WAT 2025

The 12th Workshop on Asian Translation

Proceedings of the Workshop

December 24, 2025

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-309-8

Preface

Many Asian countries are rapidly growing these days and the importance of communicating and exchanging the information with these countries has intensified. To satisfy the demand for communication among these countries, machine translation technology is essential.

Machine translation technology has rapidly evolved recently and it is seeing practical use especially between European languages. However, the translation quality of Asian languages is not that high compared to that of European languages, and machine translation technology for these languages has not reached a stage of proliferation yet. This is not only due to the lack of the language resources for Asian languages but also due to the lack of techniques to correctly transfer the meaning of sentences from/to Asian languages. Consequently, a place for gathering and sharing the resources and knowledge about Asian language translation is necessary to enhance machine translation research for Asian languages.

The Conference on Machine Translation (WMT), the world's largest machine translation conference, mainly targets on European language. The International Workshop on Spoken Language Translation (IWSLT) has spoken language translation tasks for some Asian languages using TED talk data, but there is no task for written language. The Workshop on Asian Translation (WAT) is an open machine translation evaluation campaign focusing on Asian languages. WAT gathers and shares the resources and knowledge of Asian language translation to understand the problems to be solved for the practical use of machine translation technologies among all Asian countries. WAT is unique in that it is an open innovation platform": the test data is fixed and open, so participants can repeat evaluations on the same data and confirm changes in translation accuracy over time. WAT has no deadline for the automatic translation quality evaluation (continuous evaluation), so participants can submit translation results at any time.

Following the success of the previous WAT workshops (WAT2014 – WAT2024), WAT2025 will bring together machine translation researchers and users to try, evaluate, share and discuss brand-new ideas about machine translation. For the 12th WAT, we have 1 Patent Translation Task, 2 Document Translation Tasks and 5 Multimodal Translation Tasks. We have 6 teams who submitted their results.

In addition to the shared tasks, WAT2025 also features research papers on topics related to machine translation, especially for Asian languages. We received 7 research papers submitted including ARR commitment, and the program committee accepted 4 research papers.

We would like to thank all the authors who submitted papers. We express our deepest gratitude to the committee members for their timely reviews. We also thank the AACL-IJCNLP 2025 organizers for their help with administrative matters.

WAT 2025 Organizers

Organizing Committee

Organizers

Toshiaki Nakazawa, The University of Tokyo, Japan
Isao Goto, Ehime University, Japan
Hidaya Mino, Japan Broadcasting Corporation (NHK), Japan
Naoto Shirai, Japan Broadcasting Corporation (NHK), Japan
Chenhui Chu, Kyoto University, Japan
Haiyue Song, National Institute of Information and Communications Technology (NICT), Japan
Raj Dabre, Google Research Australia, Australia
Shohei Higashiyama, National Institute of Information and Communications Technology (NICT), Japan
Anoop Kunchookuttan, Microsoft AI and Research, India
Shantipriya Parida, AMD Silo AI, Finland
Ondřej Bojar, Charles University, Prague, Czech Republic
Katsuhito Sudoh, Nara Women's University, Japan
Masaaki Nagata, NTT Corporation, Japan
Sadao Kurohashi, National Institute of Informatics, Japan

Technical Collaborators

Luis Fernando D'Haro, Universidad Politécnica de Madrid, Spain
Rafael E. Banchs, Nanyang Technological University, Singapore
Haizhou Li, National University of Singapore, Singapore
Chen Zhang, National University of Singapore, Singapore

Program Committee

Program Committee Members

Raj Dabre, Google Research Australia, Australia
Shohei Higashiyama, NICT
Chao-Hong Liu, Potamu Research Limited
Hideya Mino, NHK
Takashi Ninomiya, Ehime University
Shantipriya Parida, AMD Silo AI
Katsuhito Sudoh, Nara Women's University
Masao Utiyama, NICT

Keynote Talk

Optimizing Large Language Models for Low-resource Quality Estimation

Diptesh Kanojia
University of Surrey

2025-12-24 11:40:00 –

Abstract: Large Language Models (LLMs) are positioned as generalist models often claiming superlative performance on many Natural Language Processing (NLP) tasks. However, they tend to fail at Quality Estimation (QE) of Machine Translation (MT), particularly for low-resource languages. The talk investigates root causes of this disparity, such as tokenization inconsistencies arising from morphological richness in natural languages. To bridge this gap, the talk introduces strategies to embed annotation guidelines-based reasoning constraints directly in-context. Furthermore, our investigation on optimal cross-lingual alignment shows that intermediate Transformer layers help produce performant adapters. By attaching Low-Rank Adapter (LoRA) based regression heads to intermediate layers, we bypass the generation-specific biases of the final layer, efficiently outperforming standard instruction fine-tuning and SoTA encoders like COMETKiwi. Finally, via results from the WMT Unified Shared subtask on QE-informed Correction, we demonstrate that these precise estimations can guide LLMs to produce reliable corrections. We discuss how these signals help address the diminishing returnschallenge, enabling models to improve fluent outputs without diverging from human references.

Bio: Researcher working on problems within areas of Natural Language Processing (NLP) and Machine Learning (ML) at the Institute for People-Centred AI (PAI) and School of Computer Science and Electronic Engineering. As a research lead, I manage the NLP subgroup within the Nature Inspired Computing and Engineering group (NICE) @ Computer Science Research Centre. I also lead teaching on the NLP module offered to both undergraduate and postgraduate students.
My research focuses on developing scalable and safe human-machine interaction using foundation models. Guided by the principles of Responsible and Inclusive AI, my work emphasises cross-lingual and multimodal representation learning to address challenges like online toxicity, misinformation, and digital accessibility for low-resource languages. Our research outcomes- code, data, and models, are publicly available on the SurreyNLP GitHub and HuggingFace.
My prior roles include a Postdoctoral Fellowship at the Centre for Translation Studies, a joint PhD from IIT Bombay and Monash University, and Research Engineer at the CFILT Lab.

Table of Contents

Findings of the First Patent Claims Translation Task at WAT2025
Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Kazuhiro Kasada, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda and Masaaki Nagata 1

Ehime-U System with Judge and Refinement, Specialized Prompting, and Few-shot for the Patent Claim Translation Task at WAT 2025
Taishi Edamatsu, Isao Goto and Takashi Ninomiya 16

UTSK25 at WAT2025 Patent Claims Translation/Evaluation Task
Haruto Azami, Yin Zhang, Futo Kajita, Nobuyori Nishimura and Takehito Utsuro 30

Segmentation Beyond Defaults: Asymmetrical Byte Pair Encoding for Optimal Machine Translation Performance
Saumitra Yadav and Manish Shrivastava 35

Speech-to-Speech Machine Translation for Dialectal Variations of Hindi
Sanmay Sood, Siddharth Rajput and Md Shad Akhtar 54

A Systematic Review on Machine Translation and Transliteration Techniques for Code-Mixed Indo-Aryan Languages
H. Rukshan Dias and Deshan Sumanathilaka 66

CycleDistill: Bootstrapping Machine Translation using LLMs with Cyclical Distillation
Deepon Halder, Thanmay Jayakumar and Raj Dabre 78

Findings of the WAT 2025 Shared Task on Japanese-English Article-level News Translation
Naoto Shirai, Kazutaka Kinugawa, Hitoshi Ito, Hideya Mino and Yoshihiko Kawai 93

NHK Submission to WAT 2025: Leveraging Preference Optimization for Article-level Japanese–English News Translation
Hideya Mino, Rei Endo and Yoshihiko Kawai 98

Findings of WAT2025 English-to-Indic Multimodal Translation Task
Shantipriya Parida and Ondřej Bojar 103

OdiaGenAI participation at WAT 2025
Debasish Dhal, Sambit Sekhar, Revathy V R, Shantipriya Parida and Akash Kumar Dhaka . . 109

Does Vision Still Help? Multimodal Translation with CLIP-Based Image Selection
Deepak Kumar, Baban Gain, Kshetrimayum Boynao Singh and Asif Ekbal 115

A Picture is Worth a Thousand (Correct) Captions: A Vision-Guided Judge-Corrector System for Multimodal Machine Translation
Siddharth Betala, Kushan Raj, Vipul Betala and Rohan Saswade 124

Program

Wednesday, December 24, 2025

09:25 - 09:30 *Welcome*

09:30 - 10:30 *Shared Task - Patent Claims Translation*

Findings of the First Patent Claims Translation Task at WAT2025
Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Kazuhiro Kasada, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda and Masaaki Nagata

Ehime-U System with Judge and Refinement, Specialized Prompting, and Few-shot for the Patent Claim Translation Task at WAT 2025
Taishi Edamatsu, Isao Goto and Takashi Ninomiya

UTSK25 at WAT2025 Patent Claims Translation/Evaluation Task
Haruto Azami, Yin Zhang, Futo Kajita, Nobuyori Nishimura and Takehito Utsuro

10:30 - 11:00 *Break*

11:00 - 11:40 *Research Paper I*

Segmentation Beyond Defaults: Asymmetrical Byte Pair Encoding for Optimal Machine Translation Performance
Saumitra Yadav and Manish Shrivastava

Speech-to-Speech Machine Translation for Dialectal Variations of Hindi
Sanmay Sood, Siddharth Rajput and Md Shad Akhtar

11:40 - 12:30 *Invited Talk*: *Optimizing Large Language Models for Low-resource Quality Estimation by Diptesh Kanojia*

12:30 - 14:10 *Lunch Break*

14:10 - 14:50 *Research Paper II*

A Systematic Review on Machine Translation and Transliteration Techniques for Code-Mixed Indo-Aryan Languages
H. Rukshan Dias and Deshan Sumanathilaka

Wednesday, December 24, 2025 (continued)

CycleDistill: Bootstrapping Machine Translation using LLMs with Cyclical Distillation
Deepon Halder, Thanmay Jayakumar and Raj Dabre

14:50 - 15:30 *Shared Task - Japanese-English Article-level News Translation*

Findings of the WAT 2025 Shared Task on Japanese-English Article-level News Translation
Naoto Shirai, Kazutaka Kinugawa, Hitoshi Ito, Hideya Mino and Yoshihiko Kawai

NHK Submission to WAT 2025: Leveraging Preference Optimization for Article-level Japanese–English News Translation
Hideya Mino, Rei Endo and Yoshihiko Kawai

15:30 - 16:00 *Break*

16:00 - 17:20 *Shared Task - English-to-Indic Multimodal Translation*

Findings of WAT2025 English-to-Indic Multimodal Translation Task
Shantipriya Parida and Ondřej Bojar

OdiaGenAI participation at WAT 2025
Debasish Dhal, Sambit Sekhar, Revathy V R, Shantipriya Parida and Akash Kumar Dhaka

Does Vision Still Help? Multimodal Translation with CLIP-Based Image Selection
Deepak Kumar, Baban Gain, Kshetrimayum Boynao Singh and Asif Ekbal

A Picture is Worth a Thousand (Correct) Captions: A Vision-Guided Judge-Corrector System for Multimodal Machine Translation
Siddharth Betala, Kushan Raj, Vipul Betala and Rohan Saswade

17:20 - 17:25 *Closing*

Findings of the First Patent Claims Translation Task at WAT2025

Toshiaki Nakazawa
The University of Tokyo
nakazawa@nlab.ci.i.u-tokyo.ac.jp

Takashi Tsunakawa
Shizuoka University
tuna@inf.shizuoka.ac.jp

Isao Goto
Ehime University
goto.isao.fn@ehime-u.ac.jp

Kazuhiro Kasada
Japan Patent Information Organization
kasada.kazuhiro.cn@japio.or.jp

Katsuhito Sudoh
Nara Women's University
sudoh@ics.nara-wu.ac.jp

Shoichi Okuyama
Patent Attorney in Japan
okuyama@quon-ip.jp

Takashi Ieda
Japan Institute for
Promoting Invention and Innovation
t-ieda@jiii.or.jp

Masaaki Nagata
NTT Inc.
masaaki.nagata@ntt.com

Abstract

This paper presents the results and findings of the first shared task of translating patent claims. We provide training, development, and test data for participants and perform human evaluation of the submitted translations. This time, 2 teams submitted their translation results. Our analysis of the human-annotated translation errors revealed not only general, domain-independent errors but also errors specific to patent translation. We also found that the human annotation itself exhibited some serious issues. In this paper, we report on these findings.

1 Introduction

The performance of machine translation using Neural Machine Translation (NMT) and Large Langauge Models (LLMs) has improved dramatically and in some cases can even surpass human translation depending on the language and domain. However, currently there is no universal method to accurately evaluate the performance of machine translation. Even widely used metrics such as COMET (Rei et al., 2020) have been reported to yield unstable or inaccurate evaluation results (Kocmi et al., 2025) when applied to translations of texts from domains other than those used in COMET's training.

The same applies to the translation of patent documents. Although the average translation quality has improved significantly, it remains difficult to accurately evaluate aspects such as appropriate terminology usage and term consistency. In particular, patent claims present additional challenges due to their length and distinctive writing style, making an accurate evaluation even more difficult.

Therefore, we conducted a Shared Task focusing on translating Japanese-English patent claims[1]. The goal is not only to compete on translation quality, but also to ultimately develop an automatic evaluation method that can accurately assess translation results.

For this first iteration, our primary objective is to collect translation outputs produced by various methods and annotate them with human-identified errors, thereby creating training data for future development of models capable of accurately performing automatic evaluation of translations in the patent domain.

2 Training Data

We used the publicly available subset of JaParaPat, the Japanese-English Parallel Patent Application Corpus (Nagata et al., 2024), as the training data for the shared task. In August 2025, the authors released a subset of JaParaPat, covering the period from 2016 to 2020, which comprises more than 100 million sentence pairs, for research purposes.[2]

JaParaPat is made from the publication of unex-

[1] https://sites.google.com/view/pat-claims-trans-2025/

[2] https://www.kecl.ntt.co.jp/icl/lirg/japarapat/

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 1–15
December 24, 2025 ©2025 Association for Computational Linguistics

	jp-us	jp-x-us	us-jp	pct	sum
2016	7,241,502	1,322,124	1,181,150	10,287,313	20,032,089
2017	7,892,204	1,399,012	1,226,177	10,354,135	20,871,528
2018	7,639,692	1,262,972	1,044,728	11,171,128	21,118,520
2019	8,867,148	1,450,851	1,157,361	11,625,720	23,101,080
2020	8,617,540	1,570,684	1,088,832	10,843,470	22,120,526
sum	40,258,086	7,005,643	5,698,248	54,281,766	107,243,743

Table 1: Number of sentence pairs

amined patent applications from the Japan Patent Office (JPO) and the United States Patent and Trademark Office (USPTO) from 2000 to 2021. They are aligned based on patent family information from the DOCDB, a bibliographic database maintained by the European Patent Office (EPO).

Table 1 shows the number of sentence pairs available in the public version of JaParaPat. There are two primary routes for filing international patent applications: the Paris Convention route and the Patent Cooperation Treaty (PCT) route. JaParaPat includes data from both routes. In Table 1, within the Paris route, ‘ jp-us ’ refers to patent pairs first filed in Japan and subsequently in the United States. ‘ us-jp ’ refers to those first filed in the United States and then in Japan. ‘ jp-x-us ’ refers to patents initially filed in a country other than Japan or the United States, and subsequently filed in both Japan and the United States. The public version employs different methods for document alignment, sentence segmentation, and sentence alignment, resulting in a different number of sentence pairs compared to Table 1 in the original JaParaPat paper.

As the training data for the shared task of Patent Claim Translation, one of the most important problems of JaParaPat is its sentence segmentation and alignment for patent claims. It often segments a long claim into segments by a new line and provides segment-level alignment, which makes it difficult to reconstruct claim-level alignment. We are discussing with the authors of JaParaPat how to solve this problem.

3 Development Data

This time we focused on claims rather than specification to see how different engines will handle relatively difficult sentence structures, technical terms, non-technical terms, ambiguous language (i.e. phrases that can be interpreted in more ways than one), etc. Claims serving as development data were selected from existing patent application documents. In the selection, we mainly considered the following factors as elements impacting the difficulty of translation:

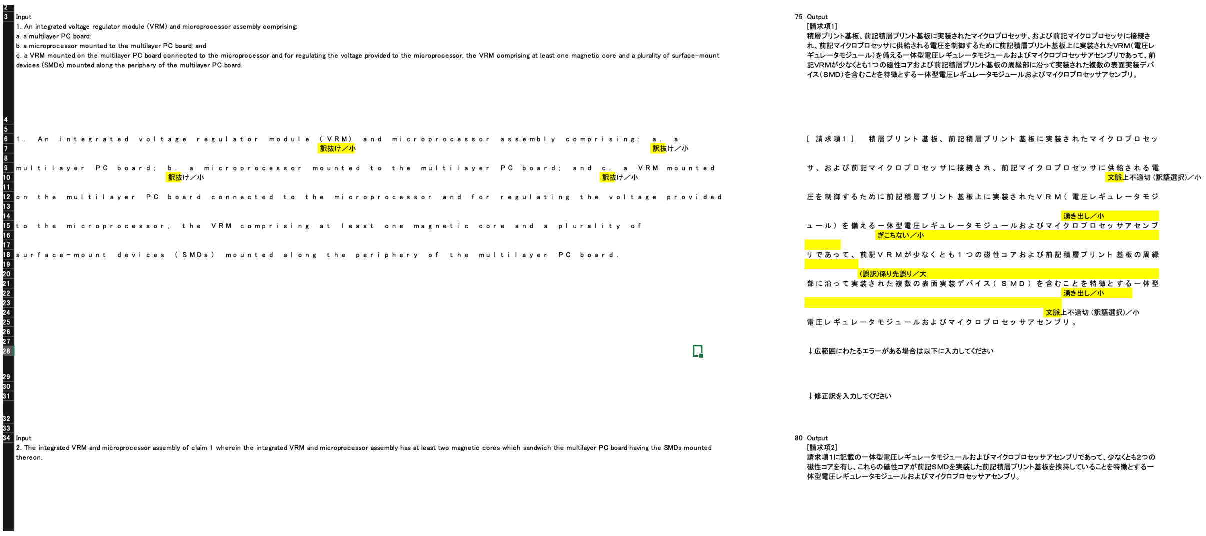
- Paragraph length
- Term peculiarity
- Construction
- Structural/semantic ambiguity (e.g. whether a given phrase should be interpreted as "A including B, and C (not included in A)", or "A including both B and C")
- Terminological ambiguity (e.g. whether the term "対向" in Japanese means "opposing", "reverse", "facing", etc.)
- Whether a term has a corresponding term/concept in target language
- Existence/lack of an official translation (e.g. a US application having a corresponding JP application)

Based on these criteria, we selected 13 Japanese documents and 11 English documents for this study. Example of development data is shown in Table 7.

We translated the development data using two types of translation engines: an NMT model trained on JaParaPat and an open-weight LLM, and conducted a preliminary human evaluation using this data. The purpose was to determine appropriate evaluation procedures and the feasible level of granularity prior to performing the main evaluation using the test data.

Figure 1 shows the excel interface of the human evaluation. We instructed the annotators to perform the following three tasks:

1. Highlight segments containing translation errors or input issues and specify the corresponding error category within the cell.



Input
1. An integrated voltage regulator module (VRM) and microprocessor assembly comprising:
a. a multilayer PC board;
b. a microprocessor mounted to the multilayer PC board; and
c. a VRM mounted on the multilayer PC board connected to the microprocessor and for regulating the voltage provided to the microprocessor, the VRM comprising at least one magnetic core and a plurality of surface-mount devices (SMDs) mounted along the periphery of the multilayer PC board.

1. An integrated voltage regulator module (VRM) and microprocessor assembly comprising: a. a
訳抜け／小 訳抜け／小
multilayer PC board; b. a microprocessor mounted to the multilayer PC board; and c. a VRM mounted
訳抜け／小 訳抜け／小
on the multilayer PC board connected to the microprocessor and for regulating the voltage provided
to the microprocessor, the VRM comprising at least one magnetic core and a plurality of
surface-mount devices (SMDs) mounted along the periphery of the multilayer PC board.

75 Output
[請求項1]
積層プリント基板、前記積層プリント基板に実装されたマイクロプロセッサ、および前記マイクロプロセッサに接続され、前記マイクロプロセッサに供給される電圧を制御するために前記積層プリント基板上に実装されたVRM（電圧レギュレータモジュール）を備える一体型電圧レギュレータモジュールおよびマイクロプロセッサアセンブリであって、前記VRMが少なくとも1つの磁性コアおよび前記積層プリント基板の周縁部に沿って実装された複数の表面実装デバイス（SMD）を含むことを特徴とする一体型電圧レギュレータモジュールおよびマイクロプロセッサアセンブリ。

［請求項１］　積層プリント基板、前記積層プリント基板に実装されたマイクロプロセッ
サ、および前記マイクロプロセッサに接続され、前記マイクロプロセッサに供給される電
文脈上不適切 (訳語選択)／小
圧を制御するために前記積層プリント基板上に実装されたＶＲＭ（電圧レギュレータモジ
湧き出し／小
ュール）を備える一体型電圧レギュレータモジュールおよびマイクロプロセッサアセンブ
ぎこちない／小
リであって、前記ＶＲＭが少なくとも１つの磁性コアおよび前記積層プリント基板の周縁
(誤訳)係り先誤り／大
部に沿って実装された複数の表面実装デバイス（ＳＭＤ）を含むことを特徴とする一体型
湧き出し／小
文脈上不適切 (訳語選択)／小
電圧レギュレータモジュールおよびマイクロプロセッサアセンブリ。

↓広範囲にわたるエラーがある場合は以下に入力してください

↓修正訳を入力してください

Input
2. The integrated VRM and microprocessor assembly of claim 1 wherein the integrated VRM and microprocessor assembly has at least two magnetic cores which sandwich the multilayer PC board having the SMDs mounted thereon.

80 Output
[請求項2]
請求項1に記載の一体型電圧レギュレータモジュールおよびマイクロプロセッサアセンブリであって、少なくとも2つの磁性コアを有し、これらの磁性コアが前記SMDを実装した前記積層プリント基板を挟持していることを特徴とする一体型電圧レギュレータモジュールおよびマイクロプロセッサアセンブリ。

Figure 1: Human evaluation interface.

2. Assign a holistic quality score to the translation on a 100-point scale.
3. Post-edit the translation.

The post-edited translations were used as reference translations to form parallel data, which we provided as development data.

4 Test Data

Source texts in Japanese and English were selected from existing patent applications. We have considered the following factors when selecting source texts.

- Type of machine translation: The type of translation was estimated to be neural machine translation (NMT) or large language model (LLM)-based translation.
- Length/construction: It is known that a longer single text without a line break may result in poorer translation quality (Kondo et al., 2021). The primary purpose of this research was not to examine how different engines would deal with length, but to see if general claim wording, which may contain one or more of the factors mentioned above or below, will be handled. As such, we selected source texts that generally contained no more than about 220 English words or 500 Japanese characters with or without one or more line breaks in them. The purpose of including a few longer texts was to see how a relatively long text would be processed.
- Existing translation: A patent application may have a family including a corresponding application in another language; for example, an application filed to the Japan Patent Office (JPO) may have a corresponding application filed to the United States Patent and Trademark Office (USPTO). Applications in the same family are linked in some search engines including Google Patents. An LLM may be able to locate an official translation of an application, i.e. correct solution, through such search engines if the application has a family. We therefore selected source texts from applications that did not have a corresponding application in the target language at least at the time when the source texts were distributed to the participants.
  Because of this factor, we cannot automatically collect reference translations from publicly available data. In addition, we do not have sufficient budget to create reference translations for the test data. Therefore, as described in Section 6, we conducted reference-free automatic evaluation (i.e., quality estimation).
- Field: The source texts come from applications in a variety of fields including information processing, communication, electric engineering, chemistry, etc.
- Ambiguity/parsing: Machine translation is processing that is based essentially or entirely on natural language information. The processing is not expected to rely on visual

or other non-natural language-based information. Meanwhile, claim wording sometimes requires reference to information based on other than natural language, a typical example of which is drawings that patent applications often contain. As the USPTO Patent Application Filing Guide states "*a patent application is required to contain drawings if drawings are necessary to understand the subject matter*", natural language *per se* could be insufficient to arrive at a correct interpretation of claim wording. In addition, there are also cases where reference to the specification is necessary to fully understand the meaning of a claim. For instance, with the phrase "a device comprising a controller that has an analyzer, a processor, and a memory", it may be necessary to refer to the specification to determine whether the "processor" and "memory" are part of the "device" or the "controller".

For the current project, we have selected source texts, the content of which was — at least to the persons in charge of the selection — comprehensible on its own without additional information. The selected source texts contain ambiguous terms such as 区間 in Japanese, which can be interpreted as a temporal concept (interval: period between two times) or dimensional concept (interval: space between two points). We allowed for the inclusion of such terms only where it was possible to ascertain the meaning of a term from the context. For example, the aforementioned 区間 is stated in the claim in which the term appears to be a section of a road (a physical interval within a road) on which a vehicle travels. So, it should be obvious that the term does not mean a temporal concept.

Selecting a source text that does not require additional information to interpret is also beneficial from the perspective of evaluating the translation: A satisfactory evaluation by either a human or non-human evaluator should be possible without additional information. This means that the respective evaluation abilities of a human evaluator and a non-human evaluator can be put to comparison essentially on the basis of their abilities to process natural language without additional information.

Team ID	Organization	Country	J-E	E-J
UTSK25	University of Tsukuba	Japan	1	3
EHIME-U	Ehime University	Japan	12	0
Commercial 1	online service	n/a	1	1
Commercial 2	closed system	n/a	1	1
Commercial 3	free LLM model for MT	n/a	1	1

Table 2: List of participants and the number of submissions for each direction. For the commercial systems, the organizers collected the translations.

