certain locations or names were over-transliterated or left untranslated, leading to awkward expressions. Literal phrasing was another common issue, with the model sometimes reproducing Assamese sentence structure too closely, as in the case of তিনি বজাত বজারটু কেনেকুয়া আসিল? becoming "Three market how was?" rather than the more fluent "How was the market at three o'clock?" Additionally,

Table 10. Qualitative translations from other models for reference

Input (Assamese)	GPT-4	Zephyr	IndicTrans2
আমি আজি শিলছার গৈছিলু	Today we went Silchar.	We are go to Silchar.	I have gone to Silchar.
তিনি বজাত বজারটু কেনেকুয়া আসিল?	How was the market at three?	What was market in 3 o'clock?	How was the market in the afternoon?
মই কালি তুমাক লগ পাইছিলু।	I met you tomorrow.	Yesterday I meet you.	I met you yesterday.
মই "শিলছার" পালু	I found " Silchar ".	I get Silchar.	I discovered Silchar.

Table 11. Comparison with prior Assamese–English NMT systems. BLEU reported for En→As

Model	Methodology Highlights	Corpus Size	BLEU
AstraMT-Mixtral (Ours)	Few-shot LLM + CAPS + Reranking + Postprocessing	∼387k + prompts	**23.0**
Laskar et al. (2022) [13]	Transformer + Data Augmentation + Postprocessing	387k + 1.3M aug.	20.04
Laskar et al. (2023) [14]	Transformer + Prior Alignment + Pretrained LM	210k	18.44
IndicTrans2 (AI4Bharat) [8]	Supervised Transformer + Samanantar + PMIndia	141k +	12.5
Samanantar Baseline [24]	Transformer + Bilingual Dictionary + Filtered Corpora	141k	10.2

Table 12. Average human evaluation scores (1–5 scale) across systems

System	Adequacy	Fluency	Overall Quality
AstraMT-Mixtral	**4.5**	**4.6**	**4.5**
AstraMT-GPT	4.3	4.4	4.3
AstraMT-Zephyr	4.1	4.2	4.1
IndicTrans2	3.7	3.8	3.7
Few-shot Mixtral (no CAPS)	3.9	4.0	3.9

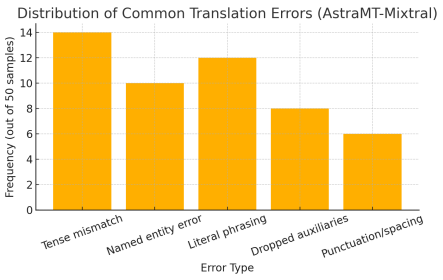

Fig. 4. Distribution of common translation errors observed in AstraMT-Mixtral outputs on FLORES-200

dropped auxiliaries such as "has," "is," or "was" occasionally resulted in grammatically incomplete translations. Finally, minor punctuation and tokenization artifacts particularly around commas, periods, and spacing were observed, especially in outputs from open source LLMs like Mixtral. These findings highlight specific linguistic challenges in low-resource Assamese–English translation and provide actionable insights for improving the reranking and postprocessing modules.

6 Conclusion

This paper presentedAstraMT, a modular, inference-time pipeline for few-shot translation of low-resource languages using instruction-tuned LLMs . Designed specifically for Assamese–English translation, AstraMT integrates four key components: a CAPS, dynamic prompt construction, multiprompt reranking with COMET/BLEU scoring, and post processing. This architecture enables significant improvements without requiring model fine-tuning. Experiments on the FLORES-200 and Samanantar datasets showed that AstraMT consistently performed better than both the baseline few-shot LLMs and the supervised IndicTrans2 model. Interestingly, AstraMT-Mixtral scored a BLEU of 23.0 on FLORES-200 and outperformed GPT-4 and other open-source systems. The significance of the CAPS module and the op-

Table 13. Examples of common translation errors made by AstraMT-Mixtral. Each row shows the original Assamese input (with transliteration), the AstraMT output, and a corrected reference.