Taking the above factors into consideration, we prepared 26 documents with 70 claims for the Japanese–English direction and 30 documents with 81 claims for the English–Japanese direction as the test data.

5 Participants

Table 2 shows the list of participants and the number of submissions from each system. The organizers collected the translations of the commercial systems. Whereas the UTSK25 conducted continual pretraining of an open-weight LLM on JaParaPat, Ehime University performed prompt tuning on a closed/proprietary LLM. For Commercial 1 we used a standard translation prompt. For Commercial 3 we performed translation using the chat template provided in its accompanying documentation. Commercial 2 is a closed system.

We selected 1 submission for each translation direction for all the systems except EHIME-U for the human evaluation. For EHIME-U, we selected 2 submissions for Ja-En because they did not submit any result for En-Ja.

6 Automatic Evaluation

Automatic evaluation of MT has been studied for a long time, along with the evolution of MT technologies. It faces new challenges, such as very long and complex claim sentences in our task. For the first attempt, we conducted the automatic evaluation in a reference-free manner using MetricX-24-Hybrid-XL[3] (Juraska et al., 2024) and WMT23-CometKiwi-DA-XL[4] (Rei et al., 2023) because the corresponding translations of the test set were not available, as mentioned above. We had two variants of automatic evaluation: *segment-level* (claim-by-claim) and *document-level*. The document-level evaluation

[3] https://github.com/google-research/metricx

[4] https://github.com/Unbabel/COMET

System	ja-en		en-ja	
	MetricX ↓	CometKiwi ↑	MetricX ↓	CometKiwi ↑
UTSK25	3.761 $\pm$1.654	0.544 $\pm$0.122	3.623 $\pm$1.474	0.641 $\pm$0.111
EHIME-U 1	2.882 $\pm$1.614	0.560 $\pm$0.134	n/a	n/a
EHIME-U 2	2.978 $\pm$1.607	0.568 $\pm$0.131	n/a	n/a
Commercial 1	2.792 $\pm$1.416	0.572 $\pm$0.133	2.916 $\pm$0.842	0.681 $\pm$0.088
Commercial 2	3.879 $\pm$2.454	0.567 $\pm$0.139	3.126 $\pm$1.031	0.676 $\pm$0.093
Commercial 3	2.920 $\pm$1.107	0.573 $\pm$0.127	2.581 $\pm$0.780	0.707 $\pm$0.078

Table 3: Segment-level automatic evaluation results

System	ja-en		en-ja	
	MetricX ↓	CometKiwi ↑	MetricX ↓	CometKiwi ↑
UTSK25	4.669 $\pm$1.439	0.313 $\pm$0.128	4.577 $\pm$1.605	0.489 $\pm$0.118
EHIME-U 1	3.827 $\pm$1.392	0.308 $\pm$0.110	n/a	n/a
EHIME-U 2	4.071 $\pm$1.613	0.305 $\pm$0.106	n/a	n/a
Commercial 1	3.471 $\pm$1.003	0.279 $\pm$0.123	3.435 $\pm$0.817	0.539 $\pm$0.093
Commercial 2	5.303 $\pm$2.153	0.259 $\pm$0.139	4.022 $\pm$1.025	0.525 $\pm$0.126
Commercial 3	3.568 $\pm$0.871	0.298 $\pm$0.127	3.183 $\pm$0.751	0.567 $\pm$0.098

Table 4: Document-level automatic evaluation results

considered the whole document as a single segment.

Tables 3 and 4 show average segment- and document-level scores, respectively.

7 Human Evaluation

Due to budget constraints, human evaluation was conducted only on a subset of the test data. The selection of evaluation files followed the same Diversity Sampling procedure used in the WMT25 General Machine Translation Shared Task(Kocmi et al., 2025), resulting in 13 files per translation direction.

Table 8 and 9 in Appendix A.2 shows the human evaluation criteria we used. We made several modifications to Freitag's metric (Freitag et al., 2021) to better adapt it to the patent-translation domain. We also referred to the MQM website[5] for the descriptions and examples. Categories shown with a gray background were deemed unnecessary for patent translation and were therefore excluded.

8 Official Results

Table 5 shows the average socre of the human evaluation. There was no system that achieved the best accuracy in both translation directions. On average, Commercial 1 exhibited the highest accuracy.

System	ja-en	en-ja
UTSK25	63.04	79.29
EHIME-U 1	81.61	n/a
EHIME-U 2	86.07	n/a
Commercial 1	87.68	70.00
Commercial 2	66.96	60.71
Commercial 3	67.50	54.11

Table 5: Average score of the human evaluation.

Table 6 shows the correlation coefficients between human evaluation and each automatic evaluation measure. Surprisingly, none of the metrics showed substantial correlation with the human evaluation. Several factors may account for this outcome:

1. Both automatic evaluation methods used in this study are reference-free, which may limit their ability to accurately assess translation quality.

2. These automatic evaluation methods may not function effectively in the patent domain.

3. The human evaluations themselves may contain inaccuracies (we discuss this in detail in

[5] https://themqm.org/the-mqm-full-typology/

Measure	ja-en	en-ja
MetricX (seg)	-0.235	-0.121
MetricX (doc)	-0.230	-0.023
CometKiwi (seg)	0.288	0.186
CometKiwi (doc)	0.029	-0.079

Table 6: Correlation coefficients between human evaluation and each automatic evaluation measure.

the Discussion section).

9 Discussion

Our analysis of the translation outputs and human annotations revealed various issues on both the translation side and the annotation side. In this section, we discuss several of these problems.

The selected source texts contained several phrases which could be interpreted or rendered in more ways than one yet the correct meaning or valid rendition of which could be derived from the context. A few examples of such phrases will be observed below along with annotations they were marked with. In view of the following examples, we shall focus on two issues that are broadly applicable to translation in general and more specifically to patent translation, namely "use of generic or specific terms" and "differences in routines/legal restrictions between Countries/intellectual property (IP) offices".

9.1 Use of Generic or Specific Terms

> Source: "前記信頼度情報が予め設定された閾値よりも小さい状態が継続している区間を補正対象区間として検知して"
> "… 前記運動状態推定部は、… 前記補正対象区間を走行している前記他車両の運動状態を推定し"

A technically correct translation should be something along the lines of:

> TR: "… detects, as a correction target section, a section in which the confidence information continues to remain below a preset threshold"
> "the motion-state estimation unit estimates the motion state of the other vehicle traveling through the correction target section"

Note that the discussion below focuses on the term "区間", which can be rendered into a number of terms including "section", "interval", "segment", "portion" or the like as long as it is clear that the term refers to a physical segment of a road, not to a time interval. From the second phrase above stating that the other vehicle travels through this section, it should be obvious that the section is not a time interval.

The following is a machine translation produced by one of the six engines.

> sys: "… detect, as a correction-target section, a section during which the reliability information remains less than the predetermined threshold"
> "the motion state estimation unit is configured to estimate the motion state of the other vehicle traveling in the correction-target section"

Renditions of the underlined phrase by other engines include:

- detects, as a correction-target section, a section in which
- detect, as a correction target section, a period during which
- detects the interval during which … as a correction target interval

Both nouns "section" and "interval" on their own could be either a physical or temporal concept. In the above context, the preposition (plus relative pronoun), i.e. "during (which)" or "in (which)", is decisive in whether the preceding noun will be interpreted as a physical or temporal concept. For the example above, it can be said that while "during" is incorrect, "in" is ambiguous (i.e. can be interpreted in more ways than one) yet potentially correct (i.e. encompasses the correct meaning). Choosing a specific term is preferable if the concept including the term is unambiguous, but if a concept is ambiguous, choosing a generic term may increase the chance of the concept being interpreted correctly.

Multiple human annotators, who must have been exposed to the concept that the "section" is a segment of a road on which a vehicle travels, did not leave any annotation to the expression "a section during which".

The following are a few examples of ambiguous terms that are often used in patent-related documents.

- 挟まれる (*hasamareru*): The term means an either physical or conceptual entity being located, interposed, or held between two or more other physical or conceptual entities. It is often rendered as "sandwiched" but incorrectly in some contexts. A generic term suggesting a location between two or more entities, e.g. simply "between", may be more suitable in some cases.

- (〜である) が ((*dearu*) *ga*): This is a highly context-sensitive particle and could mean "but", "and", "whereas", "yet", "thus", "in this regard/respect", etc. connecting the phrases before and after it to some degree and in some way. It is often rendered as "but/however", but expressions such as "in this regard/respect" may be a better option in some contexts. Moreover, the term can often be omitted entirely.

- 対象 (*taisho*): One of the most ambiguous yet convenient terms to refer to something that the writer of a text wants to refer to. "⋯ in question" should be one of the most generic English equivalents, but it can make the translation vague. In some cases, it may be necessary to explicitly say what the writer wants to refer to by converting the term into a more specific concept.

See Appendix A.3 for more details.

9.2 Differences in Routines/Legal Restrictions between Countries/IP offices

> Source: ⋯プログラムであって、
> コンピュータを、
> ⋯クリアデッキを記憶する記憶手段、
> ⋯一のクリアデッキを編成できるか否かを判
> 定する判定手段、
> ⋯コンテンツを特定コンテンツとして特定す
> る特定手段、
> ⋯取得画面を表示させる制御手段、
> として機能させる、
> プログラム。

A more or less literal/mirror translation would be something along the lines of:

> TR: A program ⋯, the program causing a computer to function as
> ⋯a storage means that stores a clear deck ⋯,
> ⋯a determination means that determines whether one clear deck can be organized ⋯,
> ⋯a specifying means that specifies, as specific content, content that is ⋯, and
> ⋯a control means that causes an acquisition screen to be displayed ⋯.

The following is a machine translation provided by one of the engines.

> sys: A program ⋯ causing a computer to: store ⋯ a clear deck ⋯; determine ⋯ whether at least one clear deck ⋯ can be organized ⋯; identify ⋯, as specific content, content that is ⋯; and display ⋯ an acquisition screen ⋯.

The term "手段 (means)" is not reproduced in this translation. From a technical point of view, "causing a computer to function as a storage means that stores information" is equivalent to "causing a computer to store information". From the perspective of patent prosecution, some patent practitioners choose not to use the term "means" or any equivalent thereof (unit, portion, etc.) to avoid means-plus-function language (see, e.g., 35 U.S.C. 112(f)), a potential cause of rejection by a US examiner. The presence of the term "means" would probably not produce any benefit in patent prosecution in other IP offices where an application can be filed in English. Thus, since the use of the term "means" does not seem to add any value to this claim and may cause an issue in the US, it may be better to omit the term.

If omission, or addition in some cases, of certain terms or concepts can improve the quality of translation from the perspective of patent prosecution in the target country/region without distorting the content of the source text more than allowed, it should be considered an appropriate "adjustment".

The annotators marked the aforementioned omission of "means" as an error, namely "omission; major". From the reasons explained above, the omission may be beneficial. Although it may be possible to mark the omission, it should not be marked as a major error.

Other examples of appropriate adjustments are as follows:

- Addition/omission: "特徴とする (characterized)" is a good example of a term/concept that may be added or omitted according to the IP office the application is filed to.

- Inconsistency vs consistency: In Japan, translating a term into multiple equivalents is generally regarded as careless inconsistency. Outside Japan, in some cases, rendering a term into multiple terms in the target language can be beneficial. For instance, the applicant can let the examiner at some IP office choose a most suitable term for them to allow the claim.

9.3 Annotation Issues

As noted above, human annotations contained significant issues. The following are examples of numerous issues we found in the annotations, which were provided by one of Japan's most well-known patent translation companies.

Failure to detect errors

- "the first electronic device comprises a thermostat" was translated as "前記第１電子デバイスがサーモスタットである (the first electronic device is a thermostat)"
- "to the motion state estimation unit.., wherein"

Failure to detect relatively minor error

- Inconsistency between "operate" and "travel" as equivalents of "走行". "A vehicle traveling" in a segment of a road suggests any type of vehicle running through that segment. "A vehicle operating" in a segment of a road may suggest a more specific type of vehicle (e.g. truck) operating in that segment for a specific purpose (e.g. moving goods).

Failure to detect relatively major error

- See above discussion on "section during which".
- The source text states "characterized in that" in one place; the translation strongly suggests a different place for it.

Error detected by annotator is not an error

- Stating the subject matter of a claim twice, i.e. at the beginning and end of the claim, was marked as a major error. This is a common claim structure in Japanese patent applications.

Minor error detected should be relatively major error

- "said first electronic device being adapted to respond to user instructions by changing device state" was translated to mean "said first electronic device being adapted change device state in response to user instructions (この第1の電子装置はユーザ指示に応じて装置の状態を変化させる)". While the Japanese translation was marked as "awkward: minor" for some reason, the error is obviously a major error significantly distorting the meaning of the source text.

Major error detected should be relatively minor error (or no error)

- "A sensor, comprising" at the beginning of the English claim was rendered as "以下の構成要素からなるセンサー： (A sensor comprising the following constituents:)" at the beginning of the Japanese claim. Although it is not a common claim structure in Japan, a JPO examiner would probably accept it.

Human annotation may serve as training data for developing automatic annotation technology. Using erroneous annotations as training data will have negative consequences. If the annotations above, provided by a major translation company, represent a typical quality of human annotation in Japan, developing accurate automatic annotation technology in this country may encounter difficulties.

10 Conclusion and Future Perspective

This paper summarizes the first shared tasks of the patent claims translation. This year, we had 2 participants who submitted their translation results. Based on the human evaluation results, no system achieved consistently strong performance in any translation direction. However, comparisons with automatic evaluation results and analyses of human annotations revealed various issues, as reported in this paper.

In subsequent years, building on the insights obtained here, we aim to define a framework for more stable and higher-quality human evaluation, as well as to use the human annotations as training data to develop highly accurate automatic evaluation methods for patent translation.

Acknowledgments

The human evaluation fee is supported by Asia-Pacific Association for Machine Translation (AAMT) and AAMT/Japio[6] Special Interest Group on Patent Translation.

References

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

[6] Japan Patent Information Organization

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.

Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149, Online. Association for Computational Linguistics.

Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. 2024. JaParaPat: A large-scale Japanese-English parallel patent application corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9452–9462, Torino, Italia. ELRA and ICCL.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

A Appendix

A.1 Example of Development Data

Table 7 shows examples of development data. Each document may contain one or more claims. Each claim is basically composed of only one sentence, but it may contain line breaks for readability.

A.2 Human Evaluation Criteria

Table 8 and 9 shows the human evaluation criteria we used.

A.3 Extensive Discussion

The discussion in the body text on the use of generic or specific terms and differences in routines/legal restrictions between countries/IP offices will be presented below with more details. Note that some of the content below is a reproduction of Section 9.

Use of Generic or Specific Terms

> Source: "前記信頼度情報が予め設定された閾値よりも小さい状態が継続している区間を補正対象区間として検知して"
> "…前記運動状態推定部は、…前記補正対象区間を走行している前記他車両の運動状態を推定し"

A technically correct translation should be something along the lines of:

> TR: "··· detects, as a correction target section, a section in which the confidence information continues to remain below a preset threshold" "the motion-state estimation unit estimates the motion state of the other vehicle traveling through the correction target section"

Note that the discussion below focuses on the term "区間", which can be rendered into a number of terms including "section", "interval", "segment", "portion" or the like as long as it is clear that the term refers to a physical segment of a road, not to a time interval. From the second phrase above stating that the other vehicle travels through this section, it should be obvious that the section is not a time interval.

The following are the machine translations produced by the six engines.

> sys1: "··· detect, as a correction-target section, a section during which the reliability information remains less than the predetermined threshold" and "the motion state estimation unit is configured to estimate the motion state of the other vehicle traveling in the correction-target section"
> sys2: "··· detects, as a correction-target section, a section in which the state that the reliability information is less than the predetermined threshold continues" and "the motion state estimation unit is configured to estimate the motion state of the other vehicle traveling in the correction-target section"
> sys3: "··· detect, as a correction target section, a period during which reliability information ··· is smaller than a predetermined threshold and such a state continues longer than a predetermined time" and "the motion state estimation unit estimates the motion state of the other vehicles traveling in the correction target section"

Japanese
[請求項1] ポリエーテルポリオール（ａ１）と有機ポリイソシアネート（ａ２）を反応させて得られるイソシアネート基末端ウレタンプレポリマーを含有する主剤（A)、並びにポリオール（ｂ１)、導電剤（ｂ２)、及び制電剤（ｂ３）を含有する硬化剤（B）からなる半導電性ウレタンエラストマー形成性組成物において、 導電剤（ｂ２）が導電性カーボン、制電剤（ｂ３）が炭素数１０～２０の脂肪族系不飽和炭化水素基及び１個のヒドロキシアルキル基を有する第４級アンモニウムカチオンとビス（トリフルオロメタンスルホニルイミド）アニオンとからなるイオン性塩であって、導電剤（ｂ２）及び制電剤（ｂ３）の、主剤（A）と硬化剤（B）の総和量における各々の含有量が（ｂ２)：０．１～１質量%、(ｂ３)：０．００１～１０質量%であることを特徴とする半導電性ウレタンエラストマー形成性組成物。 [請求項2] ポリエーテルポリオール（ａ１）と有機ポリイソシアネート（ａ２）を反応させて得られるイソシアネート基末端ウレタンプレポリマーを含有する主剤（A)、並びにポリオール（ｂ１)、導電剤（ｂ２)、及び制電剤（ｂ３）を含有する硬化剤（B）からなる半導電性ウレタンエラストマー形成性組成物において、 導電剤（ｂ２）が導電性カーボン、制電剤（ｂ３）が炭素数１０～２０の脂肪族系不飽和炭化水素基及び２個のヒドロキシアルキル基を有する第４級アンモニウムカチオンとビス（トリフルオロメタンスルホニルイミド）アニオンとからなるイオン性塩であって、導電剤（ｂ２）及び制電剤（ｂ３）の、主剤（A）と硬化剤（B）の総和量における各々の含有量が（ｂ２)：０．１～１質量%、(ｂ３)：０．００１～１０質量%であることを特徴とする半導電性ウレタンエラストマー形成性組成物。

English
1. An aftermarket vehicle communication device engageable to a vehicle for providing location information associated with the vehicle to a V2X data stream, the device comprising: a housing configured to be detachably engageable to the vehicle; a GPS circuit disposable in communication with a GPS system to receive a GPS signal therefrom, the received GPS signal being representative of a location of the vehicle when the housing is engaged to the vehicle; and an antenna circuit coupled to the housing and in communication with the GPS circuit, the antenna circuit being configured to receive the GPS signal from the GPS circuit and communicate the GPS signal to the V2X data stream; a micro computing unit (MCU) coupled to the housing and in communication with the GPS circuit and the antenna circuit, the MCU being configured to generate an alert signal communicable to the V2X data stream via the antenna circuit, the alert signal being receivable by autonomous vehicles via the V2X data stream to facilitate assigning a prescribed margin of separation to the vehicle to which the housing is engaged; the GPS circuit and the antenna circuit being configured to facilitate both the receipt of the GPS signal from the GPS system and communication of the GPS signal to the V2X data stream independent of receiving information or data from the vehicle.

Table 7: Example of development data.

> sys4: "··· reliability information ··· is smaller than a preset threshold value and the state continues for a longer time than a preset time" and "the motion state estimation unit estimates a motion state of the other vehicle traveling in the correction target section"
> (Note: There was a significant omission in this translation and the term "区間" was not reproduced in the first clause.)
> sys5: "··· detects the interval during which the reliability information remains below the predetermined threshold as a correction target interval" and "the motion state estimation unit estimates the motion state of the other vehicle while it is traveling through the correction target interval"
> sys6: "··· detects, as a correction target section, a section in which reliability information ··· is smaller than a preset threshold value and the state continues for longer than a preset time" and "the motion state estimation unit estimates the motion state of the other vehicle traveling in the correction target section"

Both nouns "section" and "interval" on their own could be either a physical or temporal concept. In the above context, the preposition (plus relative pronoun), i.e. "during (which)" or "in (which)", is decisive in whether the preceding noun will be interpreted as a physical and/or temporal concept. In this context, the preposition "in" can be said to be a more generic preposition than "during". In other words, while "a section/interval during which" represents a temporal concept, "a section/interval in which" can represent both temporal and physical concepts. For the example above, it can be said that while "during" is incorrect, "in" is ambiguous (i.e. can be interpreted in more ways than one) yet potentially correct (i.e. encompasses the correct meaning). Although the latter clause "···前記運動状態推定部は、···前記補正対象区間を走行している前記他車両の運動状態を推定し (the motion-state estimation unit estimates the motion state of the other vehicle traveling through the correction target section)" provides enough information to determine if the "section" is a physical or temporal concept, such a determination cannot be made solely from the former phrase "前記信頼度情報が予め設定された閾値よりも小さい状態が継続している区間を補正対象区間として検知して (··· detects, as a correction target section, a section in which the confidence information continues to remain below a preset threshold)".

From the above, it is conceivable that if a set of information comprising one or more words is ambiguous and remains ambiguous even with reference to other information that are processed together with said information, it is better for the engine to choose one or more generic terms that keep the interpretation of the information open-ended. Moreover, if an engine is equipped with autocorrect function, it is also conceivable that the engine flags such ambiguous information while temporarily providing a generic term to it, then after processing other sections, refers back to it to examine if a more specific, context-suited term can be provided.

Let us examine the present case from the perspective patent rights. Even if the preposition "in" may render the first instance of "section" ambiguous, the latter clause will clarify the meaning of the term. Thus, an examiner, or a judge or opponent in a court case, will understand the meaning of the term and there will be no clarity-related rejection (35 U.S.C. 112 (b)) or dispute due to the ambiguity of the term. The meaning of a term or concept in a claim is often interpreted in view of the overall technical feature that is set forth by the claim as a whole. Choosing a specific term is preferable if the concept including the term is unambiguous, but if a concept is ambiguous, choosing a generic term may increase the chance of the concept being interpreted correctly.

More than one human annotators, who were exposed to the concept that the "section" is a segment of a road on which a vehicle travels, did not leave any annotation to the expression "a section during which". The expression "as a correction target section, a period during which" was marked with the annotation "inconsistency: major", but this probably refers to the inconsistency between "section" and "period", not to the semantic/technical inaccuracy.

The following are just a few examples of ambiguous terms that are often used in patent-related documents.

- 挟まれる (*hasamareru*): The term means an either physical or conceptual entity being interposed or held between one or more other physical or conceptual entities. A patty held by a bun, an interval between the first and second halves of a concert, Chomsky’s thoughts between Marks’s and Fodor’s, an insulator between and in contact with or with a gap to two layers, Jupiter in relation to Saturn and Mars or even in relation to Uranus and Earth, a river flowing between banks, or any such concept can be described using 挟まれる. The term is often translated as "sandwiched (between ···)", but obviously the expression can be misleading or incorrect in some context. In a context in which the specific manner of interposition can be, or intended to be, interpreted in more ways than one, a specific term such as "sandwiched" should be avoided.
(Needless to say, however, that a specific term such as "held (between)" should be chosen if 挟まれる focuses on the concept of an entity being physically held by other entity/entities. Inappropriate ambiguity may lead to abstract ideas, hence to clarity-related issues in patent prosecution or litigations.)

- (〜である) が ((*dearu*) *ga*): This is a highly context-sensitive particle and could mean "but/however", "and", "whereas", "yet", "so/thus", "in this regard/respect", etc. connecting the phrases before and after it to some degree and in some way. The particle is often used in office actions issued by the JPO in the context of, for example: "文献１には〜が記載されていないが、文献２には記載されている (Document 1 does not disclose ··· but document 2 does)"; "文献１はAAを記載しているが、文献2はBBを記載しており、両者を組み合わせることは容易である (Document 1 discloses AA, whereas document 2 discloses BB, and it would be easy to combine the two)"; or "本願はCCAと記載しているが、文献１はCCBと記載しており、文献１は本願発明を開示しているに等しい (The present application sets forth CCA. In this regard, document 1 discloses CCB and can be regarded as disclosing an equivalent of the invention of the present application)". Note that the generic concept "in this regard" may replace "but" and "and" in the first two example sentences. Moreover, it may be possible to entirely omit "が" and say "Document 1 does not disclose ···. Document 2 does”,“Document 1 discloses AA; document 2 discloses BB; it would be easy to combine the two", and "The present application sets forth CCA. Document 1 discloses CCB and can be regarded as disclosing an equivalent of the invention of the present application."

- 対象 (*taisho*): This is probably one of the most ambiguous yet convenient terms to refer to something that the writer of a text wants to refer to. The term could mean "target", "··· in question", "destination", "··· to be", "subject", "object", etc. In this research, an engine translated "補正対象区

間" as "correction target section". Although the translation is not erroneous, a more accurate and natural rendition would be "a section to be corrected" or "a segment subject to correction".

In some cases, it may be necessary to explicitly say what the writer wants to refer to by converting the term into a more specific concept. For instance, in an invention in which a tune is differentiated from the tune being analyzed and the analyzed tune is referred to as 対象楽曲 (literally, e.g. "the tune in question"), it may be better to refer to this tune as "the tune being analyzed". This is the case where use of a generic term does not work and it is better to use a more specific term, which may involve some additional/supplemental/complementary concepts.

Differences in Routines/Legal Restrictions between Countries/IP offices

Source: …プログラムであって、
コンピュータを、
…クリアデッキを記憶する記憶手段、
…一のクリアデッキを編成できるか否かを判定する判定手段、
…コンテンツを特定コンテンツとして特定する特定手段、
…取得画面を表示させる制御手段、
として機能させる、
プログラム。

A more or less literal/mirror translation would be something along the lines of:

A program ⋯, the program causing a computer to function as
⋯a storage means that stores a clear deck ⋯,
⋯a determination means that determines whether one clear deck can be organized ⋯,
⋯a specifying means that specifies, as specific content, content that is ⋯, and
⋯a control means that causes an acquisition screen to be displayed ⋯.

The following are the machine translations produced by the six engines.

sys1: A program ⋯ causing a computer to: store ⋯ a clear deck ⋯; determine ⋯ whether at least one clear deck ⋯ can be organized ⋯; identify⋯, as specific content, content that is ⋯; and display ⋯ an acquisition screen ⋯.

sys2: A non-transitory computer-readable medium storing instructions ⋯ causing a computer to: store ⋯ a clear deck ⋯; determine ⋯ whether one clear deck ⋯ can be organized; identify ⋯, as specific content, content ⋯; and cause an acquisition screen ⋯ to be displayed ⋯.

sys3: A program ⋯
causing a computer to function as:
a storage means for storing ⋯ a clear deck;
a determination means for determining ⋯ whether a clear deck ⋯ can be organized;
a specifying means for specifying ⋯, as specific contents, contents that are⋯; and
a control means for displaying ⋯ an acquisition screen ⋯.

sys4: Program ⋯, wherein a storage means for storing a clear deck ⋯ the computer ⋯; a determination means for determining whether or not one clear deck ⋯ can be organized ⋯; and, the control unit causes (the player) to function as: a specifying unit that specifies content ⋯ as specific content; and a control unit that causes ⋯ to display an acquisition screen ⋯.

sys5: A program ⋯ (comprising:)
a computer configured to function as:
a memory means for associating (each quest) with a cleared deck ⋯;
a judgment means for determining ⋯ whether their owned content is sufficient to assemble ⋯;
a specification means for identifying ⋯ the content items ⋯ as specified content items; and
a control means for displaying ⋯ an acquisition screen ⋯.

sys6: (Omission ⋯) storing a clear deck ⋯; determining whether or not a clear deck ⋯ can be organized ⋯; identifying, as specific content, content that is ⋯; and causing an acquisition screen ⋯ to be displayed ⋯.

In Japanese patent-drafting routines, it is common to repeat the subject matter of a claim at the end of the claim, as it can be seen in the above text where the term "プログラム (program)" appears at the beginning and the end of the claim. In view of how applications are drafted in English-speaking countries/regions, this repetition should not be reproduced in an English translation. In this regard, most of the engines seem to have managed to adopt a relatively correct sentence construction without such repetition.

Some engines (see Sys 1, Sys2, and Sys6) omitted the term "手段 (means)" from the translation. This omission may be evaluated from a technical point of view as well as from the perspective of patent prosecution. From a technical point of view, "causing a computer to function as a storage means that stores information" is equivalent to "causing a computer to store information". Both expressions mean that a computer having a memory is caused to store information in the memory. From the perspective of patent prosecution, some patent practitioners choose not to use the term "means" or any equivalent thereof (unit, portion, etc.) to avoid means-plus-function language (see, e.g., 35 U.S.C. 112(f)). Means-plus-function language may benefit the applicant under certain conditions but may also narrow the scope of the claim, especially in the US. In other countries and regions where it is possible to file an application in English, the omission of the term "means" would probably not result in any disadvantage for the applicant. So, for the current case, since the use of the term "means" does not seem to add any technical value to the claim, it may be better to omit the term at least in terms US drafting routines. From the above, it can be said that omission of certain terms or concepts, which may be called an appropriate "adjustment", may enhance the quality of translation from the perspective of patent prosecution. Similar adjustments can often be seen in more general writing. For example, a meaningful translation of the phrase "I am all ears" will be distant from a literal/mirror translation. Transition of a phrase from one sprachbund to another may require an appropriate adjustment. The value of a patent application is bound to the routines and legal restrictions exiting in the country/region in which the application is filed. When evaluating the quality of patent translation, the value of an appropriate adjustment should be taken into account in view of the routines and legal restrictions in the country/region to which the translation is destined to.

The annotators marked the aforementioned omission of "means" as an error in the form of "omission; major". From the reasons explained above, the omission may be beneficial, and although it may be possible to mark the omission as an error, the error should not be marked as "major".

Other examples of appropriate adjustments are as follows:

- Addition/omission: 特徴 (characteristic feature) is a good example of a term that may be added or omitted according to the IP office the application is filed to. The term means the characteristics of an invention that make the invention novel and inventive over prior art. Some IP offices may request that the characterizing potion (e.g. novel engine) of a claim be distinguished from the part of the claim adopting prior-art (e.g. any automobile) by using the term "特徴".

- Inconsistency vs consistency: As a general rule, a term used in a claim should be used consistently throughout the claim and in its dependent claims. In Japanese practice, a term that is used in the specification (e.g. 音響 (e.g. audio)) and that corresponds to the term in the claim (e.g. 音信号 (sound signal)) is often also used consistently throughout the specification. Translating 音響 into two or more terms (audio, acoustic, voice, etc.) may be considered careless inconsistency. However, 音響 encompasses a wide range of concepts and different examiners in certain IP offices may have different word choice preferences. Translating 音響 into different equivalents and amending the claim according to the examiner ’ s preferred word choice may render the prosecution smoother.

Top Category	Mid Category	Sub Category	Description	Example
Accuracy	Addition		Translation includes information that is not present in the source and that is not supposed to be included.	A translation includes portions of another translation that were inadvertently pasted into the document.
	Omission		Translation is missing content from the source and the omission is inappropriate.	A paragraph present in the source is missing in the translation.
	Untranslated text		Source text has been left untranslated.	A sentence in a Japanese document translated into English is left in Japanese.
	Mistranslation		Translaiton does not accurately represent the source.	A source text states that a medicine should not be administered in doses greater than 200 mg, but the translation states that it should be administered in doses greater than 200 mg (i.e., negation has been omitted).
	(Mistranslation)	Numerals / Symbols	Translation errors related to numerals and symbols.	3000 is translated as 30000
	(Mistranslation)	Article	Incorrect use of articles	A translation uses "a" for the item which appears for the second time.
	(Mistranslation)	Incorrect dependency	The adjective phrase or parallel structure has an incorrect dependency (please point out the correct dependency)	A of B, and C is translated as A of B and C (the dependency of C is incorrect)
	(Mistranslation)	Unknown dependency	The dependency structure of the source is not maintained.	said drive link being formed of one integral metallic piece = 駆動リンクにおいて、一体成形の金属片からなり
	(Mistranslation)	Ambiguity	The translation is more ambiguous than the source text (e.g. the source text can be interpreted in two ways, whereas the translation can be interpreted in three or more ways).	
Fluency	Punctuation		Incorrect punctuation (for locale or style, including improper sentence division, since patent claims must be written in one sentence).	1) An English text uses a semicolon where a comma should be used. 2) A two-digit year reference begins with an open single quote instead of a close single quote (apostrophe). 3) A Greek text uses a question mark instead of the anticipated semicolon to express a question. 4) German quotation marks are carried over into English or French target content.
	Spelling		Incorrect spelling or capitalization.	The German word Zustellung is spelled Zustetlugn.
	Grammar		Problems with grammar, other than orthography.	An English text reads “The man was seeing the his wife.
	Register		Wrong grammatical register (eg, inappropriately informal pronouns).	A formal letter uses contractions, colloquialisms, and expressions characteristic of spoken rather than written language, and those elements come across as less serious than intended.
	Inconsistency		Internal inconsistency (not related to terminology)	1) One part of a text is written in a clear, “terse” style, while other sections are written in a more wordy style. 2) The same text recurs at several points in a large document that has been divided up and submitted to multiple translators, with the result that that text is translated in three different ways, which can involve different style as well as terminology or register differences.
	Character encoding		Characters are garbled due to incorrect encoding.	A text document in UTF-8 encoding is opened as ISO Latin-1, resulting in all “upper ASCII” characters being garbled.

Table 8: Human Evaluation Criteria.

Top Category	Mid Category	Sub Category	Description	Example
Terminology	Inappropriate for context		Terminology is non-standard or does not fit context.	The word ’river’ in an English source text is translated into French as ’rivière’ . But the river in question flows into the sea, not into a lake or another river, so the correct French translation should have been ’ fleuve’.
	Inconsistent use		Terminology is used inconsistently.	The text refers to a component as the ’brake release lever’, ’brake disengagement lever’ , ’manual brake release’, and ’manual disengagement release’.
Style	Awkward		Translation has stylistic problems.	A text is written with many embedded clauses and an excessively wordy style. While the intended meaning can be understood, and the text is grammatically correct, the text is very awkward and difficult to follow. “ However, a personal language variety (in such approaches called “ idiolect ”) usually is internally heterogeneous (it varies in particular according to different situations and/or media) and therefore not suitable to serve as the smallest unit of linguistic variation, whereby in contrast, idiolects according to the framework developed in this document, are homogeneous by definition, whereas personal varieties are sets of idiolects. ”
Locale convention	Address format		Wrong format for addresses.	
	Currency format		Wrong format for currency.	
	Date format		Wrong format for dates.	
	Name format		Wrong format for names.	
	Telephone format		Wrong format for telephone numbers.	
	Time format		Wrong format for time expressions.	
Other			Any other issue.	
Source error			An error in the source.	
Non-translation			Impossible to reliably characterize distinct errors.	

Table 9: Human Evaluation Criteria (contd.).

Ehime-U System with Judge and Refinement, Specialized Prompting, and Few-shot for the Patent Claim Translation Task at WAT 2025

Taishi Edamatsu
Ehime University
edamatsu@ai.cs.ehime-u.ac.jp

Isao Goto
Ehime University
goto.isao.fn@ehime-u.ac.jp

Takashi Ninomiya
Ehime University
ninomiya.takashi.mk@ehime-u.ac.jp

Abstract

The Ehime University team participated in the Japanese-to-English Patent Claim Translation Task at WAT 2025. We experimented with (i) Judge and Refinement, (ii) Specialized Prompting, and (iii) Few-Shot Prompting. We used GPT-5 as the LLM. Evaluation based on the LLM-as-a-Judge framework confirmed improvements for (i), while (ii) and (iii) showed no significant effects. On the other hand, the official human evaluation indicated that the translation quality of method (i) decreased.

1 Introduction

In patent documents, patent claims represent a critically important section defining the scope of rights. Patent claims often consist of extremely long sentences with complex structures, making it difficult to translate them while maintaining correct legal interpretation. Additionally, selecting appropriate translations for patent-specific expressions and technical terminology presents challenges. The emergence of large language models (LLMs) in recent years has enabled machine translation to achieve results surpassing existing tasks. In the patent claim translation task, Azami et al. (2025) performed continued pre-training and fine-tuning of publicly available LLMs using parallel patent-translation data. However, human evaluation was not conducted for patent claim translations, leaving the challenges in patent claim translation unclear.

This paper describes the Ehime-U team's Japanese-to-English translation system for the WAT2025 patent claim translation task. We implemented three approaches in our LLM-based translation system. First, to address the issue that the challenges in patent claim machine translation have not been clearly identified, we introduce (i) Judge and Refinement based on the method of Chen et al. (2024) and (ii) Specialized Prompting . Furthermore, to improve terminology selection and consistency, we search training data for usage examples and employed them as (iii) Few-Shot training. We use GPT-5 as the base LLM. Evaluation using the LLM-as-a-Judge framework confirmed the effectiveness of (i) Judge and Refinement. However, (ii) Specialized Prompting and (iii) Few-Shot showed no discernible effect. On the other hand, the official human evaluation, which assessed only method (i), showed no improvement of method (i). This result indicates that the performance of the LLM-as-a-Judge framework was not sufficient in this case. Although the three methods evaluated in this study improved surface-level quality errors, we observed an increase in errors related to the fidelity of the original patent claims. This suggests that, when constraints are imposed through prompting, the LLM used in this work struggles to satisfy those constraints without degrading the overall fidelity of the content.

2 System Description

In this section, we describe the three techniques incorporated into our system: Judge and Refinement, Specialized Prompting, and Few-Shot.

2.1 Judge and Refinement

Judge and Refinement (Judge&Refinement) consists of three processing stages, and the procedure of each stage is described in order.

(1) Base Translation The Japanese patent claims are translated by an LLM on a per-claim basis while preserving line breaks. We defined a `PROMPT_POLICY` for the model as follows:

- Ensuring fidelity to the source text, including prohibiting additions, omissions, changes in legal meaning or legal scope, alterations of dependencies, and modifications of numerical values;

- Enforcing the distinction between independent claims, which are written without referencing preceding claims, and dependent claims, which must explicitly reference preceding claims;
- Standardizing punctuation;
- consistent antecedent references;
- complete preservation of numerical values, units, and formulas;
- consistent terminology across technical domains.

After that, We instructed the model to translate Japanese patent claims into U.S.-style English claims by using the policy as the persona of a professional patent-claim translator. In addition, we instructed the model not to add any annotations and to avoid any addition, omission, splitting, or merging of content. The detailed prompt is shown in Figure 1 in Appendix A. This method is called as Base Translation.

(2) Judge Using the source text and the generated translation, an LLM as a Judge evaluates the translation quality. The evaluation is conducted across the six criteria (Table 1), and an overall score (0–100 points) is calculated by averaging them equally. The detailed prompt is shown in Figure 3 in Appendix A.

(3) Refinement Without using any reference translations, the model is instructed to automatically extract and organize translation errors from the evaluation report, and then retranslate accordingly. From the LLM evaluation results obtained in (2), the model performs knowledge distillation to generalize the insights useful for refinement. Instead of focusing on specific errors (e.g., individual grammar or lexical mistakes), it abstracts recurring error patterns and systematic weaknesses into generalized categories, which serve as revision policies for refinement. Since the goal is to apply generic rather than case-specific corrections, all specific and unique information are removed, and each error is labeled according to one of the six categories used in (2). Common patterns within each category are then rewritten into rule-like sentences, typically following a two-part structure: "Symptom → Expected Form." For example: Symptom: "Range expressions use 'X–Y'." Expected Form: "Write 'X to Y' in ascending order." This design clarifies the purpose of the correction while avoiding semantic changes or redundant fixes.

we provide the extracted evaluation results, the Japanese source text, and the Base Translation as input to the LLM, expecting it to produce an English output with only minimal modifications. Here as well, we instructed the model to translate the text into U.S.-style English patent claims, in the same manner as in the Base Translation. The detailed prompt is shown in Figures 4 and 5 in Appendix A.

2.2 Specialized Prompting

In this section, we describe three methods for improving the translation prompts introduced in Section 2.1 to achieve translations that adhere more closely to U.S. claim conventions.

Specialized Base Translation

Instead of the simple instruction in Section 2.1 (1), "Translate into U.S. claim style," we adopt a strict audit-based translation prompt. The main revisions are as follows:

- **Pre-output audit (SILENT QA)**: The model self-verifies claim type, numbers/units, antecedents, and sentence structure before output.

- **Stronger output constraints**: Restriction to ASCII only, single-sentence structure, and enforcement of "colon + semicolon + `; and`" pattern.

- **Explicit prohibitions**: Elimination of "and/or," non-ASCII symbols, ambiguous pronouns, and unnecessary respectively.

- **Fixed terminology and style**: Explicit enforcement of standard phrases such as "apparatus," "configured to," and "equal to or greater than ...".

This enables the translator to function simultaneously as a self-auditing agent, ensuring both legal and structural consistency. The detailed prompt is shown in Figure 6 in Appendix B.

Select of Evaluation Results

The phase that extracts only the information necessary for refinement from the evaluation output is

Criterion	Description
fidelity_legal scope	Fidelity to legal scope and limitations
us_style structure	Conformity to the format and structure of U.S. patent claims
numbers_units ranges	Accuracy of numbers, units, ranges, and formulas
antecedent dependency	Consistency of antecedents and referential dependencies
terminology	Accuracy and consistency of terminology
naturalness	Naturalness and readability of expressions

Table 1: Evaluation criteria used in the LLM-as-a-Judge framework.

	Development Data	Test Data
Number of patents	13	26
Number of claims (sentences)	19	70

Table 2: number of claims

redesigned as a systematic error-category extraction prompt as follows:

- **Priority of extraction**: Fidelity > dependency > numbers/units > legal format.
- **Controlled output volume**: Limited to 10–15 representative issues, merging duplicates and superficial errors.
- **Unified output format**: Exampled as "Symptom > Expected Form" structure.
- **Noise filtering**: Extraction limited to essential issues that affect legal meaning.

This allows the system to identify the core issues to be fixed in refinement using the evaluation results. The detailed prompt is shown in Figure 7 Appendix B.

Refinement
In the refinement phase, a minimal-edit policy is introduced to suppress overcorrection.

- **Two-layered objective**: (1) Maximize semantic and legal consistency, (2) Preserve n-grams for minimal editing (BLEU retention).
- **Limited edit scope**: Revise only the portions listed in "Issues to fix."
- **Format revalidation**: Re-enforce the U.S. claim structure (colon, semicolon, "; and", single-sentence rule).
- **Local correction policy**: Prohibit any rephrasing beyond essential grammatical corrections.

Through this approach, refinement is defined not as a full rewrite but as a localized legal correction phase. The detailed prompt is shown in Figure 8 in Appendix B.

2.3 Few-shot Prompting

We extend the method described in Section 2.1 by incorporating a few-shot mechanism (Brown et al., 2020) using translation examples based on FAISS (Douze et al., 2024) and SentenceTransformer (Reimers and Gurevych, 2019). FAISS is a high-speed library for vector similarity search, designed to efficiently retrieve "similar vectors" from large-scale vector datasets. When constructing the FAISS index, we use bilingual Japanese–English sentence pairs from the Patent Cooperation Treaty (PCT) route portion of the JaParaPat (Nagata et al., 2024) corpus, which is the training data for this task. Under the PCT route, a single international patent application is submitted to multiple national offices through translation, making the resulting multilingual publications effectively parallel. Because these pairs represent direct translations of the same application, they can be regarded as highly reliable parallel data.

The Japanese claim sentences are embedded using the multilingual sentence embedding model (intfloat/multilingual-e5-base), enabling the system to evaluate semantic similarity between sentences based on cosine similarity. Consequently, for a given input claim, semantically similar Japanese–English pairs can be efficiently retrieved and utilized as reference examples in few-shot translation. Few-shot prompting is applied to the

System	LLM as a judge score (%)
Base Translation	91
Base Translation + Judge&Refinement	**92**
Specialized Prompting	80
Specialized Prompting + Judge&Refinement	84
Few-shot (sentence)	84
Few-shot (sentence) + Judge&Refinement	84
Few-shot (sentence) + Specialized Prompting	78
Few-shot (sentence) + Specialized Prompting + Judge&Refinement	79
Few-shot (term)	78
Few-shot (term) + Judge&Refinement	83
Few-shot (term) + Specialized Prompting	82
Few-shot (term) + Specialized Prompting + Judge&Refinement	83

Table 3: Evaluation results based on LLM as a judge for the test data

System	COMET	BLEU
Base Translation	84.59	53.81
Base Translation + Judge&Refinement	84.95	48.89
Specialized Prompting	85.35	**56.55**
Specialized Prompting + Judge&Refinement	**85.48**	55.64
Few-shot (sentence)	84.45	50.09
Few-shot (sentence) + Judge&Refinement	84.67	51.43
Few-shot (sentence) + Specialized Prompting	85.14	53.88
Few-shot (sentence) + Specialized Prompting + Judge&Refinement	85.08	52.43
Few-shot (term)	85.01	53.54
Few-shot (term) + Judge&Refinement	84.97	51.92
Few-shot (term) + Specialized Prompting	85.16	52.92
Few-shot (term) + Specialized Prompting + Judge&Refinement	85.15	52.21

Table 4: Evaluation results based on COMET and BLEU for the development data

translation and refinement stages.

Two types of few-shot examples are used in this study:

Sentence-Level Example The first method performs cosine similarity search against the FAISS index built from full-sentence vectors. The top three most similar Japanese–English pairs are retrieved and inserted into the translation prompt as few-shot (sentence) examples.

Term-Level Example To retrieve translation examples including important terms in the source sentence, the second method uses the LLM to extract three terms from the input sentence and uses them as queries. These queries are used for FAISS retrieval, and the retrieved bilingual sentence pairs are organized into a few-shot sentence. This method is referred to as few-shot (word) in the following evaluation.

Additionally, both of these few-shot methods are combined with the specialized prompt described in Section 2.2 for comparative evaluation. The detailed prompt is shown in Figure 9 in Appendix C.

2.4 LLM

In this system, we use OpenAI's GPT-5[1] as the underlying LLM.

2.5 Dataset

We use the official development and test data provided for the WAT 2025 "Patent Claims Translation / Evaluation Tasks". The development data

[1] https://platform.openai.com/docs/models/gpt-5

consist of source-language patent claims and their corresponding translations, whereas the test data contain only the source-language patent claims. We show the number of patent and patent claims for each data point in the Table 2. In addition, we use data from JaParaPat for the Few-Shot Prompting. JaParaPat is a large-scale Japanese–English parallel corpus aligned between Japanese and English patent application documents. It consists of approximately 107 million Japanese–English sentence pairs automatically extracted from patent document families filed between 2016 and 2020, and includes metadata such as application-type labels and document IDs. From this corpus, we used only the sentence pairs whose document IDs correspond to claim sections.

3 Evaluation

3.1 Our Evaluation

Patent claim translation involves very long and syntactically complex sentences, making it difficult to fully understand the structure of each claim. Furthermore, accuracy must be preserved across multiple dimensions—not only in meaning but also in legal scope and technical terminology—thus, existing automatic evaluation methods struggle to precisely assess translation adequacy. In contrast, LLM-as-a-Judge, which evaluates translations using an LLM, is expected to consistently assess the appropriateness of translations across all parts of a long sentence. Therefore, we employ the LLM-as-a-Judge as our primary evaluation method. The evaluation criteria are the same as those defined in Section 2.1 (2).

For evaluation, we use the test dataset described in Section 2.5, which does not include reference translations. The results are shown in Table 3. The Judge&Refinement configuration achieved a higher score than the Base Translation. On the other hand, the Specialized Prompting score was lower than the baseline, and both Few-shot (sentence) and Few-shot (term) also showed lower scores than the baseline. Therefore, the effectiveness of these few-shot and specialized prompting methods was not confirmed.

For reference, Table 4 presents the results of automatic evaluation using the COMET (wmt22-comet-a; Rei et al., 2022) and BLEU (sacrebleu; Post, 2018) metrics. These scores were calculated using the development dataset, which includes reference translations, instead of the test dataset.

3.2 Official Evaluation

As the official evaluation for WAT 2025, the task organizers conducted human assessment. Manual error annotations and evaluation scores were assigned to each source sentence and its translated output by human evaluators. Error annotations were assigned to problematic segments based on error categories such as mistranslation, omission, and hallucination, with each error being labeled for severity (major or minor). In addition, an official score out of 100 points was assigned to each sentence. After assigning evaluation priorities to the translation results of the test data and submitting all results shown in Table 2, two systems—Base Translation and Judge&Refinement—were evaluated by the organizers. For each system, the 28 sentences out of the 70 test sentences were evaluated by humans. The official human evaluation results for error categories and average scores are presented in Tables 5 and 6. Compared with the Base Translation, the number of major errors in the Refinement output increased from 18 to 42, and the number of minor errors increased from 119 to 150. Therefore, the total number of errors increased from 137 to 192. In addition, the average score decreased from 86.07 for Base Translation to 81.60 for Judge&Refinement.

4 Analysis

4.1 Analysis Based on the Official Evaluation

We analyze the reasons why Judge&Refinement received lower evaluation scores than Base Translation While surface-level errors—such as grammatical errors and punctuation issues involving the use of commas and semicolons—were improved, no improvements were observed for other types of errors. In particular, substantial increases were observed in hallucination, omission, and mistranslation errors, indicating a rise in errors related to the fidelity of the original patent claims. However, many of the mistranslation errors were attributable to article-related issues, such as incorrect selection of "a" or "the" and omitted articles. When these article errors are excluded, the number of remaining mistranslations becomes much closer, with 11 for Base Translation and 13 for Judge&Refinement. Although the change in the number of these errors was not large, many errors related to terminology consistency and contextual inappropriateness were also observed.

	Base Translation		Judge&Refinement	
Error Category	Major	Minor	Major	Minor
Omission	8	13	24	14
Terminology Consistency	1	33	0	36
Grammar	1	8	2	1
Mistranslation	5	18	7	25
Other	0	2	0	5
Contextually Inappropriate	3	11	2	14
Hallucination	0	10	5	34
Source Text Error	0	3	1	2
Punctuation	0	17	0	8
Lack of Consistency	0	0	0	4
Awkward Expression	0	4	0	5
Article Error	0	0	1	2
Total Errors	18	119	42	150
Total (Major+Minor)	137		192	

Table 5: Official human evaluation results (number of error categories).

	Base translation	Judge&Refinement
Average score	86.07	81.60

Table 6: Official human evaluation results (average score)

In the Judge&Refinement method, the initial translation is evaluated using the LLM-as-a-Judge framework, and the output is refined based on the abstract error types extracted from the evaluation report. In addition to the strict U.S.-style constraints defined in the PROMPT_POLICY used for the Base Translation, the model is explicitly instructed to revise the English text in accordance with the identified issues. As a result, while surface-level improvements were made—such as corrections to grammar and punctuation, better adherence to U.S. claim style, and the introduction of common expressions used in patent translation—we consider that there was also an increase in errors related to loss of fidelity to the original text, including incorrect scope or comparison direction, erroneous antecedent references (mistranslations), the addition of elements not present in the source (hallucination), and the omission of obligatory elements (omission). In particular, the refinement step appears to prioritize producing well-formed English over maintaining strict fidelity to the source text, as it tends to rewrite the entire sentence rather than apply minimal edits.