Error Type	Input (Assamese)	AstraMT Output	Corrected Output
Tense Mismatch	মই কালি আহিলু *moi kali ahilu*	I come yesterday.	I came yesterday.
Named Entity Error	মই শিলছাৰই গলু *moi silcharloi golu*	I went to Silchar city.	I went to Silchar.
Literal Phrasing	তিনি বজাত বজাৰটু কেনেকুৱা আসিল? *tini bojat bojartu ki asil ?*	Three market how was?	How was the market at three o'clock?
Dropped Auxiliaries	তেও বজাৰলই গিসৈল *teo bojarloi goisil*	He to market gone.	He had gone to the market.
Punctuation Error	মই "শিলছাৰ" পালু *moi "Silchar" palu*	I found "Silchar "	I found "Silchar."

timal prompt size was demonstrated in ablation studies, whereas the assessment of changes in fluency, named entity handling, and grammaticality was also performed through qualitative analyses. Beyond accuracy, AstraMT is adaptable to multiple LLM backends and offers practical trade-offs between translation quality, latency, and token usage. These findings suggest AstraMT is well suited for low-resource, real world deployments where labeled data is scarce, and compute constraints exist. Future work will explore extending AstraMT to multilingual settings, refining reranking criteria, and integrating explainability features such as attention visualization and attribution methods to better interpret model predictions.

Data Availability

The code generated and the datasets used and/or analyzed in the present work are with the corresponding author and may be supplied upon a reasonable request.

References

1. **Ahmed, M. A., Kashyap, K., Talukdar, K., Boruah, P. A. (2023).** Iterative back translation revisited: An experimental investigation for low-resource English–Assamese neural machine translation. Proceedings of the 20th International Conference on Natural Language Processing (ICON), NLP Association of India (NLPAI), Goa University, Goa, India, pp. 172–179.
2. **An, S., Zhou, B., Lin, Z., Fu, Q., Chen, B., Zheng, N., Chen, W., Lou, J.-G. (2023).** Skill-based few-shot selection for in-context learning. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, pp. 13472–13492.
3. **Artetxe, M., Schwenk, H. (2019).** Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Transactions of the Association for Computational Linguistics, Vol. 7, pp. 597–610.
4. **Bahdanau, D., Cho, K., Bengio, Y. (2014).** Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
5. **Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020).** Language models are few-shot learners. Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020), Curran Associates Inc., Red Hook, NY, USA, pp. 159:1–159:25.
6. **do Carmo, F., Shterionov, D., Moorkens, J., Wagner, J., Hossari, M., Paquin, E., Schmidtke, D., Groves, D., Way, A. (2021).** A review of the state-of-the-art in automatic post-editing. Machine Translation, Vol. 35, No. 2, pp. 101–143. DOI: 10.1007/s10590-020-09252-y.
7. **Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W. (2022).** Language-agnostic bert sentence embedding. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, pp. 878–891.