A comparison between the official evaluation and the LLM-as-a-Judge evaluation shows that, although the score for Judge&Refinement improved under the LLM-as-a-Judge framework, its translation quality deteriorated in the human evaluation. Therefore, it was confirmed that the performance of the LLM-as-a-Judge framework was not sufficient in this study.

4.2 Analysis of Results Not Assigned Official Evaluation

For Specialized Prompting and Few-Shot Prompting, we conducted our evaluation using the LLM-as-a-Judge framework. Compared with Judge&Refinement, Specialized Prompting and Few-Shot Prompting improved consistency with U.S. patent-claim style, the naturalness of the English output, and the stability of terminology and unit expressions. As a result, their scores for us_style_structure and naturalness in Table 1 increased. However, incorrect modifications of claim scope and the insertion of erroneous dependency relations led to decreases in fidelity_legal_scope and antecedent_dependency scores. In Specialized Prompting, the model is strongly biased toward producing “natural English” and adhering to “U.S. claim style” whereas essential aspects of patent translation—such as structural preservation and legal fidelity—tend to degrade. We consider that this imbalance led to lower LLM-evaluation scores. Similarly, Few-Shot Prompting showed improvements in stylistic aspects of the translation, including punctuation placement, element enumeration, lexical consistency such as the use of “configured to,” and stabilization of U.S.-claim-specific sentence patterns. However, while Few-Shot Prompting improves stylistic consistency and terminology, we consider that it is strongly influenced by the structural bias of the retrieved examples, causing structural distortions in the translated output—such as reorganization of elements, shifts in clause positions, and unnecessary insertions of wherein. These issues likely resulted in substantial score reductions in the fidelity and antecedent_dependency categories of Table 1. We consider that the performance of Few-Shot Prompting declined relative to Specialized Prompting because the model was heavily influenced by the complexity of the retrieved examples. This influence led to several structural distortions, such as subtle alterations of numerical and range expressions, shifts in the positions or antecedents of modifiers and conditional clauses, the splitting of a single original element into multiple parallel components, and the unnecessary insertion of wherein clauses. Since these distortions are treated as structural deviations from the source text in the evaluation, substantial penalties were applied to the fidelity and antecedent dependency categories.

Based on our analysis, when constraints are imposed on the LLM through prompting, it is difficult for the model used in this study to satisfy those constraints without reducing the overall fidelity of the content, indicating that this remains an important challenge for future work.

5 Conclusion

For patent claim translation using LLMs, we explored three different approaches. Among them, Judge and Refinement successfully improved the evaluation scores under the LLM-as-a-Judge framework. the other two approaches—Specialized Prompting and Few-shot did not show any improvement in the LLM-as-a-Judge evaluation. In the official human evaluation, the comparison between Judge and Refinement and Base Translation showed that the total number of errors increased, and the average score dropped from 86.07 for Base Translation to 81.60 for Judge and Refinement, confirming that human-evaluated quality declined. This result indicates that the performance of the LLM-as-a-Judge framework was not sufficient in this study. The analysis showed that although the three methods improved surface-level quality errors, they also led to an increase in errors related to the fidelity of the original patent claims. When constraints are imposed on the LLM through prompting, it is difficult for the model used in this study to satisfy those constraints without reducing the overall fidelity of the content, making this an important challenge for future work.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP24K15071. These research results were obtained from the commissioned research (No.22501) by National Institute of Information and Communications Technology (NICT), Japan.

References

Haruto Azami, Minato Kondo, Takehito Utsuro, and Masaaki Nagata. 2025. Patent claim translation via continual pre-training of large language models with parallel data. In *Proceedings of Machine Translation Summit XX: Volume 1*, pages 300–314, Geneva, Switzerland. European Association for Machine Translation.

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. *Preprint*, arXiv:2005.14165.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2024. Iterative translation refinement with large language models. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 181–190, Sheffield, UK. European Association for Machine Translation (EAMT).

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Masaaki Nagata, Makoto Morishita, Katsuki Chousa, and Norihito Yasuda. 2024. JaParaPat: A large-scale Japanese-English parallel patent application corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9452–9462, Torino, Italia. ELRA and ICCL.

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, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

A Prompt for Judge and Refinement

```
You are a professional Japanese→English translator for patent claims.
Follow the full policy below strictly. Translate the single input line into exactly one line of
US-style claim English.
Return only the translation (no notes). Do not add/omit/split/merge content.
```

Figure 1: System Prompt for Base Translation

```
Translate this single Japanese claim line into English:
```

Figure 2: User Prompt for Base Translation

```
You are a meticulous patent-claim reviewer. Evaluate an English translation (PRED) against the
original Japanese claim (JA) with NO reference translation.

Rubric categories (each 0-100; all categories have equal weight):
- fidelity_legal_scope
- us_style_structure
- numbers_units_ranges
- antecedent_dependency
- terminology
- naturalness

Output MUST be valid GitHub-flavored Markdown with the following sections:

Findings (>= min_findings items)
- Bullet list of concrete issues or confirmations (legal style, fidelity,
numbers/units/formulas/ranges, terminology consistency, antecedent basis, dependency,
punctuation/format, naturalness).
- Each item starts with a label like [Fidelity], [Numbers/Units], etc., and quotes the exact
snippet(s) from JA/PRED.

Fix Suggestions
- Bullet list mapping to the Findings, each with a minimally-edited corrected English fragment.
- Preserve scope; do not introduce new elements.

One-line Verdict
- One sentence stating whether the PRED is acceptable for filing.

Subscores
Provide a JSON code block with EXACT keys and integer values 0-100:
'''json
{{ "fidelity_legal_scope": 0,
"us_style_structure": 0,
"numbers_units_ranges": 0,
"antecedent_dependency": 0,
"terminology": 0,
"naturalness": 0
}}'''
```

Figure 3: Prompt for Judge

```
あなたは翻訳品質監査官です。入力は“日本語＋翻訳のみで作成された評価レポート”。
この文書から参照文に依存しない抽象的な『問題タイプ/再発しやすい症状』だけを抽出し、
箇条書き（10〜15項目）で**具体的**に要約してください（『症状 → 期待される形』）。
```

Figure 4: Prompt for Error-pattern Distillation

```
You are a professional patent-claims translator performing a second pass.
Obey the full policy below. You do NOT see any reference translation.
Given the Japanese line, your first-pass English line, and abstract issue types, fix the English
**without adding/removing content**.
Return exactly one English line in US claim style.
```

Figure 5: Prompt for Refinement

B Prompt for Specialized Prompting

```
You are a professional Japanese→English translator specialized in **US-style patent claims**.
Translate **each input line** into **exactly one English claim line**.
Output: **ONLY** the final English claim line (no notes, no bullets, no brackets, no extra spaces).

HARD CONSTRAINTS (must all hold):
- **One sentence** per claim; period at the end; ASCII-only characters.
- US claim formatting: colon after the preamble; **semicolons** between parallel elements; **'; and'** before the last element.
- **Do not add/omit/reorder** content; preserve all numbers, units, symbols, ranges ("X to Y"), inequalities (<=, >=, <, >), equations, and dependencies.
- Maintain claim category (apparatus/method/etc.), numbering, and antecedent basis (first mention "a/an/at least one [X]" → thereafter "the [X]"; keep singular/plural consistent).
- For dependent claims: "The [subject] according to claim X (or X or Y/any one of claims X to Y), wherein . . . ." No new elements introduced in dependents.
- Forbidden: "and/or", non-ASCII dashes (—–〜), ambiguous pronouns without antecedent, "respectively" unless explicitly warranted by the JP text.

SILENT QA (do internally and **do not print** the checks):
1) **Category map**: identify claim type; keep it unchanged.
2) **Numbers/units audit**: list every value/unit/range/inequality/equation and verify 1:1 preservation; replace wave dashes with "to"; add a space between number and unit (10 mm); % is attached (10%).
3) **Antecedent map**: ensure every "the [X]" has a prior "a/an [X]" (or "first/second [X]").
4) **Format skeleton**: preamble + colon; element list with semicolons; insert "; and" before the last element; final period.
5) **Terminology lock**: prefer "apparatus" (when appropriate), "configured to", "equal to or greater than/less than or equal to", "idle channel", "suction air temperature", "thermo-OFF/ON", etc., as aligned with the policy below.
```

Figure 6: Prompt for Specialized Base Translation

```
あなたは翻訳品質の監査官です。入力は JA と PRED のみで作られた評価レポートです。
人手評価の得点改善に**直結**する 10〜15 件の「問題タイプ（症状→期待形）」を、重複をまとめて一般
化して抽出してください。
優先順位:
1) 忠実性の逸脱（主語/述語/条件/比較/包含/選択/否定/数量/因果）
2) 係り受け・依存関係（antecedent、wherein の接続、要素導入/再登場の不整合）
3) 数値・単位・範囲・不等号・式（ASCII/順序/包含条件/桁区切り/単位スペース）
4) 法的フォーマット（コロン/セミコロン/"; and"/一文制/終止）
※ 表層の言い換えのみは除外。意味/法的効果に影響する項目を優先。

出力形式（例）：
- 【範囲表現】"A〜B" を "A to B" に統一。境界の≦/≧は JA に忠実。
- 【antecedent】"the X" には先行 "a/an/first X" を必須化。再登場での冠詞逸脱を是正。
- 【列挙体裁】パラレル要素はセミコロン列挙＋最後に "; and"。. . .
```

Figure 7: Prompt for Specialized Error-pattern Distillation

```
You are a senior patent-claims translator performing a **targeted second pass** with **no
reference translation**.
Objectives (in this order):
1) **Semantic adequacy  legal correctness** (maximize human adequacy judgment).
2) **Minimal-edit policy** to preserve n-grams/phrases of the first-pass English **outside the
problematic spans** (helps BLEU and perceived consistency).

HARD CONSTRAINTS:
- Do NOT add/remove meaning vs. Japanese; preserve all numbers, units, symbols, ranges ("X to Y"),
inequalities, equations, dependencies, and claim category.
- Enforce US claim style: one sentence; colon after preamble; semicolons between parallel elements;
"; and" before the last element; final period; ASCII-only.
- Maintain antecedent basis; attach "wherein" to the correct antecedent; do not introduce new
elements in dependent claims.
- **Edit only spans implicated by the "Issues to fix" section**; elsewhere keep tokens identical
to the first-pass output unless grammar requires a local micro-fix.
- Avoid "and/or" and non-ASCII dashes; keep spacing for numbers/units; keep thousands separators;
"μ"→"um".

SILENT QA BEFORE OUTPUT (do not print): numbers/units audit, antecedent map, format skeleton, and
dependency sanity check.
```

Figure 8: Prompt for Specialized Refinement

C Prompt for Few-shot Prompting

```
"あなたは日本語特許請求項の専門家です。以下の日本語1行（1クレーム）について、"
"FAISSで高精度に用例を拾うための『日本語クエリ』を**ちょうど3件**、JSON配列で出力してください。"
"各クエリは（A）中核技術語（専門語・化学名・機械要素・電気回路名など）、（B）構成要素/機能語（～
部、～手段、～回路、configured to 等）、"
"（C）決定的な制約（wherein条件、数値レンジ、不等号、単位、選択肢列挙、依存関係）を**過不足な
く**含めてください。"
"一般語（装置、処理、データ等）や曖昧語を避け、品詞は名詞句中心で**8～24文字程度**に収めます。"
"括弧・全角記号・機種依存文字は使用しません。"
"出力は**厳密に**次のフォーマットのみ："
"...",
"...",
"..."


"説明や余計な文字は一切付けないでください。"
```

Figure 9: Prompt for generating FAISS search queries

Source Sentence
[請求項2] 前記第1推定モデルは、前記第1ユーザ群のそれぞれのユーザの特徴量と前記第1テーブルデータを構成するそれぞれの項目の特徴量とから前記第1ユーザ群のそれぞれのユーザに対応する前記それぞれの項目の値を推定し、 前記第2推定モデルは、前記共通ユーザ群のそれぞれのユーザの特徴量と前記第2テーブルデータを構成するそれぞれの項目の特徴量とから前記共通ユーザ群のそれぞれのユーザに対応する前記それぞれの項目の値を推定し、 前記推定部は、前記第1推定モデルで利用される前記第1ユーザ群のそれぞれのユーザの特徴量と前記第2推定モデルで利用される前記第2テーブルデータを構成するそれぞれの項目の特徴量とに基づいて、前記共通ユーザ群を除く前記第1ユーザ群についての前記第2データの値を推定する、請求項1に記載の推定装置。
Base Translation
2. The estimation apparatus according to claim 1, wherein: the first estimation model is configured to estimate values of respective items corresponding to each user of the first user group based on feature quantities of each user of the first user group and feature quantities of the respective items constituting the first table data; the second estimation model is configured to estimate values of respective items corresponding to each user of the common user group based on feature quantities of each user of the common user group and feature quantities of the respective items constituting the second table data; and the estimation unit is configured to estimate values of the second data for the first user group excluding the common user group, based on the feature quantities of each user of the first user group used by the first estimation model and the feature quantities of the respective items constituting the second table data used by the second estimation model.
Refinement
2. The estimation apparatus according to claim 1, wherein: the first estimation model is configured to estimate values of the respective items corresponding to each user of the first user group based on features of each user of the first user group and features of the respective items that constitute the first table data; the second estimation model is configured to estimate values of the respective items corresponding to each user of the common user group based on features of each user of the common user group and features of the respective items that constitute the second table data; and the estimation unit is configured to estimate values of the second data for users of the first user group excluding users of the common user group based on (i) the features of each user of the first user group that are used by the first estimation model and (ii) the features of the respective items that constitute the second table data and that are used by the second estimation model.

Table 7: Example Output of Judge and Refinement
Examples in which, although punctuation errors were reduced, hallucinations and omissions occurred.

Source Sentence

[請求項1] 水面下で軸線が前後方向に延びるように配置され、前側を上流側とするとともに後側を下流側とする流路を形成する筒部と、
前記筒部の内部に配置され、前記軸線方向に延びる軸部と、
前記軸部に装着され、前記流路内で前記軸線の径方向に延びるとともに前記軸線の周方向に配列された複数のプロペラ翼を有し、前記軸線回りに回転可能なプロペラと、
前記流路内で前記プロペラに対して前側及び後側の少なくともいずれか一方に設けられ、前記径方向に延びるとともに前記周方向に配列され、前記軸部を支持する複数のストラットと、
を備え、
前記複数のストラットは、上下方向に延びて前記軸線を通る対称線に対して左右対称に配置され、
前記ストラット同士の角度間隔のうち少なくとも1つの角度間隔は、他の前記ストラット同士の角度間隔と異なり、
前記軸線を通るように水平方向に延在する水平面に対して、上下方向両側に少なくとも1つの前記ストラットが配置されている、推進装置。

Base Translation

1. A propulsion apparatus comprising: a tubular portion arranged below a water surface such that an axis extends in a fore-aft direction, the tubular portion forming a flow path in which a front side is an upstream side and a rear side is a downstream side; a shaft portion disposed inside the tubular portion and extending in an axial direction; a propeller mounted to the shaft portion, the propeller having a plurality of propeller blades that extend in a radial direction of the axis and are arranged in a circumferential direction of the axis, the propeller being rotatable about the axis; and a plurality of struts provided in the flow path on at least one of a front side and a rear side with respect to the propeller, the plurality of struts extending in the radial direction and being arranged in the circumferential direction, the plurality of struts supporting the shaft portion; wherein: (i) the plurality of struts extend in a vertical direction and are disposed left-right symmetrically with respect to a symmetry line passing through the axis; (ii) at least one angular interval among angular intervals between the struts is different from other angular intervals between the struts; and (iii) with respect to a horizontal plane extending horizontally so as to pass through the axis, at least one of the struts is disposed on each of both sides in a vertical direction.

Refinement

1. A propulsion apparatus comprising: a tubular portion arranged below a water surface such that an axis of the tubular portion extends in a fore-aft direction, the tubular portion forming a flow path in which a front side is an upstream side and a rear side is a downstream side; a shaft portion disposed inside the tubular portion and extending in an axial direction; a propeller mounted to the shaft portion, the propeller having a plurality of propeller blades that extend in a radial direction of the axis and are arranged in a circumferential direction of the axis, the propeller being rotatable about the axis; and a plurality of struts provided in the flow path on at least one of a front side and a rear side of the propeller, the plurality of struts extending in the radial direction and being arranged in the circumferential direction, the plurality of struts being configured to support the shaft portion; wherein: (i) the plurality of struts extend in a vertical direction and are disposed left-right symmetrically with respect to a line of symmetry that passes through the axis; (ii) at least one angular interval between the struts differs from the other angular intervals between the struts; and (iii) with respect to a horizontal plane that passes through the axis, at least one of the struts is disposed on each of an upper side and a lower side.

Table 8: Example Output of Judge and Refinement
Examples of increased hallucinations and omissions.

UTSK25 at WAT2025 Patent Claims Translation/Evaluation Task

Haruto Azami, Zhang Yin, Futo Kajita, Nobuyori Nishimura, Takehito Utsuro
Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba

Abstract

This paper presents the submission of UTSK25 for the English–Japanese and Japanese–English at the WAT2025 Patent Claims Translation/Evaluation Task. We use a single translation model for both translation directions, built from a large language model through monolingual and bilingual continual pretraining and bilingual supervised fine-tuning. We finally generate translations via prompt engineering to reduce omissions and hallucinations.

1 Introduction

We describe our UTSK25 translation system for the WAT2025 English–Japanese (En–Ja) and Japanese–English (Ja–En) Patent Claims Translation/Evaluation Task. Our translation model is trained on a pretraining large language model (LLM), rinna/llama-3-youko-8b[1]. We combine two training stages (Kondo et al., 2024; Azami et al., 2025) to train a single model for both directions: continual pretraining (CPT) (Ke et al., 2023) and supervised fine-tuning (SFT) (Zhang et al., 2024). After training the single translation model, we generate translations with prompt engineering techniques designed to mitigate omissions and hallucinations. The following sections show the details of our system.

2 Approaches

2.1 Training

Continual pretraining Continual pretraining (CPT) extends the training of LLMs by further optimizing the causal language modeling objective on new monolingual corpora (Ke et al., 2023). The goal is to optimize the model parameters θ by minimizing the negative log-likelihood $\mathcal{L}_{\text{CPT}}$ over a corpus $\mathcal{D}_{\text{CPT}}$. Given a corpus $\mathcal{D}_{\text{CPT}} := \{\mathbf{y}_i\}_{i=1}^{|\mathcal{D}_{\text{CPT}}|}$ composed of token sequences $\mathbf{y} = (y_1, \ldots, y_{|\mathbf{y}|})$ from the vocabulary $\mathcal{V}$ (where $\mathbf{y} \in \mathcal{V}^*$), the loss is defined as:

$$\underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{y} \in \mathcal{D}_{\text{CPT}}} \mathcal{L}_{\text{CPT}}(\mathbf{y}; \theta), \tag{1}$$

$$\mathcal{L}_{\text{CPT}}(\mathbf{y}; \theta) := -\sum_{t=1}^{|\mathbf{y}|} \log p_\theta(y_t | \mathbf{y}_{<t}). \tag{2}$$

This objective trains the model to predict the next token y_t given its history $\mathbf{y}_{<t}$. For efficiency, practical implementations often limit the context to a fixed-size window c, using $\mathbf{y}_{[t-c,t)} := (y_{t-c}, \ldots y_{t-1})$ as the condition instead of the full sequence $\mathbf{y}_{<t}$. This formulation is identical to the standard pretraining objective for causal LMs.

Supervised fine-tuning Supervised fine-tuning (SFT) optimizes pretrained model parameters θ for downstream tasks using a labeled dataset (Zhang et al., 2024). This dataset, $\mathcal{D}_{\text{SFT}} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}_{\text{SFT}}|} \subset \mathcal{V}^* \times \mathcal{V}^*$, contains pairs of an input $\mathbf{x}$ and its corresponding ground-truth output $\mathbf{y}$. The optimization objective is to minimize the negative log-likelihood $\mathcal{L}_{\text{SFT}}$ over all pairs in $\mathcal{D}_{\text{SFT}}$:

$$\underset{\theta}{\operatorname{argmin}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_{\text{SFT}}} \mathcal{L}_{\text{SFT}}(\mathbf{x}, \mathbf{y}; \theta), \tag{3}$$

$$\mathcal{L}_{\text{SFT}}(\mathbf{x}, \mathbf{y}; \theta) := -\log p_\theta(\mathbf{y} \mid \mathbf{x}). \tag{4}$$

This process steers the model to generate outputs conditioned on the input that are consistent with the human-annotated targets.

2.2 prompt engineering

We generate translations with prompt engineering techniques designed to mitigate omissions and hallucinations only for the En–Ja translation.

3 Submission System

We train the En–Ja and Ja–En single translation model from a pretrained LLM, llama3-youko-8b.

[1] https://huggingface.co/rinna/llama-3-youko-8b

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 30–34
December 24, 2025 ©2025 Association for Computational Linguistics

Submission	**En–Ja Prompt Used**
System 1 (Primary)	Prompt 2
System 2 (Not Primary)	Prompt 3
System 3 (Not Primary)	Prompt 1

Table 1: Submitted systems. All systems use the identical bilingual (En–Ja/Ja–En) model trained with CPT and SFT, differing only in the prompt used for the En–Ja direction.

According to our preliminary experiments and subjective judgment, we selected the combinations of training methods and prompts.

We show the system overview in Table 1.

3.1 Continual Pretraining

We perform bilingual CPT for our translation model. For CPT, we use a subset of the JParaPat dataset (Nagata et al., 2025).Table 3 summarizes the data statistics for CPT.

We filter this subset to remove entries where the English side contains "(canceled.)". The CPT corpus is balanced, containing 50% English-to-Japanese (En–Ja) and 50% Japanese-to-English (Ja–En) examples.

The CPT hyperparameters are listed in Table 2.

3.2 Supervised Fine-tuning

Following CPT, we conduct supervised fine-tuning (SFT). For SFT, we use the 2020 patent claims data from JParaPat (Nagata et al., 2025).

While the original dataset consists of line-by-line parallel data, some patent claims span multiple lines. To address this, we first construct claim-level pairs by segmenting the Japanese text at kuten (。) and the English text at periods (.). We also filter out pairs containing "(canceled.)" on the English side, similar to the CPT data preparation.

From this processed dataset, we then sample our final training data, selecting only pairs with LaBSE (Feng et al., 2022) embedding similarity scores between 0.9 and 0.95. Table 3 summarizes the SFT data statistics. The final SFT corpus is also balanced, with 50% En–Ja and 50% Ja–En pairs.

The SFT hyperparameters are also listed in Table 2.

3.3 Prompt Engineering

We use one prompt for Ja–En translation and three distinct prompts for En–Ja translation.

Hyperparameter	CPT	SFT
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$) (Loshchilov and Hutter, 2019)	
Learning rate	2.5×10^{-5}	1×10^{-6}
Scheduler	cosine	inverse square root
Warmup ratio	1%	1%
Weight decay	0.1	0.1
Gradient clip	1.0	1.0
Epoch	1	3
Batch size	1,024 chunks	64 sentence pairs
Chunk size	2,048 tokens	N/A
Accelerator	DeepSpeed ZeRO-2 (Rasley et al., 2020)	
Precision	bfloat16	bfloat16

Table 2: Hyperparameters of CPT and SFT.

Usage	**Time Period**	**Data Type**	**Sentence Pairs**	**English Words**
CPT	2016∼2019, 2020(non-claims)	train	97,491,362	3.09B
		dev	10,000	317K
SFT	2020	train	30,000	824K
		dev	3,000	88.9K

Table 3: Usage and Details of Patent Parallel Data

Ja–En Prompt

The prompt used for Ja–En translation is as follows:

Ja–En Prompt

これを日本語から英語に翻訳してください。
ただし文頭に関係のない数字を出さないようにしてください。：
日本語: {japanese_text}
英語:

The English translation of the above prompt is: "Translate this from Japanese to English. However, do not start the sentence with an irrelevant number."

En–Ja Prompts

The three distinct prompts used for En–Ja translation are shown below.

En–Ja Prompt 1 : Not Primary

Translate this from English to Japanese:
English: {English_text}
Japanese:

SFT Configuration		En–Ja(Ref 2)		Ja–En			
Data Construction	Data Filtering	BLEU	COMET	Ref 1		Ref 2	
				BLEU	COMET	BLEU	COMET
line-by-line	length-based	48.0	89.63	59.4	84.59	65.3	84.96
line-by-line	LaBSE and length-based	49.4	89.59	58.5	84.65	65.3	85.13
claim-level	**LaBSE-based**	**49.3**	**89.41**	**63.0**	**85.10**	**70.4**	**85.62**

(a) Comparison of SFT Data Preparation Strategies (All SFT models are initialized from the CPT model.)

Training Configuration		En–Ja (Ref 2)		Ja–En			
CPT	SFT	BLEU	COMET	Ref 1		Ref 2	
				BLEU	COMET	BLEU	COMET
✗	✓	24.5	87.67	16.8	75.54	20.0	75.97
✓	✓	**49.3**	**89.41**	**63.0**	**85.10**	**70.4**	**85.62**

(b) Comparison of SFT-only vs CPT+SFT.

Table 4: Automatic Evaluation Results on the WAT2025 Development Sets (Underlined configuration denotes the one used in our submission system.)

En–Ja Prompt 2 : Primary

Translate from English to Japanese.
Keep all meanings. Do not skip or invent anything.
English: {English_text}
Japanese:

En–Ja Prompt 3 : Not Primary

Translate this from English to Japanese.
Do not include anything unrelated to the input.
English: {English_text}
Japanese:

4 Experiments

4.1 Ablation study of training methods

We investigate the effects of each training method.

Setup To validate our SFT data preparation strategy, we conduct a comparative study on different configurations, as detailed in Table 4a. All SFT models are initialized from the same CPT model. To separately analyze the contribution of CPT itself, we additionally report a comparison between models trained with SFT only and those trained with CPT followed by SFT. The results are summarized in Table 4b. Specifically, we investigate the impact of data construction and the corresponding filtering methods:

- **Data Construction:** We compare models trained on the original **line-by-line** data against the **claim-level** data used in our submission system.
- **Data Filtering:** We apply filtering strategies appropriate for each construction method. For the **line-by-line** data, which includes many short segments, we test a **length-based** filter and a combination of **LaBSE and length-based** filters. For our **claim-level** data, where sentences are already concatenated and sufficiently long, we apply only the **LaBSE-based** filter (Sub.).

All models are trained with the same hyperparameters as our submission system, as described in Section 4.1, unless otherwise noted.

For En-Ja translation, we used prompt 2, as described in Section 3.3.

Results The results of the automatic evaluation on the WAT2025 Patent Claims Translation/Evaluation Tasks development sets are presented in Table 4. Although two references are publicly available for both En–Ja and Ja–En, only reference 2 is used for the En–Ja evaluation due to omissions found in reference 1, while both references are reported for Ja–En. As shown in Table 4a, our submission configuration—claim-level data construction combined with LaBSE-based filtering—achieves the best performance across both Ja–En reference sets (Ref 1: 63.0 BLEU, Ref 2: 70.4 BLEU) and also maintains competitive performance in En–Ja. Furthermore, Table 4b demonstrates that CPT+SFT yields substantial improvements over SFT-only training in all evaluation settings, confirming the effectiveness of CPT as a pretraining stage.

5 Conclusion

We built our system for the WAT2025 Patent Claim Translation/Evaluation Task. Our model was trained with the combinations of CPT and SFT, initializing from a pretrained LLM (rinna/llama-3-youko-8b). To mitigate omissions and hallucinations, we generated translations via prompt engineering, especially for the En–Ja direction.

In our experiments, we observed that the SFT data preparation strategy is a critical factor for patent translation. We demonstrated that our submission's approach—using **claim-level** data construction and **LaBSE-based** filtering—yielded the best performance, particularly in the Ja–En direction. This highlights the importance of aligning SFT data with the logical structure of patent claims, rather than using simple line-by-line data.

Nevertheless, as patent claims often contain complex dependencies, eliminating omissions and hallucinations remains a challenge. We hope to further improve the adequacy and robustness of patent claim translation in future work.

Limitations

Small Development Set Size Although we conducted a comparative analysis in our ablation study (Section 4.1), the development sets provided by the task organizers are small. Therefore, it remains uncertain whether the strong performance of our submission configuration will generalize robustly across all types of patent claims.

Data Construction Imperfections Our claim-level data construction method relies on automatic segmentation using end-of-sentence symbols (Section 3.2). However, exceptions to these rules exist, which may lead to some data pairs having broken parallel relationships. Although we employed LaBSE-based filtering to mitigate this issue, it is not guaranteed that this filtering process successfully eliminated all such misaligned pairs from the SFT dataset.

Acknowledgements

This work partially used computational resources of Pegasus provided by the Multidisciplinary Cooperative Research Program in the Center for Computational Sciences, University of Tsukuba.

Author Contributions

Haruto Azami applied CPT and SFT,conducted translation experiments as described in Section 4.1 and other preliminary experiments,and selected the submission system.
Zhang Yin filtered SFT data using LaBSE embedding similarity scores.
Futo Kajita checked translation results to select the submission system.
Nobuyori Nishimura checked translation results to select the submission system.
Takehito Utsuro built and managed our team.

References

Haruto Azami, Minato Kondo, Takehito Utsuro, and Masaaki Nagata. 2025. Patent claim translation via continual pre-training of large language models with parallel data. In Proceedings of Machine Translation Summit XX: Volume 1, pages 300–314, Geneva, Switzerland. European Association for Machine Translation.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In The Eleventh International Conference on Learning Representations.

Minato Kondo, Takehito Utsuro, and Masaaki Nagata. 2024. Enhancing translation accuracy of large language models through continual pre-training on parallel data. In Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024), pages 203–220, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

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

Masaaki Nagata, Katsuki Chousa, and Norihito Yasuda. 2025. Japarapat: A large-scale japanese-english parallel patent application corpus.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey.

Segmentation Beyond Defaults: Asymmetrical Byte Pair Encoding for Optimal Machine Translation Performance

Saumitra Yadav and **Manish Shrivastava**
Language Technologies Research Center, KCIS,
International Institute of Information Technology Hyderabad, India
saumitra.yadav@research.iiit.ac.in and m.shrivastava@iiit.ac.in

Abstract

Existing Machine Translation (MT) research often suggests a single, fixed set of hyperparameters for word segmentation models, **symmetric Byte Pair Encoding** (BPE), which applies the same number of merge operations (NMO) to train tokenizers for both source and target languages. However, we demonstrate that this uniform approach doesn't guarantee optimal MT performance across different language pairs and data sizes. This work investigates BPE segmentation recipes across various data volumes and language pairs to evaluate MT system performance. We find that utilizing **asymmetric BPE**—where the source and target languages have different NMOs—significantly improves results over the symmetric approach, especially in low-resource settings (50K, 100K, and 500K sentence pairs). Specifically, asymmetric BPE yield statistically significant ($p<0.05$) average gains of 5.32, 4.46, and 0.7 CHRF++ on English-Hindi in low-resource setups (50K, 100K, and 500K sentence pairs, respectively). We validated this trend across six additional language pairs (English↔Telugu, Shona, Norwegian, Kyrgyz, Hausa, and Inuktitut), observing statistically significant improvement in 10 out of 12 systems compared to symmetric BPE. Our findings indicate a high NMO for the source (4K to 32K) and a low NMO for the target (0.5K to 2K) provides optimal results, particularly benefiting low-resource MT.

1 Introduction

Efforts have been made to include low-resource language pairs in Neural Machine Translation (NMT), e.g. Workshop on Technologies for MT of Low Resource Languages. Often, successful past methodologies on high-resource language pairs, like hyperparameters for preprocessing, are used without considering their suitability for specific language pairs. For example, if we take a preprocessing step, such as word segmentation, a key preprocessing step, divides words into subwords to enhance learning and manage vocabulary size, handling rare and unknown words to boost MT performance. Notable Techniques include BPE (Sennrich et al., 2016), word piece (Devlin et al., 2019), sentence piece (Kudo and Richardson, 2018), and morfessor (Smit et al., 2014). BPE compresses data by merging frequent character pairs into symbols (Gage, 1994), with the *number of merge operations* (NMO) as a key parameter. A lower NMO (e.g., 500, Table 1) reduces vocabulary size with more segmentation, while a higher NMO (e.g., 32K) results in larger vocabularies and less segmentation. Typically, the same NMO is applied to both source and target languages. Recent work have shown that examining BPE parameters in low-resource MT is vital (Ding et al., 2019; Abid, 2020), but uniform NMOs for source and target (symmetrical BPE) (Huck et al., 2017; Ortega et al., 2020; Lankford et al., 2021; Domingo et al., 2023; Lee et al., 2024) prevail, with little exploration of asymmetrical BPE in MT. Earlier work Ngo Ho and Yvon (2021) looked at asymmetric BPE for language alignment, not for MT. Our work is a result of a multi-year exploration of the impact of asymmetrical subword segmentation in bilingual MT systems.

While we acknowledge the rise of multilingual and decoder-only models, our study focuses on the effect of asymmetric BPE in bilingual setups, particularly in low-resource conditions where pretrained tokenizers or joint vocabularies may be unavailable. Bilingual systems remain a research focus, with studies in Cantonese-Mandarin (Liu, 2022), English-Luganda (Kimera et al., 2025), Wolof-French (Dione et al., 2022), Bavarian-German (Her and Kruschwitz, 2024), and English-Manipuri (Singh et al., 2023; Singh and Singh, 2022) using bilingual data and transformer-based architectures with customized subword segmentation like BPE or morphology-aware tokenization. These efforts, along with Li et al. (2024),

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 35–53
December 24, 2025 ©2025 Association for Computational Linguistics

Sentence	bosusco , 54 , runs an adventure tourism bureau .
500 NMO	bo@@ su@@ sc@@ o , 5@@ 4 , r@@ un@@ s an ad@@ v@@ en@@ ture t@@ our@@ is@@ m bu@@ re@@ a@@ u .
32K NMO	bo@@ su@@ sco , 54 , runs an adventure tourism bureau .

Table 1: Effect of NMO variation: 500 NMO yields highly segmented tokens, while 32K retains most vocabulary

cover underrepresented languages and diverse writing systems, proving the continued relevance of bilingual systems. Our work investigates asymmetrical BPE's impact on bilingual MT systems, utilizing different merge operation counts for source and target languages across varied dataset sizes and resources. Extending these results to multilingual or decoder-only models is beyond this work's scope but represents an interesting future direction.

We define the "BPE configuration" as $m_1\_m_2$, with m_1 and m_2 representing the merge operations for source and target languages. Our study on symmetric and asymmetric BPE configurations for English–Hindi under varying data conditions shows asymmetric configurations performing best, especially in low-resource context. We extend these insights to six additional language pairs—English ↔ Telugu, Shona, Norwegian, Kyrgyz, Hausa, Inuktitut—selected for diverse language families and morphological typologies. **Our findings consistently demonstrate that, in low-resource environments, the most effective BPE configuration for the majority of language translation directions tends to be asymmetric. Specifically, setups with *4K* to *32K* NMO for the source and *500* to *2K* for the target outperform symmetric BPE configurations.**

Section 2 summarizes previous efforts to use symmetric BPE merge operations to improve MT performance. Section 3 explains our motivation for finding optimal BPE configurations by exploring asymmetric BPE. Section 4 outlines our experimental setup and presents the performance of the English-Hindi MT system on FLORES and Domain testsets. Section 5 evaluates the setup for other language pairs in low resource context, concluding our observations in Section 6.

2 Related Work - Symmetrical BPE

Most bilingual MT systems—especially for low-resource pairs—use the same number of merge operations (NMO) for source and target languages. Studies show that smaller vocabularies (0–4K NMO) outperform the common 32K setting by up to 4 BLEU points in low-resource scenarios (Ding et al., 2019); similar patterns are reported for English–Egyptian, English–Levantine (Abid, 2020), and English–Irish (Lankford et al., 2021).

Other work adapts segmentation for polysynthetic languages (Ortega et al., 2020), rich morphology (Lee et al., 2024), or target-side variation (Domingo et al., 2023). Alternative strategies include cascading segmentations (Huck et al., 2017), vocabulary refinement (Xu et al., 2021), and multi-BPE–setting corpora (Poncelas et al., 2020). While (Ngo Ho and Yvon, 2021) varied NMOs for alignment, no prior study systematically evaluates asymmetric BPE—using different NMOs for source and target—across resource levels. This work addresses that gap.

Though multilingual MT research now dominates, bilingual MT remains vital for low-resource pairs, where symmetric BPE is still common (Liu, 2022; Kimera et al., 2025; Dione et al., 2022; Her and Kruschwitz, 2024; Singh et al., 2023; Singh and Singh, 2022). Recent work on Parity-Aware BPE (Foroutan et al., 2025) introduces fairness-oriented subword allocation, reducing disadvantages for low-resource languages in multilingual tokenization. Although our experiments are limited to bilingual MT, asymmetric BPE could complement such fairness-aware methods in multilingual systems; extending this remains outside our current scope.

3 Exploring Asymmetrical BPE

In practice, for a BPE configuration $m_1\_m_2$, the values of m_1 and m_2 are usually the same, with the number of merge operations (NMO) ranging from *8K* to *40K* (Wu, 2016; Denkowski and Neubig, 2017; Cherry et al., 2018; Renduchintala et al., 2019). However, Ding et al. (2019); Dewangan et al. (2021) found these settings suboptimal for low-resource language pairs. Ding et al. (2019) observed that $m_1 = m_2 \leq$ *4K* NMO outperforms *32K* in low-resource conditions, consistent with our experiments on 0.1 million sentence pairs (English ↔ {Hindi, Telugu}) (Figure 1). Dewangan et al.

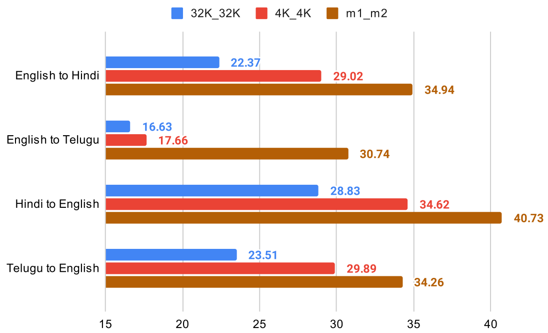


Figure 1: CHRF++ Scores for Symmetrical BPE (*32K*,*4K*) vs Asymmetrical BPE ($m_1 \neq m_2$)

(2021) further showed that identical BPE configurations yield differing performance across language pairs, exemplified by English-Hindi vs. English-Telugu comparisons at *4K* NMO (Figure 1).

Work by Ortega et al. (2020); Mujadia and Sharma (2021) suggests that selecting NMO should be done while considering dataset size and language pair, as nuanced BPE strategies benefit morphologically complex languages. We study symmetrical BPE configurations with identical NMOs for source and target, and investigate alternatives by varying m_1 and m_2 independently in English-Hindi across datasets from 50K to 8M sentences. This approach improves results in low-resource settings (Figure 1). Extensive experiments on English-Hindi, evaluated on FLORES (Goyal et al., 2022), confirm better performance of atypical BPE for tokenization. We further validate these findings by extending experiments to English ↔ {Telugu, Shona, Norwegian, Kyrgyz, Hausa, Inuktitut}. Our results strongly support optimizing NMO based on training data size and language pair. Figure 2 presents a conceptual overview of the **optimal ranges** for **BPE configurations** found in English-Hindi across resource settings. Here, “ranges” indicate the spectrum of NMO values used as hyperparameters for source and target subword tokenization in word segmentation. The performance gap between the best and symmetrical BPE systems is shown by shades of green, with the largest gains in low-resource scenarios (darker green). As dataset size increases, performance differences among configurations diminish (lighter green).

4 Evaluation on English ↔ Hindi

We explore BPE configurations with the Samanantar dataset (Ramesh et al., 2022) for English-Hindi containing 8 million parallel sentences. English text is tokenized, normalized, and lowercased using Moses scripts[1], while preprocessing of Hindi utilizes the Indic NLP library (Kunchukuttan, 2020). We simulate various training set sizes by grouping sentences based on English sentence length (Table 2) and randomly sample datasets of sizes 0.05M, 0.1M, 0.5M, 1M, 4M, and 8M, maintaining sentence length proportions (see Appendix A.1 for details). The BPE tokenizer is trained per language and dataset size with eight NMOs: *0.5K*, *1K*, *2K*, *4K*, *8K*, *16K*, *25K*, and *32K*.

All possible BPE configurations (e.g., src$_{500}$-tgt$_{500}$, src$_{500}$-tgt$_{1000}$) are trained using the Transformer architecture (Vaswani et al., 2017) with hyperparameters detailed in Appendix A.2. Training a single BPE configuration $m_1\_m_2$ across all dataset sizes averages 1040 GPU hours on a 1080TI, resulting in 64 configurations per language direction and 768 total systems (64 configurations × 6 dataset sizes × 2 directions). For evaluation, we use the FLORES dataset (Goyal et al., 2022) and report CHRF++ scores (Popović, 2015) to analyze the impact of different BPE configurations. We adopt CHRF++ rather than embedding-based metrics such as COMET (Rei et al., 2022), as not all language pairs have COMET support and we aim to compare performance using a consistent metric across all pairs. Validation and test set statistics are provided in Appendix A.8.

4.1 Best and Worst Configurations

To maintain clarity and brevity in our observations, Tables 3 and Table 4 show the performance of five selected configurations out of 64. For each dataset size, the systems represented are:

- High A and B: The two systems with the highest performance across all asymmetric configurations for each dataset size.
- Low A and B: The two systems with the lowest performance across all asymmetric configurations for each dataset size.
- Baseline: The best system among all symmetric BPE configurations (m_m, where $m \subset \{500,1K,2K,4K,8K,16K,25K,32K\}$).

Performance of all configurations for all systems is provided in the Appendix A.3.

[1] https://github.com/moses-smt/mosesdecoder/

Length bin	1 to 10	11 to 15	16 to 20	21 to 25	26 to 30	31 to 35	35 to 40	>=41	Total
No. of sentences	2792334	1655162	1150396	854091	617318	420583	275774	414926	8180584
Percentage	34.13	20.23	14.06	10.44	7.55	5.14	3.37	5.07	100

Table 2: Distribution of sentences in groups based on token length for full data

Dataset Size	0.05 M				0.1 M				0.5 M			
Performance Tier	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ
Low A	500	1K	19.56	-3.93	500	25K	23.36	-15.92	2K	32K	48.92	-3.53
Low B	500	2K	19.58	-3.91	1K	32K	24.2	-15.08	25K	32K	49.62	-2.83
Baseline	4K	4K	23.49	0	500	500	39.28	0	4K	4K	52.45	0
High B	25K	500	**28.47***	4.98	16K	500	**40.66***	1.38	8K	2K	**53.19***	0.74
High A	16K	500	**29.33***	5.84	8K	500	**40.75***	1.47	4K	500	**53.37***	0.92
Dataset Size	1 M				4 M				8 M			
Performance Tier	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ
Low A	500	32K	53.27	-1.77	500	1K	56.1	-1.73	500	2K	56.26	-2.45
Low B	1K	32K	53.58	-1.46	1K	2K	56.3	-1.53	500	500	56.43	-2.28
Baseline	8K	8K	55.04	0	32K	32K	57.83	0	32K	32K	58.71	0
High B	16K	8K	55.19	0.15	32K	16K	58.06	0.23	16K	25K	58.74	0.03
High A	16K	4K	55.39	0.35	25K	16K	58.18	0.35	4K	32K	58.75	0.04

Table 3: Performance of the top 2 (High A, High B) and bottom 2 (Low A, Low B) tokenization configurations compared to the symmetric baseline for Hindi-to-English across dataset sizes. Bold indicates statistically significant improvement over baseline ($p < 0.05$); bold with * denotes high significance ($p < 0.01$). δ shows CHRF++ difference from best baseline. **src** and **tgt** are source and target merge operations (NMO).

Dataset Size	0.05 M				0.1 M				0.5 M			
Performance Tier	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ
Low A	1K	25K	13	-5.39	500	32K	16.49	-12.55	500	32K	43.57	-3.5
Low B	500	4K	13.55	-4.84	500	25K	16.74	-12.3	1K	32K	43.88	-3.19
Baseline	8K	8K	18.39	0	4K	4K	29.04	0	4K	4K	47.07	0
High B	16K	500	**23.19***	4.8	16K	500	**34.73***	5.69	8K	500	47.12	0.05
High A	8K	500	**23.83***	5.44	8K	500	**35***	5.96	4K	500	**47.55**	0.48
Dataset Size	1 M				4 M				8 M			
Performance Tier	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ	src	tgt	CHRF++	δ
Low A	1K	32K	47.23	-1.93	8K	2K	50.64	-1.12	500	1K	50.79	-1.84
Low B	2K	32K	47.83	-1.33	500	2K	50.73	-1.03	32K	2K	51.29	-1.34
Baseline	8K	8K	49.16	0	16K	16K	51.76	0	25K	25K	52.63	0
High B	4K	2K	**49.74**	0.58	16K	32K	51.95	0.19	25K	32K	52.63	0
High A	8K	2K	**49.75**	0.59	32K	25K	52	0.24	16K	25K	**53**	0.37

Table 4: Performance of the top 2 (High A, High B) and bottom 2 (Low A, Low B) tokenization configurations compared to the symmetric baseline for English-to-Hindi across dataset sizes. Bold indicates statistically significant improvement over baseline ($p < 0.05$); bold with * denotes high significance ($p < 0.01$). δ shows CHRF++ difference from best baseline. **src** and **tgt** are source and target merge operations (NMO).

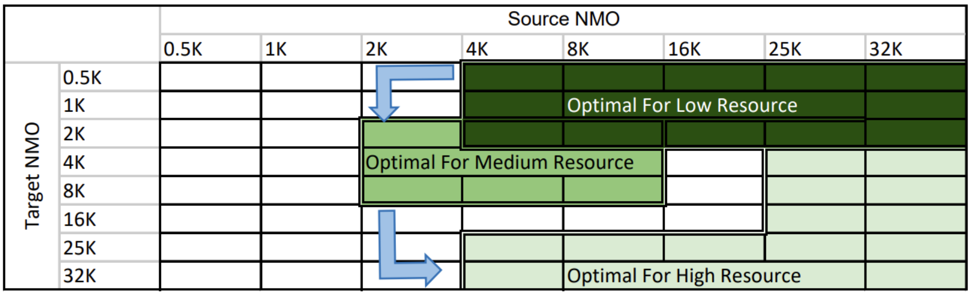


Figure 2: Changes in Optimal BPE Configuration from Low- to High-Resource Settings

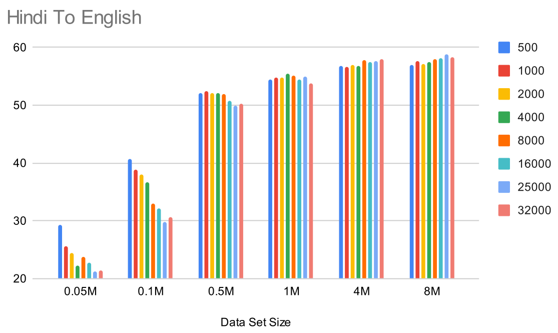


Figure 3: CHRF++ scores for 0.1M sentence pairs for *Hindi-to-English* MT systems using configurations of the form *16K_x*, where $x \in \{$*500*, *1K*, *2K*, *4K*, *8K*, *16K*, *25K*, *32K*$\}$.

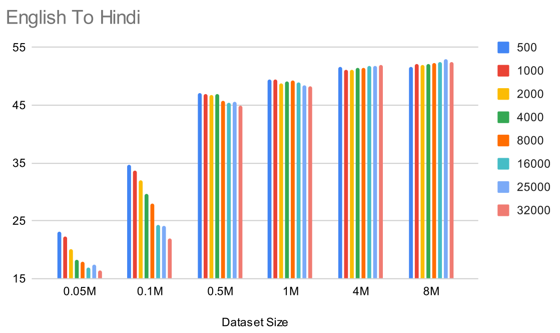


Figure 4: CHRF++ scores for 0.1M sentence pairs for *English-to-Hindi* MT systems using configurations of the form *16K_x*, where $x \in \{$*500*, *1K*, *2K*, *4K*, *8K*, *16K*, *25K*, *32K*$\}$.

As shown in Tables 3 and 4, for low-resource settings (<1M), the best system outperforms the weakest by ≈15 CHRF++ scores and the best symmetric BPE by ≈5. In medium-resource scenarios (1M), the optimal source and target NMO shift to the medium range (2K–8K), with smaller performance variation (≈3 CHRF++). For high-resource settings, the difference between best and worst configurations is minimal (< 2 CHRF++), with the best system using 32K NMO on the target. This highlights the advantage of asymmetric BPE in low-resource contexts. This trend of shifting optimal BPE values with dataset size also appears when varying target NMO while keeping source NMO fixed. For example, English↔Hindi systems with source NMO fixed at 16K on 0.1M data (Figures 3 and 4) show gradual performance changes as target NMO varies from 500 to 32K. Similar patterns with other fixed source or target values are detailed in Appendix A.3. This highlights that modifying the NMO on the target side, especially in a low-resource scenario, plays a vital role in determining the optimal BPE configuration.

We conclusively find that symmetric BPE configurations underperform compared to asymmetric ones in low-resource MT systems. As dataset size grows, symmetric configurations perform comparably to asymmetric. Nonetheless, asymmetric BPE yields statistically significant improvements in low-resource settings.

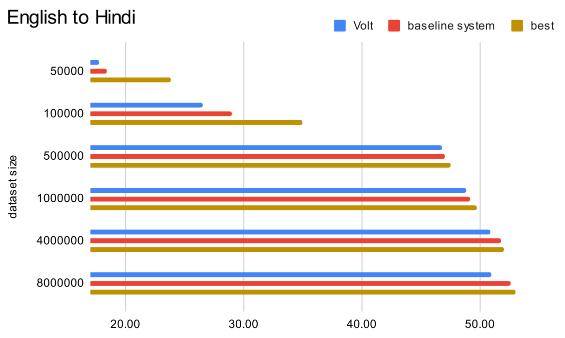


Figure 5: CHRF++ score comparison of Asymmetric BPE with VOLT for English to Hindi

We compare our systems with optimal BPE configurations against VOLT (Xu et al., 2021)[2]. Figures 5 and 6 show CHRF++ comparisons between VOLT tokenization, optimal BPE, and "best" baseline symmetric BPE (source NMO = target NMO)

[2] Using hyperparameters specified in the original paper.