8. **Gala, J., Chitale, P. A., K., R. A., Gumma, V., Doddapaneni, S., Kumar, A., Nawale, J., Sujatha, A., Puduppully, R., Raghavan, V., Kumar, P., Khapra, M. M., Dabre, R., Kunchukuttan, A. (2023).** IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages.
9. **Guerreiro, N. M., Rei, R., van Stigt, D., Coheur, L., Colombo, P., Martins, A. F. T. (2024).** xcomet: Transparent machine translation evaluation through fine-grained error detection. Transactions of the Association for Computational Linguistics, Vol. 12, pp. 979–995. DOI: 10.1162/tacl_a_00683.
10. **Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Bou Hanna, E., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Le Scao, T., Gervet, T., Lavril, T., Wang, T., Lacroix, T., El Sayed, W. (2024).** Mixtral of experts. arXiv preprint arXiv:2401.04088.
11. **Joshi, P., Santy, S., Budhiraja, A., Bali, K., Choudhury, M. (2020).** The state and fate of linguistic diversity and inclusion in the nlp world. ACL.
12. **Kumar, A., Puduppully, R., Dabre, R., Kunchukuttan, A. (2023).** Ctqscorer: Combining multiple features for in-context example selection for machine translation. Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, pp. 7736–7752.
13. **Laskar, S. R., Manna, R., Pakray, P., Bandyopadhyay, S. (2022).** A domain specific parallel corpus and enhanced English–Assamese neural machine translation. Computación y Sistemas, Vol. 26, No. 4, pp. 1669–1687. DOI: 10.13053/cys-26-4-4423.
14. **Laskar, S. R., Paul, B., Dadure, P., Manna, R., Pakray, P., Bandyopadhyay, S. (2023).** English–Assamese neural machine translation using prior alignment and pre-trained language model. Computer Speech & Language, Vol. 82, pp. 101524. DOI: 10.1016/j.csl.2023.101524.
15. **Le, T. D. e. a. (2023).** Mixtral of experts: Sparse mixture of experts with 8x7b llms. https://mistral.ai/news/mixtral-of-experts/.
16. **Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., Chen, W. (2022).** What makes good in-context examples for GPT-3? Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, Association for Computational Linguistics, Dublin, Ireland and Online, pp. 100–114.
17. **Manevich, A., Tsarfaty, R. (2024).** Mitigating hallucinations in large vision-language models (lvlms) via language-contrastive decoding (lcd). Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, pp. 6008–6022.
18. **Merx, R., Mahmudi, A., Langford, K., de Araujo, L. A., Vylomova, E. (2024).** Low-resource machine translation through retrieval-augmented llm prompting: A study on the mambai language. Proceedings of the 2nd Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia (EURALI) @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, pp. 1–11.
19. **Nath, B., Sarkar, S., Mukhopadhyay, S., Roy, A. (2024).** Improving neural machine translation by integrating transliteration for low-resource English–Assamese language. Natural Language Processing, Vol. 31, No. 2, pp. 306–327. DOI: 10.1017/nlp.2024.20.
20. **NLLB, T., Fan, A., Bhosale, S., Schwenk, H., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Gelly, S., Grave, E., Auli, M., Joulin, A. (2022).** No Language Left Behind: Scaling Human-Centered Machine Translation. arXiv preprint, Vol. arXiv:2207.04672.
21. **Och, F. J. (2003).** Minimum error rate training in statistical machine translation. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sapporo, Japan, pp. 160–167.
22. **OpenAI (2023).** Gpt-4 technical report. https://openai.com/research/gpt-4.
23. **Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002).** BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135.
24. **Ramesh, G., Doddapaneni, S., Bheemaraj, A., Jobanputra, M., AK, R., Sharma, A., Sahoo, S., Diddee, H., J, M., Kakwani, D., Kumar, N., Pradeep, A., Nagaraj, S., Deepak, K., Raghavan, V., Kunchukuttan, A., Kumar, P., Khapra, M. S.**

(2022). Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics, Vol. 10, pp. 145–162. DOI: 10.1162/tacl_a_00449.

25. **Rubin, O., Herzig, J., Berant, J. (2022).** Learning to retrieve prompts for in-context learning. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, pp. 2655–2671.
26. **Touvron, H. e. a. (2023).** Llama 2: Open foundation and fine-tuned chat models. https://ai.meta.com/llama/.
27. **Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., Wolf, T. (2023).** Zephyr: Direct distillation of LM alignment. arXiv preprint arXiv:2310.16944.
28. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. NeurIPS.
29. **Voita, E., Serdyukov, P., Sennrich, R., Titov, I. (2018).** Context-aware neural machine translation learns anaphora resolution. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, pp. 1264–1274.

Article received on 07/08/2025; accepted on 02/09/2025.
**Corresponding author is Basab Nath.*