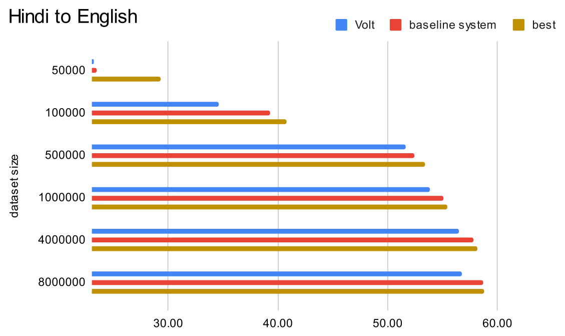


Figure 6: CHRF++ score comparison of Asymmetric BPE with VOLT for Hindi to English

Domain	# of Sentences	English Tokens	Hindi Tokens
Artificial Intelligence	389	6965	8441
Chemistry	392	7761	9368

Table 5: Statistics of ICON 2020 Domain Adaptation Testset

configuration. Systems using asymmetric BPE outperform VOLT across all dataset sizes, with statistically significant improvements ($p < 0.05$) especially in low-resource settings.

4.2 Performance on Domain Test

Subword models must handle rare or unseen words, making domain-specific datasets effective for evaluating asymmetric BPE in MT systems. Thus, to demonstrate the impact of segmentation strategies, we evaluate all systems on Artificial Intelligence (AI) and Chemistry (CH) domain test sets from the ICON 2020 Domain Adaptation Task[3]. Table 5[4] presents domain test data statistics. Table 6 show the performance of configurations from Table 4 on domain datasets for English-to-Hindi systems. Performance of Hindi-to-English systems is given in Appendix A.4.

For English↔Hindi domain test set translation, we observe:

- In low- to medium-resource settings, asymmetric BPE systems outperform baselines significantly when source NMO is much higher than target NMO. This aligns with FLORES results (Tables 3 and 4) and highlights asymmetric BPE benefits for domain translation with limited data.
- In high-resource settings, symmetric and asymmetric systems perform similarly.

These results demonstrate the potential translation improvements from asymmetric BPE in new domains under limited-resource conditions. Performances of all systems on AI and CH test sets is in Appendices A.5 and A.6, respectively.

Figure 7 illustrates, with an example on AI domain, the advantage of asymmetric BPE over symmetric BPE for 0.1M parallel sentences. Configurations like *16K_500* or *8K_500* produce more natural, semantically faithful Hindi translations than symmetric *32K_32K* or *4K_4K* setups. Translation improves as we move from symmetric high NMO (*32K_32K*), to symmetric low NMO (*4K_4K*), to asymmetric (*16K_500* or *8K_500*).

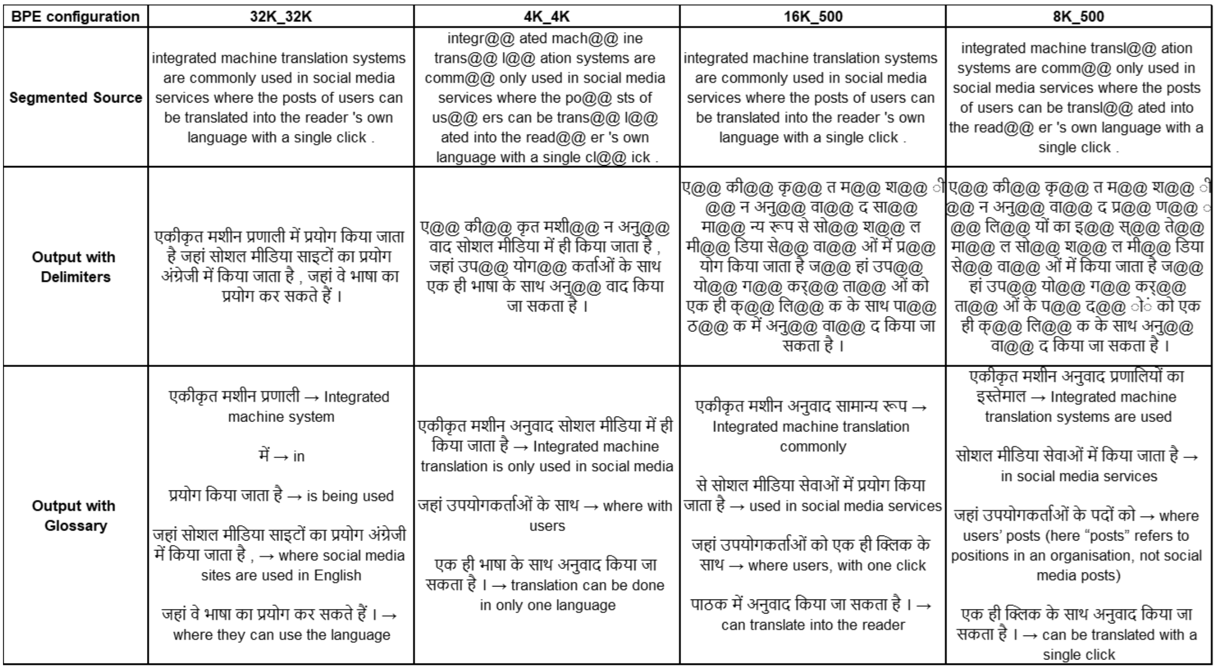
- ***32K_32K*** – In the output with delimiters, most of the tokens are already fully merged into complete words. While this segmentation yields a large vocabulary, in low-resource conditions, it results in sparsity: many source and target tokens appear too infrequently for effective parameter learning. Consequently, the network fails to learn robust mappings, leading to incomplete or inaccurate translations despite having fully merged tokens.
- ***4K_4K*** – The glossary shows an improvement in overall translation fluency, but important content words such as system, commonly and click are missing, both explicitly and implicitly (meaning that they cannot be inferred from context). The improvement is due to the increased recurrence of subword units in the training data from the reduced vocabulary size, which strengthens learned associations, but at the cost of certain semantic details.
- **Asymmetric (*16K_500*, *8K_500*)**: Better meaning preservation than symmetric. Whereas *16K_500* omits "post" and drops final language reference, *8K_500* conveys almost full meaning but mistranslates "post" as a job title. From a learning perspective, the smaller decoder vocabulary improves the alignment and connection learning between the source and target segments (similar to Ngo Ho and Yvon (2021)), aligning with previous findings (Domingo et al., 2023) that the target side vocabulary influences NMT performance. Although overly constrained vocabularies can still introduce semantic drift in rare or domain-specific terms, overall transla-

[3] We thank task organizers for access.

[4] After removing 12 and 5 lines from AI and CH test sets respectively, that overlapped with the 8M training set.

Dataset Size	0.05M				0.1M				0.5M			
Performance Tier	src	tgt	AI	CH	src	tgt	AI	CH	src	tgt	AI	CH
Low A	1K	25K	15.98	14.13	500	32K	18.46	16.67	500	32K	53.32	47.44
Low B	500	4K	15.97	15.03	500	25K	18.80	16.86	1K	32K	53.99	47
Baseline	8K	8K	20.76	19.34	4K	4K	35.79	32.19	4K	4K	58.63	50.64
High B	16K	500	**26.76***	**24.03***	16K	500	**42.97***	**37.94***	8K	500	58.91	50.94
High A	8K	500	**28.28***	**25.14***	8K	500	**44.05***	**38.57***	4K	500	58.70	**51.53**
Dataset Size	1M				4M				8M			
Performance Tier	src	tgt	AI	CH	src	tgt	AI	CH	src	tgt	AI	CH
Low A	1K	32K	58.58	51.78	8K	2K	62.23	54.55	500	1K	61.91	54.78
Low B	2K	32K	58.88	51.65	500	2K	61.51	54.01	32K	2K	62.52	54.63
Baseline	8K	8K	61.22	53.6	16K	16K	63.12	55.14	25K	25K	63.95	55.65
High B	4K	2K	60.39	53.55	16K	32K	63.21	**55.84**	25K	32K	63.9	55.92
High A	8K	2K	60.01	53.27	32K	25K	63.6	55.74	16K	25K	63.53	55.69

Table 6: Performance of the top 2 (High A and High B) and bottom 2 (Low A and Low B) systems with respective tokenisation configurations compared to the symmetric baseline for *English-to-Hindi* systems across dataset sizes for **AI** and **CH Domains**. Bold scores indicate statistically significant improvements over the baseline ($p < 0.05$); bold scores with an asterisk ($*$) indicate high significance ($p < 0.01$)



BPE configuration	32K_32K	4K_4K	16K_500	8K_500
Segmented Source	integrated machine translation systems are commonly used in social media services where the posts of users can be translated into the reader 's own language with a single click .	integr@@ ated mach@@ ine trans@@ l@@ ation systems are comm@@ only used in social media services where the po@@ sts of us@@ ers can be trans@@ l@@ ated into the read@@ er 's own language with a single cl@@ ick .	integrated machine translation systems are commonly used in social media services where the posts of users can be translated into the reader 's own language with a single click .	integrated machine transl@@ ation systems are comm@@ only used in social media services where the posts of users can be transl@@ ated into the read@@ er 's own language with a single click .
Output with Delimiters	एकीकृत मशीन प्रणाली में प्रयोग किया जाता है जहां सोशल मीडिया साइटों का प्रयोग अंग्रेजी में किया जाता है , जहां वे भाषा का प्रयोग कर सकते हैं ।	ए@@ की@@ कृत मशी@@ न अनु@@ वाद सोशल मीडिया में ही किया जाता है , जहां उप@@ योग@@ कर्ताओं के साथ एक ही भाषा के साथ अनु@@ वाद किया जा सकता है ।	ए@@ की@@ कृ@@ त म@@ श@@ ी @@ न अनु@@ वा@@ द सा@@ मा@@ न्य रूप से सो@@ श@@ ल मी@@ डिया से@@ वा@@ ओं में प्र@@ योग किया जाता है ज@@ हां उप@@ यो@@ ग@@ कर्@@ ता@@ ओं को एक ही क्@@ लि@@ क के साथ पा@@ ठ@@ क में अनु@@ वा@@ द किया जा सकता है ।	ए@@ की@@ कृ@@ त म@@ श@@ ी @@ न अनु@@ वा@@ द प्र@@ ण@@ ा @@ लि@@ यों का इ@@ स्@@ ते@@ मा@@ ल सो@@ श@@ ल मी@@ डिया से@@ वा@@ ओं में किया जाता है ज@@ हां उप@@ यो@@ ग@@ कर्@@ ता@@ ओं के प@@ द@@ ों को एक ही क्@@ लि@@ क के साथ अनु@@ वा@@ द किया जा सकता है ।
Output with Glossary	एकीकृत मशीन प्रणाली → Integrated machine system में → in प्रयोग किया जाता है → is being used जहां सोशल मीडिया साइटों का प्रयोग अंग्रेजी में किया जाता है , → where social media sites are used in English जहां वे भाषा का प्रयोग कर सकते हैं । → where they can use the language	एकीकृत मशीन अनुवाद सोशल मीडिया में ही किया जाता है → Integrated machine translation is only used in social media जहां उपयोगकर्ताओं के साथ → where with users एक ही भाषा के साथ अनुवाद किया जा सकता है । → translation can be done in only one language	एकीकृत मशीन अनुवाद सामान्य रूप → Integrated machine translation commonly से सोशल मीडिया सेवाओं में प्रयोग किया जाता है → used in social media services जहां उपयोगकर्ताओं को एक ही क्लिक के साथ → where users, with one click पाठक में अनुवाद किया जा सकता है । → can translate into the reader	एकीकृत मशीन अनुवाद प्रणालियों का इस्तेमाल → Integrated machine translation systems are used सोशल मीडिया सेवाओं में किया जाता है → in social media services जहां उपयोगकर्ताओं के पदों को → where users' posts (here "posts" refers to positions in an organisation, not social media posts) एक ही क्लिक के साथ अनुवाद किया जा सकता है । → can be translated with a single click

Figure 7: Examples of English-to-Hindi translations across different BPE configurations, showing segmented source text, outputs with delimiters '@@', and output without delimiters with corresponding English glossaries for each segment.

tion remains improved compared to symmetric configurations.

5 Exploring Asymmetrical BPE Configurations for other language pairs

To evaluate the transferability of optimal subword segmentation from English–Hindi to typologically diverse languages, we extend experiments to English↔{Telugu, Shona, Norwegian, Kyrgyz, Hausa, Inuktitut}. Corpora sources are:

- **English–{Hausa, Shona, Norwegian, Kyrgyz}**: Gowda et al. (2021)
- **English–Telugu**: Ramesh et al. (2022)
- **English–Inuktitut**: Joanis et al. (2020)

To simulate low-resource settings, we sampled 0.1M sentence pairs per language via sentence-length binning, analogous to English–Hindi, statistics are in Appendix A.7.

These language pairs were chosen to assess the impact of symmetric and asymmetric BPE configurations in low-resource scenarios across diverse language families with varying morphological and typological complexity. Baselines used symmetric BPE (*4K_4K*, *32K_32K*), while asymmetric settings (*8K_500*, *16K_500*) derive from English-Hindi optimal configurations at 0.1M sentence pairs. For evaluating we use the FLORES test set, except English↔Inuktitut tested on Joanis et al. (2020) (Appendix A.8).

Experiments are repeated three times for reproducibility (sampling, BPE training, model training). Figures 8 and 9 compare average asymmetric and symmetric BPE results for translations to and from English. Asymmetric BPE significantly improves four of six *L-to-English* systems and all *English-to-L* systems ($p < 0.05$, indicated by *), underscoring the benefits of asymmetric BPE and the need to explore beyond conventional settings for low-resource pairs.

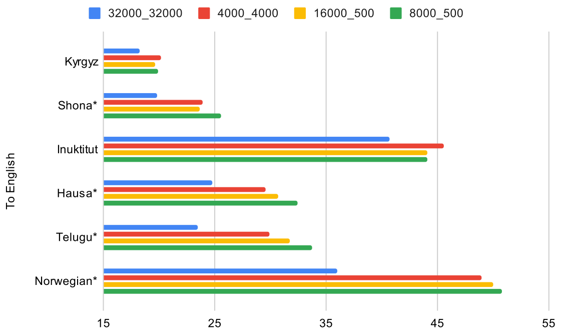


Figure 8: CHRF++ scores improvement with asymmetrical over symmetrical BPE for English to *L* Languages

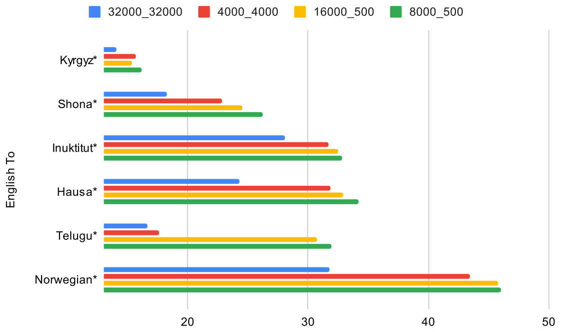


Figure 9: CHRF++ scores improvement with asymmetrical over symmetrical BPE from *L* Languages to English

6 Conclusion

In-depth examination of BPE configurations across diverse language pairs and differing dataset sizes reveals that typical configurations (*n_n*) do not always produce optimal results. As referenced in Section 2, in low-resource settings, systems benefit from using symmetric *n* NMO configurations when *n* is significantly smaller than *32K*; our experiments with asymmetric BPE *n_m* show that further improvement in translation performance is possible, under low-resource conditions, when *n* » *m* where *n*, *m* represent NMOs for source and target respectively. This study highlights the need to go beyond default segmentation in machine translation, especially for low-resource languages. While symmetric BPE configurations may suffice with medium to large datasets, their effectiveness drops in low-resource settings. Using asymmetric BPE—with a higher number of merge operations for the source language and fewer for the target—yields significant translation quality gains. These configurations consistently outperform across varied language families and morphological complexities, underscoring the importance of tailored segmentation for optimizing low-resource translation.

Limitation

This study is limited by the computational cost of exhaustively analysing all BPE configurations for each language pair and by its focus only on bilingual encoder–decoder NMT. However, the results show that certain configuration ranges consistently improve translation quality in low-resource settings, substantially reducing the search space. These findings suggest promising extensions to multilingual models, potentially combined with fairness-aware tokenisation such as Parity-Aware BPE (Foroutan et al., 2025) to deliver both performance gains and balanced vocabulary distribution.

References

Wael Abid. 2020. The SADID evaluation datasets for low-resource spoken language machine translation of Arabic dialects. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6030–6043, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Shubham Dewangan, Shreya Alva, Nitish Joshi, and Pushpak Bhattacharyya. 2021. Experience of neural machine translation between indian languages. *Machine Translation*, 35(1):71–99.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Cheikh M. Bamba Dione, Alla Lo, Elhadji Mamadou Nguer, and Sileye Ba. 2022. Low-resource neural machine translation: Benchmarking state-of-the-art transformer for Wolof<->French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6654–6661, Marseille, France. European Language Resources Association.

Miguel Domingo, Mercedes García-Martínez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2023. How much does tokenization affect neural machine translation? In *Computational Linguistics and Intelligent Text Processing*, pages 545–554, Cham. Springer Nature Switzerland.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. 2025. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization. *Preprint*, arXiv:2508.04796.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

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.

Wan-hua Her and Udo Kruschwitz. 2024. Investigating neural machine translation for low-resource languages: Using Bavarian as a case study. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 155–167, Torino, Italia. ELRA and ICCL.

Matthias Huck, Simon Riess, and Alexander Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Richard Kimera, DongNyeong Heo, Daniela N. Rim, and Heeyoul Choi. 2025. Data augmentation with back translation for low resource languages: A case of english and luganda. In *Proceedings of the 2024 8th International Conference on Natural Language Processing and Information Retrieval*, NLPIR '24, page 142–148, New York, NY, USA. Association for Computing Machinery.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Seamus Lankford, Haithem Alfi, and Andy Way. 2021. Transformers for low-resource languages: Is féidir linn! In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 48–60, Virtual. Association for Machine Translation in the Americas.

Jungseob Lee, Hyeonseok Moon, Seungjun Lee, Chanjun Park, Sugyeong Eo, Hyunwoong Ko, Jaehyung Seo, Seungyoon Lee, and Heuiseok Lim. 2024. Length-aware byte pair encoding for mitigating oversegmentation in Korean machine translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2287–2303, Bangkok, Thailand. Association for Computational Linguistics.

Fuxue Li, Beibei Liu, Hong Yan, Mingzhi Shao, Peijun Xie, Jiarui Li, and Chuncheng Chi. 2024. A bilingual templates data augmentation method for low-resource neural machine translation. In *Advanced Intelligent Computing Technology and Applications: 20th International Conference, ICIC 2024, Tianjin, China, August 5–8, 2024, Proceedings, Part III*, page 40–51, Berlin, Heidelberg. Springer-Verlag.

Evelyn Kai-Yan Liu. 2022. Low-resource neural machine translation: A case study of Cantonese. In *Proceedings of the Ninth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 28–40, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Vandan Mujadia and Dipti Misra Sharma. 2021. English-Marathi neural machine translation for LoResMT 2021. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 151–157, Virtual. Association for Machine Translation in the Americas.

Anh Khoa Ngo Ho and François Yvon. 2021. Optimizing word alignments with better subword tokenization. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 256–269, Virtual. Association for Machine Translation in the Americas.

John E Ortega, Richard Castro Mamani, and Kyunghyun Cho. 2020. Neural machine translation with a polysynthetic low resource language. *Machine Translation*, 34(4):325–346.

Alberto Poncelas, Jan Buts, James Hadley, and Andy Way. 2020. Using multiple subwords to improve English-Esperanto automated literary translation quality. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 108–117, Suzhou, China. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Adithya Renduchintala, Pamela Shapiro, Kevin Duh, and Philipp Koehn. 2019. Character-aware decoder for translation into morphologically rich languages. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 244–255, Dublin, Ireland. European Association for Machine Translation.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kshetrimayum Boynao Singh, Ningthoujam Avichandra Singh, Loitongbam Sanayai Meetei, Sivaji Bandyopadhyay, and Thoudam Doren Singh. 2023. NITS-CNLP low-resource neural machine translation systems of English-Manipuri language pair. In *Proceedings of the Eighth Conference on Machine Translation*, pages 967–971, Singapore. Association for Computational Linguistics.

Salam Michael Singh and Thoudam Doren Singh. 2022. Low resource machine translation of english–manipuri: A semi-supervised approach. *Expert Syst. Appl.*, 209(C).

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

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.

Yonghui Wu. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373, Online. Association for Computational Linguistics.

A Appendix

A.1 English–Hindi Training Data Statistics

We use an 8-million-sentence English–Hindi corpus from the Samanantar dataset and execute stratified random sampling across sentence length bins to simulate different resource availability levels. Table 7 summarises the statistics for sentence pairs corresponding to each level of resource availability.

A.2 Hyperparameters for Training Transformer Model

We followed the official Fairseq tutorial instructions for preprocessing, training, and translation[5], and customised the parameters given in Table 8 with respective values for all experiments.

A.3 Performance of all systems for English ↔ Hindi for all dataset scenarios

Figures 10 present the performance of all configurations for English ↔ Hindi systems in a low resource scenario (for data set sizes of 0.05M, 0.1M and 0.5M). And Figures 11 show the performance of all configurations on 1M, 4M and 8M dataset sizes. Each subgraph represents performance on a particular dataset size, with the x-axis being the source NMO. The black stepped dotted lines indicate the maximum CHRF++ score for each dataset size considering for each source NMOs. In figure 10 for low-resource environments (0.05M, 0.1M and 0.5M) systems, as noted by (Ding et al., 2019), the use of symmetric BPE configuration with lower NMOs improves performance over high NMOs. However, the best results are achieved using asymmetric BPE configurations when the source has a higher NMO than the target. We see a maximum performance gain when the source NMO is very high and the target NMO very low (we see consistent performance with the target NMO = *500*). Conversely, when the target's NMO is greater than that of the source, performance declines, like for the Hindi to English 0.1M dataset, performance of *500_25K* and *500_32K* was worse than symmetric BPE configurations.

A.4 Performance of Hindi-To-English Selected Configurations on Domain Test set

Table 9 shows the performance of the Highest and Lowest performing asymmetric BPE systems with baseline systems for Hindi-To-English systems. Like in English to Hindi systems, we see significant improvement when using asymmetric BPE configurations in low-resource settings.

A.5 Evaluation of English ↔ Hindi systems on AI for all BPE Configurations

Figures 12 and 13 depict the performance of all configurations for English ↔ Hindi systems during translations in the **AI** domain. A similar performance pattern appears across configurations here, as observed with the FLORES test set (see Appendix A.3).

A.6 Evaluation of English ↔ Hindi systems on Chemistry for all BPE Configurations

Figures 14 and 15 depict the performance of all configurations for English ↔ Hindi systems during translations in the **Chemistry** domain. A similar performance pattern appears across configurations here, as observed with the FLORES test set (see Appendix A.3).

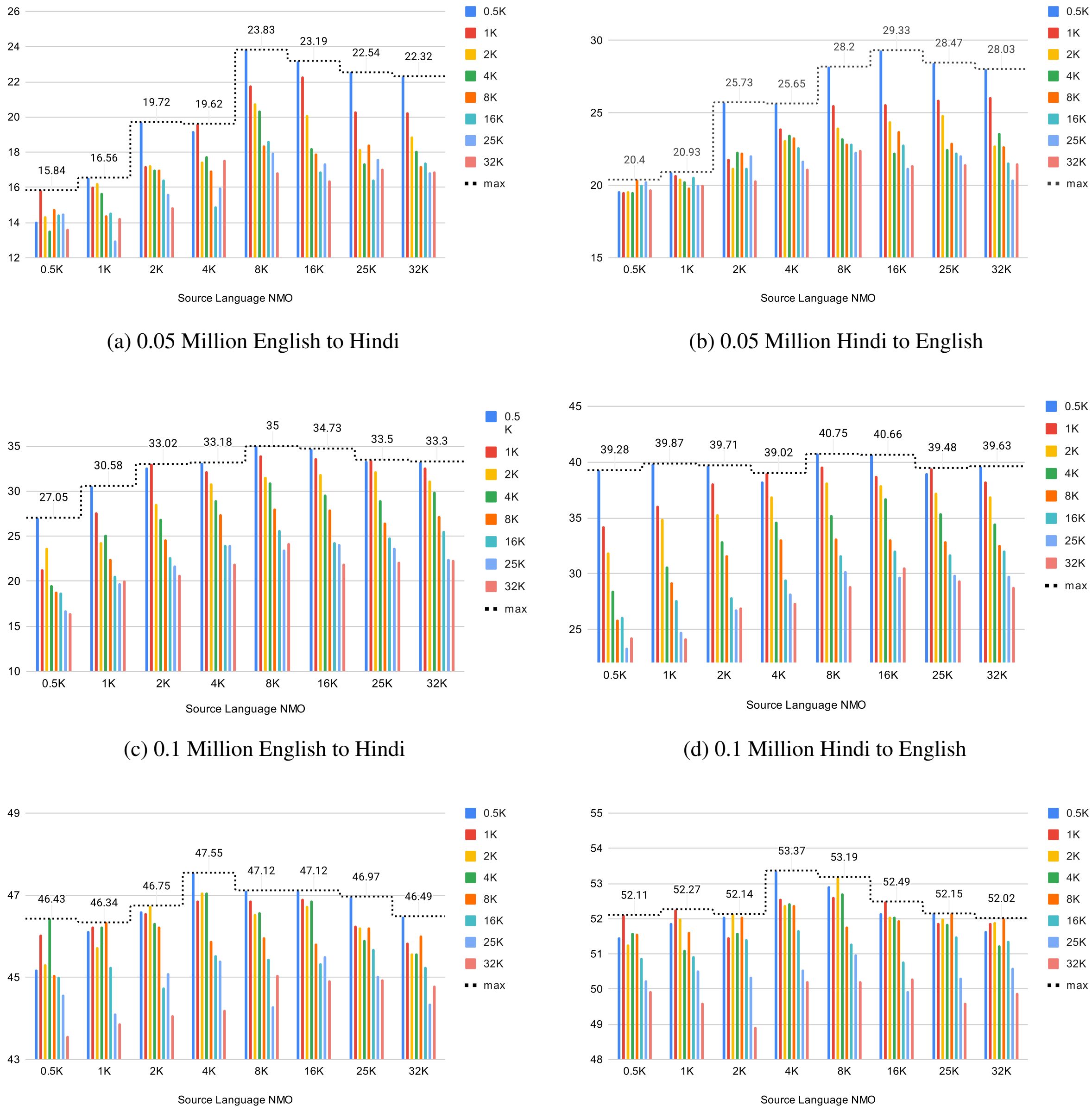
A.7 Statistics of Bitext for secondary set of experiments

Table 10 gives the statistics of the original bitext that we obtained for the secondary set of experiments, to see the transferability of asymmetric BPE configurations. And to simulate low-resource settings, we sampled 0.1M sentence pairs per language using sentence-length binning, as done for English–Hindi; statistics are shown in Table 11.

A.8 Validation and Test Set Statistics

As noted, for English–Inuktitut validation and test sets, we use Joanis et al. (2020). For all other language pairs, the FLORES dataset was used. Table 12 shows token-level statistics for validation and test sets across all language pairs.

[5] https://fairseq.readthedocs.io/en/latest/getting_started.html



(a) 0.05 Million English to Hindi

(b) 0.05 Million Hindi to English

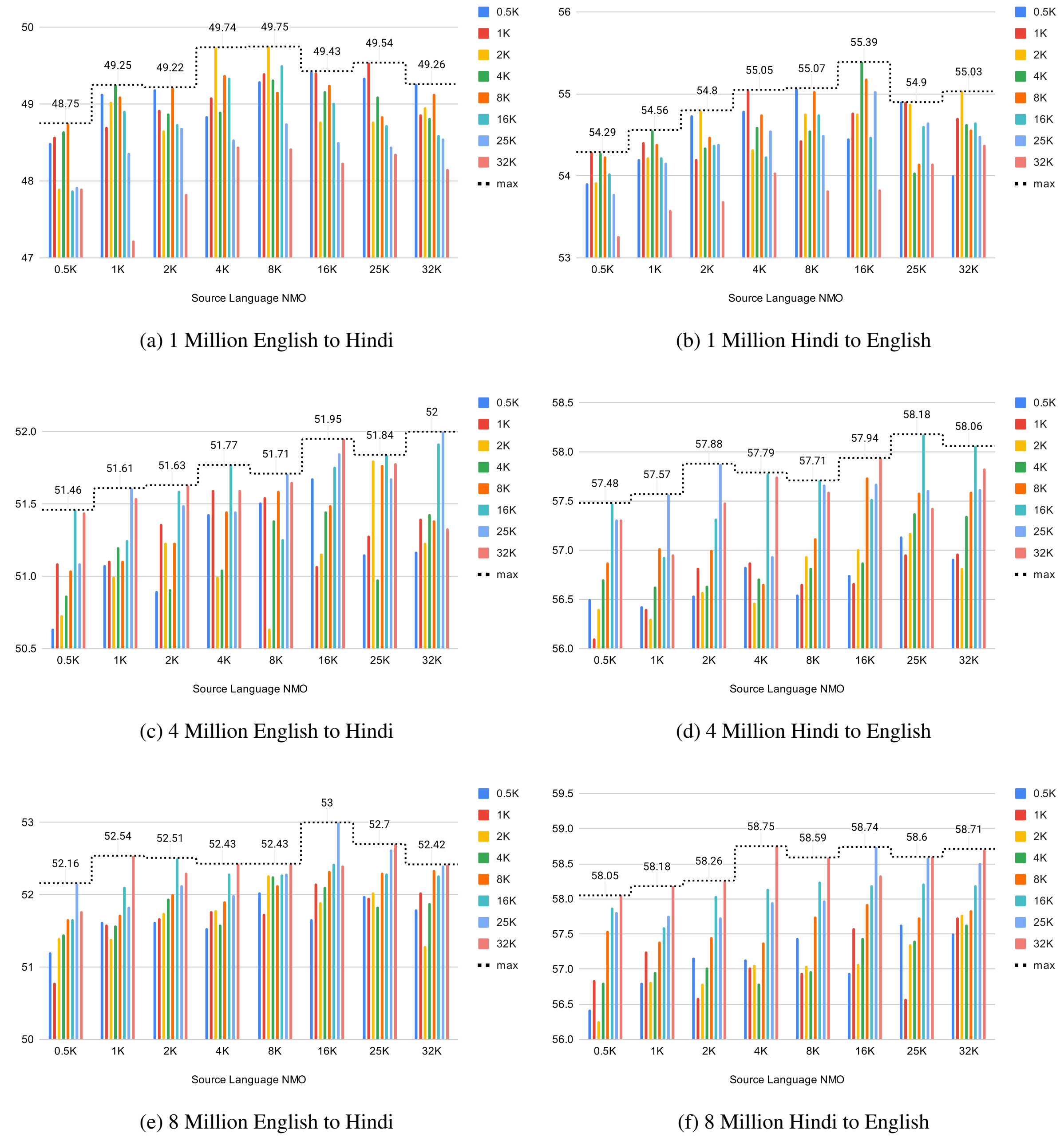
(c) 0.1 Million English to Hindi

(d) 0.1 Million Hindi to English

(e) 0.5 Million English to Hindi

(f) 0.5 Million Hindi to English

Figure 10: Evaluation of English ↔ Hindi MT Systems for 0.05M, 0.1M and 0.5M dataset sizes on FLORES, x-axis is source NMO and y-axis is CHRF++ scores



(a) 1 Million English to Hindi

(b) 1 Million Hindi to English

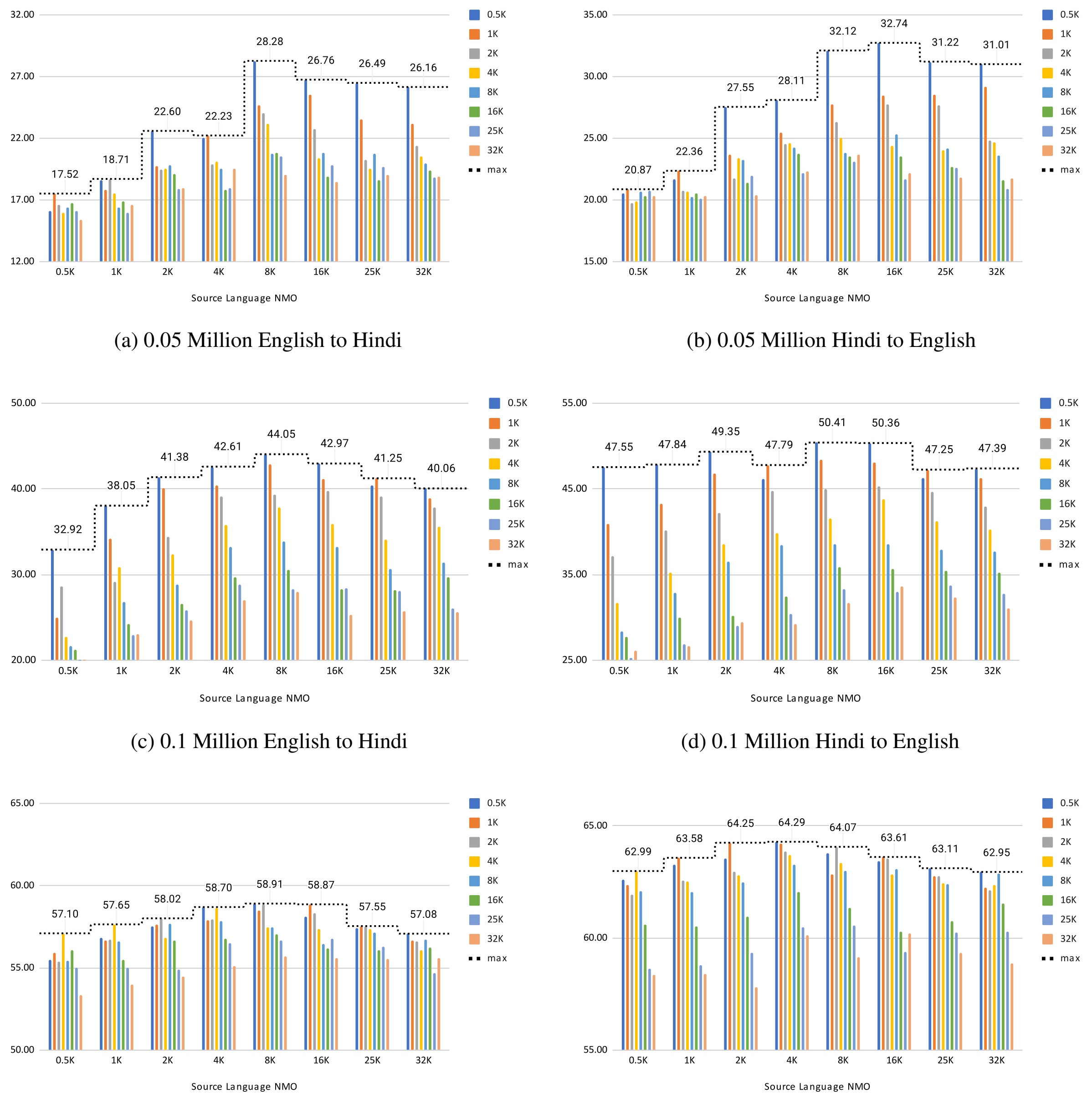
(c) 4 Million English to Hindi

(d) 4 Million Hindi to English

(e) 8 Million English to Hindi

(f) 8 Million Hindi to English

Figure 11: Evaluation of English ↔ Hindi MT Systems for 1M, 4M and 8M dataset sizes on FLORES, x-axis is source NMO and y-axis is CHRF++ scores



(a) 0.05 Million English to Hindi

(b) 0.05 Million Hindi to English

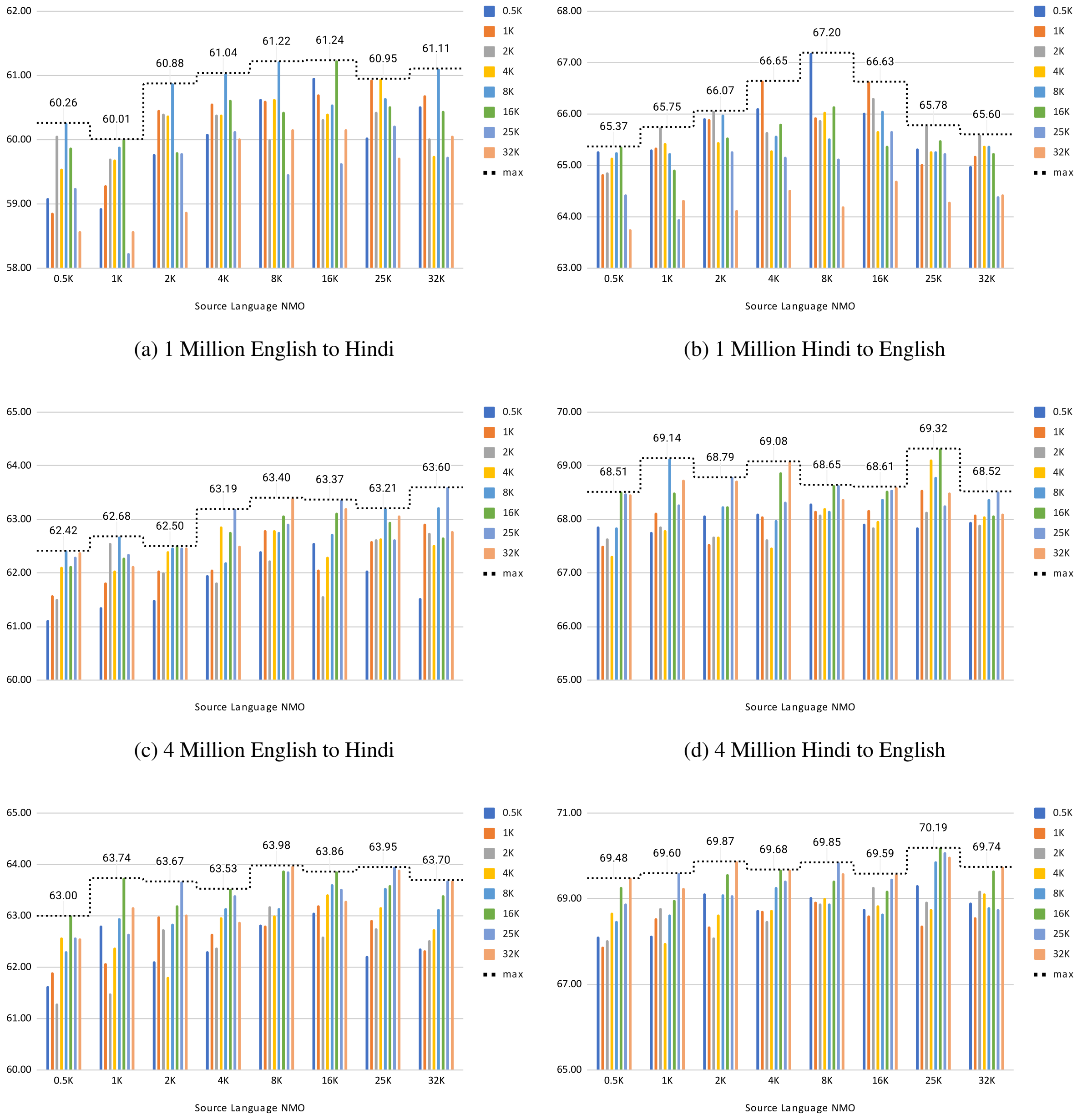
(c) 0.1 Million English to Hindi

(d) 0.1 Million Hindi to English

(e) 0.5 Million English to Hindi

(f) 0.5 Million Hindi to English

Figure 12: Evaluation of English ↔ Hindi MT Systems for 0.05M, 0.1M and 0.5M dataset sizes on **AI**, x-axis is source NMO and y-axis is CHRF++ scores



(a) 1 Million English to Hindi

(b) 1 Million Hindi to English

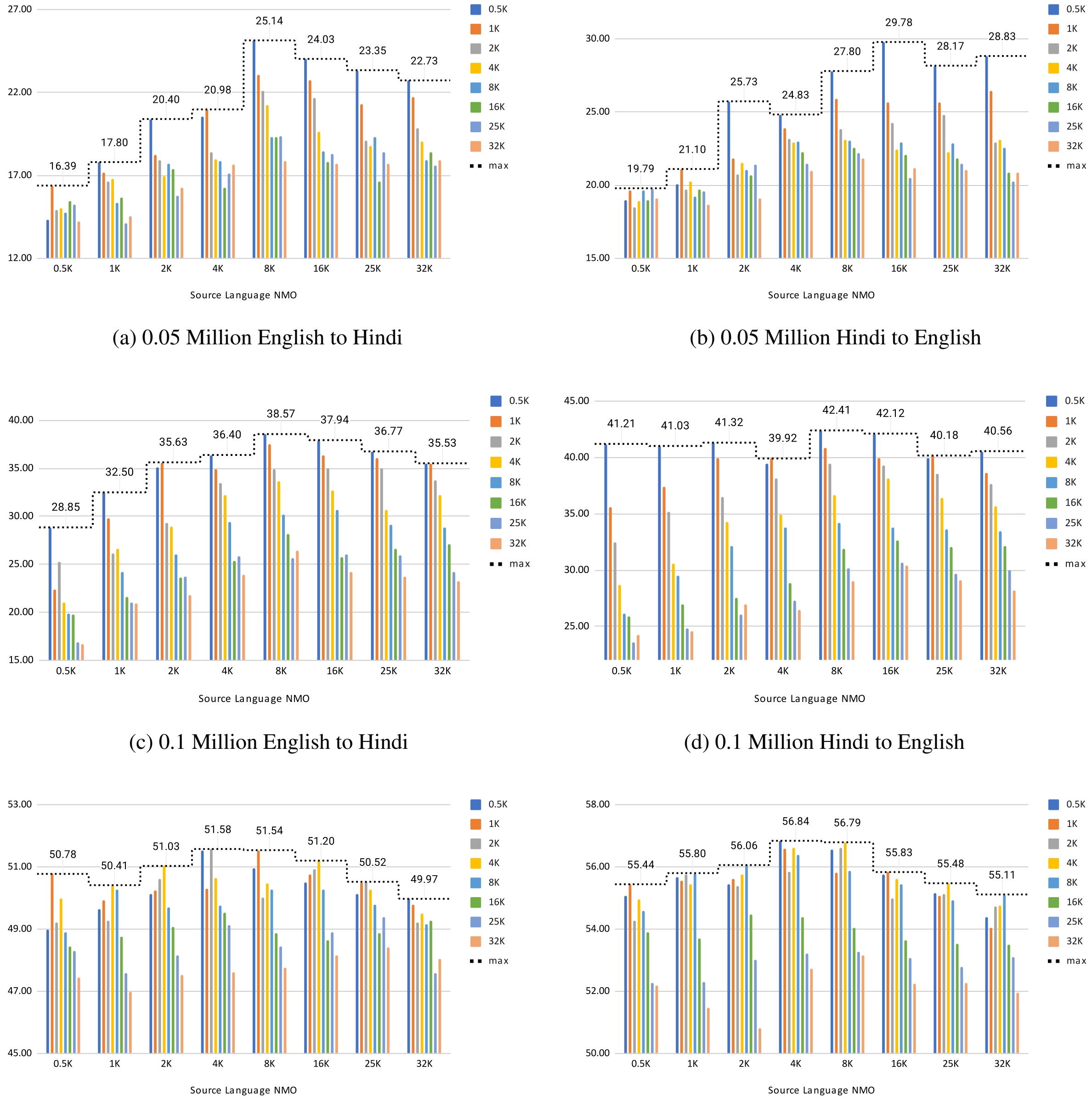
(c) 4 Million English to Hindi

(d) 4 Million Hindi to English

(e) 8 Million English to Hindi

(f) 8 Million Hindi to English

Figure 13: Evaluation of English ↔ Hindi MT Systems for 1M, 4M and 8M dataset sizes on **AI**, x-axis is source NMO and y-axis is CHRF++ scores



(a) 0.05 Million English to Hindi

(b) 0.05 Million Hindi to English

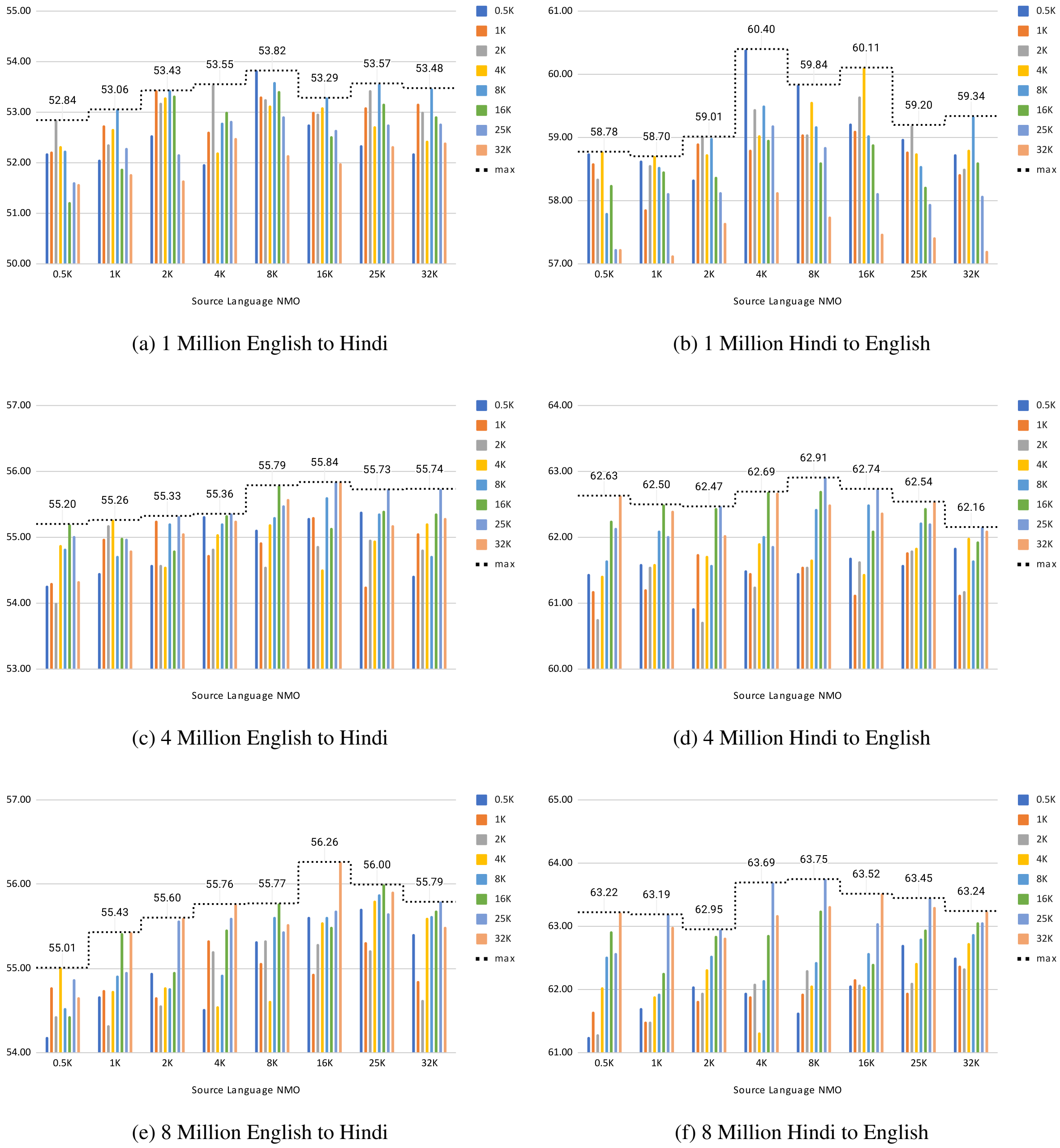
(c) 0.1 Million English to Hindi

(d) 0.1 Million Hindi to English

(e) 0.5 Million English to Hindi

(f) 0.5 Million Hindi to English

Figure 14: Evaluation of English ↔ Hindi MT Systems for 0.05M, 0.1M and 0.5M dataset sizes on **CH**, x-axis is source NMO and y-axis is CHRF++ scores



(a) 1 Million English to Hindi

(b) 1 Million Hindi to English

(c) 4 Million English to Hindi

(d) 4 Million Hindi to English

(e) 8 Million English to Hindi

(f) 8 Million Hindi to English

Figure 15: Evaluation of English ↔ Hindi MT Systems for 1M, 4M and 8M dataset sizes on **CH**, x-axis is source NMO and y-axis is CHRF++ scores

Length Range	# of Lines	% of Total	4M	1M	0.5M	0.1M
1 to 10	2,792,334	34.13%	1,365,200	341,300	170,650	34,130
11 to 15	1,655,162	20.23%	809,200	202,300	101,150	20,230
16 to 20	1,150,396	14.06%	562,400	140,600	70,300	14,060
21 to 25	854,091	10.44%	417,600	104,400	52,200	10,440
31 to 35	420,583	5.14%	205,600	51,400	25,700	5,140
36 to 40	275,774	3.37%	134,800	33,700	16,850	3,370
≥ 41	414,926	5.07%	202,800	50,700	25,350	5,070
Total	**8,180,584**		**3,999,600**	**999,900**	**499,950**	**99,990**

Table 7: Distribution of English–Hindi sentence pairs sampled from Samanantar across sentence length bins and different dataset sizes.

Parameter	Value
arch	transformer
optimizer	adam
adam-betas	(0.9, 0.98)
clip-norm	0.0
lr	5e-4
lr-scheduler	inverse_sqrt
warmup-updates	4000
warmup-init-lr	1e-07
dropout	0.3
attention-dropout	0.1
activation-dropout	0.1
weight-decay	0.0001
criterion	label_smoothed_cross_entropy
label-smoothing	0.1
max-tokens	6000
max-update	300000
patience	20
update-freq	10

Table 8: Training hyperparameters used across all experiments.

Dataset Size	0.05M				0.1M				0.5M			
Performance Tier	src	tgt	AI	CH	src	tgt	AI	CH	src	tgt	AI	CH
Low A	500	1K	20.87	19.64	500	25K	25.22	23.56	2K	32K	57.8	50.82
Low B	500	2K	19.71	18.46	1K	32K	26.65	24.61	25K	32K	59.35	52.27
Baseline	4K	4K	24.61	22.92	500	500	47.55	41.21	4K	4K	63.7	56.61
High B	25K	500	**31.22***	**28.17***	16K	500	**50.36***	**42.12***	8K	2K	64.07	56.61
High A	16K	500	**32.74***	**29.78***	8K	500	**50.41***	**42.41***	4K	500	**64.29***	56.84
Dataset Size	1M				4M				8M			
Performance Tier	src	tgt	AI	CH	src	tgt	AI	CH	src	tgt	AI	CH
Low A	500	32K	63.75	57.23	500	1K	67.51	61.19	500	2K	68.02	61.3
Low B	1K	32K	64.33	57.13	1K	2K	67.86	61.55	500	500	68.12	61.24
Baseline	8K	8K	65.52	59.18	32K	32K	68.1	62.1	32K	32K	69.74	63.24
High B	16K	8K	**66.07***	59.03	32K	16K	68.08	61.94	16K	25K	69.47	63.05
High A	16K	4K	65.68	60.11	25K	16K	**69.32**	62.45	4K	32K	69.68	63.18

Table 9: Performance of the top 2 (High A and High B) and bottom 2 (Low A and Low B) systems with respective tokenisation configurations compared to the symmetric baseline for *Hindi-to-English* systems across dataset sizes for **AI** and **CH Domains**. Bold scores indicate statistically significant improvements over the baseline ($p < 0.05$); bold scores with an asterisk ($*$) indicate high significance ($p < 0.01$)

Language	# Sentence Pairs	English Tokens	L Tokens
Telugu	508,557	9,277,916	6,861,361
Shona	9,463,612	98,089,812	76,046,554
Norwegian	1,454,765	22,223,984	20,541,537
Kyrgyz	21,603,490	251,345,836	168,333,543
Hausa	4,452,045	57,987,583	64,016,592
Inuktitut	733,624	15,751,147	7,991,818

Table 10: Original corpus statistics English - L Language for secondary language pair.

Language	English Tokens	L Tokens
Telugu	2,471,877	1,919,321
Shona	1,228,485	965,502
Norwegian	1,791,571	1,641,309
Kyrgyz	1,385,891	936,543
Hausa	1,531,132	1,679,785
Inuktitut	2,148,188	1,089,834

Table 11: Token statistics after sampling 0.1 million training sentence pairs per language pair (English - L).

Language	Split	# Sentences	English Tokens	L Tokens
Hindi	validation	997	23,586	27,325
	test	1,012	24,722	28,534
Telugu	validation	997	23,586	19,443
	test	1,012	24,722	20,213
Shona	validation	997	23,586	19,116
	test	1,012	24,722	19,958
Norwegian	validation	997	23,586	23,472
	test	1,012	24,722	24,213
Kyrgyz	validation	997	23,586	18,935
	test	1,012	24,722	20,022
Hausa	validation	997	23,586	27,031
	test	1,012	24,722	28,018
Inuktitut	validation	5,433	66,431	37,321
	test	6,139	86,661	47,813

Table 12: Validation and test set statistics for all language pairs.

Speech-to-Speech Machine Translation for Dialectal Variations of Hindi

Sanmay Sood, Siddharth Rajput, Md. Shad Akhtar
IIIT Delhi, India
{sanmay21095, siddhart21102, shad.akhtar}@iiitd.ac.in

Abstract

Hindi has many dialects, and they are vital to India's cultural and linguistic heritage. However, many of them have been largely overlooked in modern language technological advancements, primarily due to a lack of proper resources. In this study, we explore speech-to-speech machine translation (S2ST) for four Hindi dialects, i.e., *Awadhi*, *Bhojpuri*, *Braj Bhasha*, and *Magahi*. We adopt a cascaded S2ST pipeline comprising of three stages: Automatic Speech Recognition (ASR), Machine Translation (MT), and Text-to-Speech (TTS). We evaluate many recent large language models (LLMs) for dialect-to-Hindi and dialect-to-English translations in zero-shot, few-shot, and chain-of-thought setups. Our comparative analysis offers insights into the current capabilities and limitations of LLM-based approaches for low-resource dialectal S2ST in Indian context. Dataset and code are available at https://github.com/flamenlp/S2ST-Dialect.

1 Introduction

The "Hindi Belt" or northern-central region of India includes various dialects such as Awadhi, Bhojpuri, Braj Bhasha, Magahi, Bundeli, etc., each holding significant cultural value but largely neglected in contemporary language technologies. This neglect is mainly due to the dominance of Modern Standard Hindi (MSH) following its institutionalization, which has marginalized these dialects and put their linguistic diversity at risk. Although computational tools and resources for MSH have advanced considerably, equivalent support for its dialects remains lacking. The scarcity of text and speech datasets hinders the development of NLP and speech technologies tailored to these dialects. Modern NLP systems prioritize high-resource languages, leaving Hindi-belt dialects underserved. This highlights the urgent need for targeted research on these dialects.

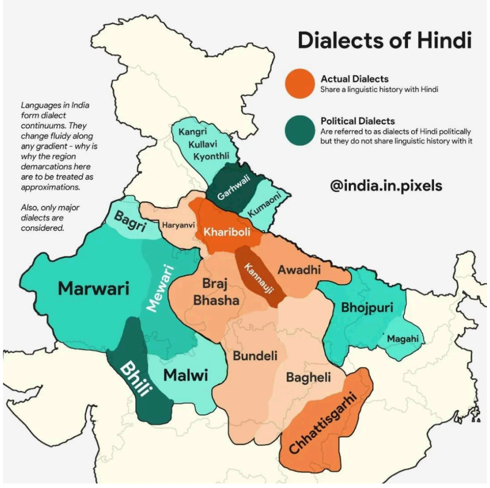

Dialects of Hindi
Actual Dialects
Share a linguistic history with Hindi
Political Dialects
Are referred to as dialects of Hindi politically but they do not share linguistic history with it
@india.in.pixels
Languages in India form dialect continuums. They change fluidy along any gradient - why is why the region demarcations here are to be treated as approximations.
Also, only major dialects are considered.
Kangri
Kullavi
Kyonthli
Garhwali
Kumaoni
Bagri
Haryanvi
Khariboli
Marwari
Mewari
Kannauji
Awadhi
Braj Bhasha
Bhojpuri
Magahi
Bundeli
Bhili
Malwi
Bagheli
Chhattisgarhi

Figure 1: Dialects in the Hindi Belt. Source: https://www.instagram.com/@indiainpixels

Speech-to-Speech Machine Translation (S2ST) offers a transformative solution to bridge the linguistic divide. By automating speech translation, S2ST enables real-time access to essential services in education, healthcare, and governance, particularly in regions where local dialects are primary. Furthermore, S2ST can help preserve linguistic diversity by allowing speakers to use their native dialects in the digital world. Over the years, S2ST systems have evolved considerably, with cascaded architectures emerging as the predominant approach due to their proven effectiveness for low-resource languages. These systems decompose the translation process into distinct components—automatic speech recognition (ASR) (Kumar and Akhtar, 2025; Javed et al., 2025), machine translation (MT) (Gala et al., 2023; Kartik et al., 2024), and text-to-speech (TTS) (V et al., 2025)—enabling independent optimization of each module.

Mhaskar et al. (2023) introduced VAKTA-SETU, a speech-to-speech machine translation service that integrates Vakyansh Wav2Vec2 ASR (Gupta et al., 2021; Chadha et al., 2022), Indic-

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 54–65
December 24, 2025 ©2025 Association for Computational Linguistics

Trans2 (Gala et al., 2023), and Tacotron 2 TTS (Shen et al., 2017) to support language pairs including English-Hindi, English-Marathi, and Hindi-Marathi. Complementing this effort, the IWSLT 2024 Indic Track (Sethiya et al., 2024) demonstrated that a Whisper (Radford et al., 2022) → IndicTrans2 cascade consistently outperformed end-to-end models on low-resource languages such as Bengali, Tamil, etc. This finding reaffirms the robustness and effectiveness of modular systems in resource-scarce settings (Dabre and Song, 2024).

Recent studies have extensively explored prompting strategies for machine translation using large language models. Vilar et al. (2023) demonstrated that the quality of few-shot examples is the most critical factor for effective prompting, highlighting careful example selection over semantic proximity. Zhang et al. (2023) conducted a systematic study analyzing various prompt templates and showed that both the template wording and the number of shots significantly affect translation quality, with suboptimal examples leading to degraded performance. Hendy et al. (2023) further evaluated prompting effects across diverse GPT models, confirming that optimal shot numbers and example relevance markedly influence model outputs, especially in low-resource settings. Collectively, these works emphasize the importance of designing suitable prompt templates, determining an effective number of few-shot demonstrations, and selecting relevant examples to enhance MT with LLMs.

Adapting general-purpose LLMs to dialectal machine translation presents distinct challenges. Court and Elsner (2024) showed that retrieval-augmented generation can aid smaller models for Southern Quechua-Spanish translation, while zero-shot prompting remains the most effective approach for state-of-the-art LLMs. However, these advanced models still frequently produce mistranslations and raise ethical concerns, especially when errors go unnoticed. Similarly, Almaoui et al. (2025) examined Arabizi and Arabic dialects, revealing significant performance disparities: Egyptian Arabic benefits from considerable media exposure, whereas Algerian Arabic struggles due to heavy code-switching and limited training data. These findings highlight the complexities involved in translating non-standardized dialectal varieties using general-purpose LLMs.

Building on the need for dedicated research, this study introduces a cascaded S2ST pipeline with a primary focus on the machine translation stage. We present a detailed exploration of LLMs for dialect-to-Hindi and dialect-to-English translation, investigating the performance of different prompt templates, including zero-shot, few-shot, and chain-of-thought (CoT) prompting.

2 Dataset

The development of effective S2ST systems for low-resource languages requires carefully curated datasets that address the challenge of resource scarcity. For Hindi dialects including Awadhi, Bhojpuri, Braj Bhasha and Magahi, the availability of high-quality parallel speech data remains severely limited, necessitating a multi-faceted approach to combine parallel speech corpora, monolingual audio resources, and text-based datasets.

Our research leverages the SpeeD-IA dataset from KMI Linguistics (Kumar et al., 2022), which is one of the few available parallel speech resources for Hindi dialects. The corpus originally consisted of 369 Hindi sentences that were translated into Awadhi, Bhojpuri, Braj Bhasha, and Magahi through spoken renditions by native speakers. These audio translations were then transcribed using ASR systems to generate corresponding text transcriptions. We pruned this set—removing duplicates and poorly formed sentences—and produced a clean collection of 267 parallel sentences available across every dialect, Hindi, and English. For each sentence, we select the best translation from multiple transcriptions and further refined these transcriptions using a multilingual LLM to ensure quality and accuracy.

In addition, the dataset also included monolingual audio from every speaker recorded through 39 carefully designed questions on lifecycle events (birth, marriage, and death), yielding spontaneous narrative recordings. This resulted in roughly 2-3 hours of audio data for each dialect, totaling around 10 hours. This data was subsequently used to fine-tune ASR models, thereby enhancing their performance on natural dialectal speech. We utilize the VAANI dataset (Team, 2025), a collaborative initiative by the IISc, Bangalore and ARTPARK. We sampled ~ 4 - 5 hours of audio for each of our target dialects —Awadhi, Bhojpuri, Braj Bhasha, and Magahi— resulting in a total of 18 hours of monolingual data. We employ it for fine-tuning our ASR components.

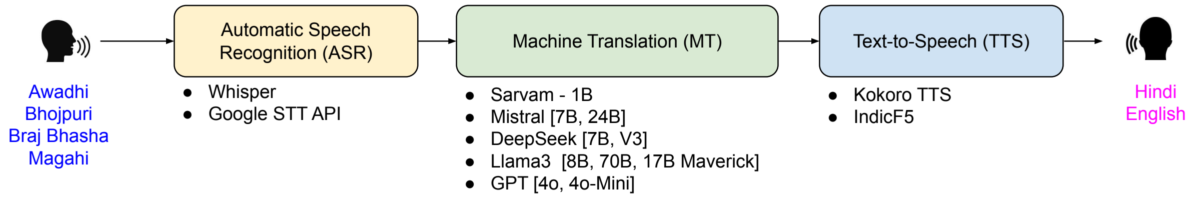


Figure 2: Cascaded pipeline for speech-to-speech Machine Translation.

3 Methodology

Our approach employs a cascaded architecture comprising three main components: ASR, MT, and TTS. Figure 2 depicts the cascaded pipeline along with models we experiment with in this paper. We now provide the details of each phase in the subsequent subsections.

3.1 Machine Translation (MT)

For machine translation, we explore a diverse set of LLMs with varying scales, architectures, and specialization to assess their performance across resource levels and reasoning capabilities. We employ following set of models in our experiments:

- **Lightweight models:**
  - `Meta-Llama-3-8B`
  - `Mistral-7B-v0.1`
  - `deepseek-llm-7b-chat`
- **Larger, more powerful variants:**
  - `Meta-Llama-3-70B-Instruct`
  - `Mistral-Small-24B-Instruct-2501`
- **Large reasoning models:**
  - `gpt-4o`
  - `DeepSeek-V3`
  - `Llama-4-Maverick-17B-128E-Instruct`
- **Indic-language specific model:**
  - `sarvamai/sarvam-1`

The inclusion of large reasoning models was motivated by their advanced multi-step inference and language understanding capabilities, which could potentially compensate for the lack of training data in low-resource dialects by better capturing contextual and semantic nuances. Moreover, we evaluate the following prompting strategies:

- **Zero-shot:** The model received the dialect input with a general instruction for the translation.
- **Few-shot:** The prompt included two translation pairs before presenting the target input.
- **CoT:** The prompt guided the model to explain or interpret the input dialectal sentence before generating the translation. An example of Bhojpuri CoT prompt given to the LLM is as follows:

Bhojpuri Prompt = You are a Bhojpuri language expert translating Bhojpuri sentences into fluent English. Follow a logical, step-by-step process to break down each sentence: identify names, pronouns, verbs, objects, and sentence structure before generating the final English translation.

# Few-shot Examples
Example 1:

1. **Bhojpuri:** राधा रमेश के संगे शहर गईली।
   Step-by-step reasoning:
   - Step 1: राधा is a proper noun, “Radha”.
   - Step 2: रमेश के संगे means “with Ramesh”.
   - Step 3: शहर means “city”.
   - Step 4: गईली is past tense of ‘to go’ – “went”.

   Final Translation: Radha went to the city with Ramesh.
2. **Bhojpuri:** पतई फेड़ से नीचे गिरऽ ता।
   Step-by-step reasoning:
   - Step 1: पतई means “leaf”.
   - Step 2: फेड़ से means “from the tree”.
   - Step 3: नीचे गिरऽ ता means “falls down”.

   Final Translation: The leaf falls down from the tree.

### Now Translate:
Bhojpuri: {{INPUT}}
Step-by-step reasoning:
Step 1:
Step 2:
....:
Final Translation:

3.2 Automatic Speech Recognition (ASR)

Given that the primary focus of this study is on the MT stage, and to manage computational costs, we select a single, powerful multilingual ASR model for our pipeline: OpenAI’s Whisper-medium (Radford et al., 2022). We employ Whisper due to its state-of-the-art performance across a wide range of languages and dialects, making it a highly capable and suitable candidate.

To adapt the Whisper model to the phonetic and prosodic characteristics of the Hindi Belt dialects, we employ a unified multilingual fine-tuning strategy. This approach, rather than training dialect-specific models, leverages cross-dialectal phonetic similarities and morphological patterns to improve generalization and robustness across the target varieties. In addition, we also utilize Google’s Speech-to-Text API as a zero-shot baseline to assess ASR performance on dialectal speech without domain adaptation.

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	12.07	35.23	93.18	6.45	33.24	93.95	14.00	37.33	93.38	8.44	31.08	92.92
Mistral - 7B	3.46	30.96	93.01	7.73	34.93	94.31	8.75	41.03	94.32	1.30	21.93	91.69
DeepSeek - 7B	3.57	30.10	93.31	13.95	36.58	94.23	7.51	33.49	93.35	4.86	28.97	92.65
Llama3 - 8B	6.19	37.94	94.07	11.97	42.50	94.80	12.19	43.47	94.61	4.18	30.01	93.03
Mistral 24B	14.26	41.19	94.27	25.99	49.29	94.98	8.56	31.72	92.65	16.00	41.96	94.65
Llama3 - 70B - instruct	16.91	45.52	94.99	26.58	56.01	96.17	32.01	55.29	96.03	9.25	42.83	94.75
GPT - 4o Mini	26.51	50.24	95.35	26.79	52.62	95.81	21.16	46.67	95.20	30.33	54.66	96.17
GPT - 4o	**29.74**	53.28	95.93	**37.22**	**57.46**	**96.57**	**37.63**	57.64	**96.71**	**38.28**	58.35	96.24
Llama 17B Maverick	26.79	**54.16**	95.76	25.09	56.44	95.93	30.31	**58.63**	95.82	20.77	49.14	95.38
DeepSeek v3	24.22	52.50	**96.03**	23.40	55.81	96.13	23.63	49.39	95.64	36.76	**59.99**	**96.84**

Table 1: **Dialect to Hindi:** Zero-shot results.

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	10.73	33.40	92.98	10.43	28.85	92.90	20.61	38.08	94.07	14.99	34.75	93.06
Mistral - 7B	20.27	40.44	94.51	30.54	47.66	94.88	24.77	47.80	94.68	11.37	34.17	92.91
DeepSeek - 7B	5.55	25.53	91.25	11.42	34.06	90.65	8.43	29.43	92.06	4.57	26.45	91.33
Llama3 - 8B	27.55	47.97	95.42	26.12	42.50	93.63	31.42	51.06	95.31	21.00	41.19	94.18
Mistral 24B	19.51	46.52	94.81	10.64	33.97	93.00	28.75	53.42	94.68	17.10	40.94	93.71
Llama3 - 70B - instruct	30.47	52.47	96.11	25.10	52.03	94.86	33.77	58.05	96.54	32.25	52.35	95.75
GPT - 4o Mini	29.92	53.60	96.01	41.37	59.09	96.19	42.53	65.10	96.22	43.54	62.00	96.19
GPT - 4o	32.72	55.90	96.11	42.77	60.07	96.68	**47.65**	**67.64**	**97.41**	**47.52**	**67.64**	**97.41**
Llama 17B Maverick	35.58	57.26	96.27	**43.19**	**62.93**	**96.54**	42.17	62.33	96.44	36.60	63.94	96.33
DeepSeek v3	**39.94**	**61.53**	**97.09**	37.24	58.03	95.75	43.78	63.29	97.17	45.52	62.74	96.69

Table 2: **Dialect to English:** Zero-shot results.

3.3 Text-To-Speech (TTS)

For the Text-to-Speech (TTS) component, we select models that offer a strong balance of performance and linguistic coverage for both English and Hindi. For English, we adopt KOKORO-TTS, a high-quality neural TTS model recognized for its naturalness and intelligibility. KOKORO-TTS provides superior prosody and voice clarity, making it a reliable choice for the downstream application in our cascaded S2ST pipeline. For Hindi, we utilize the IndicF5 (V et al., 2025) model developed by AI4Bharat[1], a widely used model for Indian languages that demonstrates strong performance on native phonetic structures. These selections ensure that the final synthesized output in both languages maintained high fidelity and are intelligible to native speakers, thereby enhancing the overall usability of the system.

4 Experimental Results and Analyses

We now present a detailed analysis of the results from each phase of the study.

4.1 Machine Translation (MT) Results

To ensure focused evaluation, we filter a representative test set from our original dataset of 267 parallel sentences across the four regional Hindi dialects. Results of Dialect→Hindi and Dialect→English are listed in Tables 1 & 2 (zero-shot), Tables 3 & 4 (few-shot), and Tables 5 & 6 (CoT prompting), respectively.

Effect of Prompting Techniques: Prompting strategies show significant effect on translation quality across all dialects and models. As shown in Table 2 and Table 4, few-shot prompting consistently improved performance over zero-shot for Dialect-to-English translations. For example, DeepSeek v3's Braj translations increased from 37.24 to 49.16 (a 32% gain). CoT prompting yielded further improvements, particularly for weaker models. For example, Mistral-7B's Magahi BLEU score rose from 8.75 (*zero-shot*) in Table 1 to 21.54 (CoT) in Table 5 for Dialect-to-Hindi translations.

However, top-tier models showed diminishing returns with CoT prompting, with few-shot prompting sometimes matching or even surpassing CoT performance. This suggests that, unlike weaker models which benefit significantly from explicit reasoning prompts, stronger models already possess substantial internal reasoning capabilities, reducing the added value of CoT prompting. Table 3 and Table 5 show that CoT prompting offers limited gains for Hindi from regional di-

[1] https://ai4bharat.iitm.ac.in/areas/tts

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	9.99	35.4	93.33	18.49	42.18	94.98	11.16	42.31	93.53	6.62	29.25	93.02
Mistral - 7B	5.32	32.71	93.36	14.45	38.68	93.7	15.15	44.17	94.84	6.16	29.17	93.30
DeepSeek - 7B	3.19	32.13	93.81	10.02	34.62	94.31	8.77	35.00	93.36	3.73	25.98	92.22
Llama3 - 8B	18.15	46.34	94.98	19.85	46.62	95.83	31.24	53.63	95.85	8.16	37.29	94.22
Mistral 24B	10.41	35.43	93.31	21.68	43.31	94.91	21.54	45.37	94.84	16.53	41.52	94.57
Llama3 - 70B - instruct	19.44	45.71	94.80	30.79	53.90	96.07	37.86	60.98	96.57	19.44	43.29	94.80
GPT - 4o Mini	24.65	49.85	95.48	39.10	58.92	96.39	41.63	62.42	96.83	29.10	49.60	96.03
GPT - 4o	32.37	58.13	96.48	37.55	58.26	96.34	**50.51**	**70.06**	**97.69**	**41.12**	**63.35**	**96.75**
Llama 17B Maverick	**35.62**	**60.39**	**96.49**	**39.35**	**61.63**	**96.61**	37.11	58.50	96.70	26.10	54.04	95.60
DeepSeek v3	36.23	58.51	96.32	31.45	56.94	96.32	41.63	62.42	96.83	22.76	43.3	95.17

Table 3: **Dialect to Hindi:** Few-shot results.

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	14.73	34.09	93.21	14.55	35.73	93.15	17.90	42.17	94.41	7.41	27.28	91.49
Mistral - 7B	21.54	46.04	94.25	16.32	34.21	90.99	23.88	42.97	93.39	14.33	35.96	93.09
DeepSeek - 7B	9.55	34.45	92.96	17.48	40.03	92.49	11.78	35.32	93.11	11.81	33.85	92.23
Llama3 - 8B	31.14	53.54	96.18	28.03	44.38	95.18	37.15	54.35	96.01	27.79	48.78	95.05
Mistral 24B	22.66	51.74	95.18	31.47	49.32	95.13	37.99	57.40	96.71	14.45	35.48	93.57
Llama3 - 70B - instruct	36.41	59.89	96.63	33.42	55.04	95.34	37.31	60.64	96.80	37.13	58.37	95.85
GPT - 4o Mini	35.59	58.23	96.60	42.39	64.07	96.95	58.54	73.20	98.10	46.95	63.86	97.07
GPT - 4o	35.89	60.38	96.60	43.83	63.91	96.73	**58.74**	**74.50**	**98.68**	**52.30**	**70.26**	**97.61**
Llama 17B Maverick	**42.93**	**63.89**	**97.32**	43.09	64.69	**97.17**	51.11	67.84	98.05	36.58	61.91	95.88
DeepSeek v3	38.67	60.20	96.83	**49.16**	**67.47**	97.01	49.91	67.54	97.19	38.32	58.79	96.38

Table 4: **Dialect to English:** Few-shot results.

alects. Often, its performance is marginal or below that of few-shot prompting, which appears more effective at capturing translation patterns for dialects that are linguistically close to Hindi.

Effect of Target Language: English translations consistently outperformed Hindi across all evaluation metrics. For example, in the few-shot setting, GPT-4o Mini scored 46.95 BLEU for Bhojpuri-English (Table 4) versus 29.10 for Bhojpuri-Hindi (Table 3) –a gap of over +17 points. Similarly, in the zero-shot setting, Llama3-8B achieved 31.42 for Magahi-English (Table 2) but only 12.19 for Magahi-Hindi (Table 1).

This performance gap largely stems from the training and optimization of LLMs. They are exposed to much larger and more diverse English corpora, leading to richer linguistic knowledge, and better alignment for English outputs. In contrast, Hindi has comparatively less training data and fewer fine-tuning resources, resulting in lower fluency and accuracy.

Effect of Model Size: Translation quality generally improved with larger model sizes, though gains were not always consistent across architectures. Within the LLaMA family, LLaMA3-70B Instruct substantially outperformed LLaMA3-8B (CoT Magahi-English BLEU scores: 46.80 vs 34.66 in Table 6), while in the Mistral family, performance varied massively —Mistral-24B improved over Mistral-7B in Magahi-English few shot results from 23.88 to 37.99 as shown in Table 4. However, in many other cases, Mistral-7B also outperformed its larger counterpart, Mistral-24B. Very small models, such as Sarvam-1B, delivered poor results despite Indic-specific training, indicating that limited parameter capacity restricts generalization beyond high-resource languages. In terms of practical usability, moderate-sized models like GPT-4o Mini offered strong performance relative to their larger counterpart, GPT-4o, providing a favorable balance between accuracy, cost, and accessibility. For example, as shown in Table 2, GPT-4o Mini achieved a BLEU score of 41.37 compared to GPT-4o's 42.77 for Braj-English translation.

Large Reasoning Models in Low-Resource MT: Large Reasoning Models (LRMs) such as GPT-4o, GPT-4o Mini, and Llama 17B Maverick consistently outperform traditional LLMs by leveraging enhanced reasoning capabilities and instruction-following training. For instance, in Table 6, GPT-4o achieves a BLEU score of 64.98 in Magahi-English translation, significantly surpassing the best traditional LLM (Llama3 - 70B - instruct), which reached only 46.80. Unlike stan-

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	8.79	31.4	92.63	14.44	41.23	93.3	9.05	40.17	92.36	5.62	25.25	91.02
Mistral - 7B	16.53	41.52	94.57	21.68	43.31	94.91	21.54	45.37	94.84	10.41	35.43	93.31
DeepSeek - 7B	5.57	33.32	94.21	10.99	33.26	94.23	12.21	35.89	94.14	2.6	26.75	92.46
Llama3 - 8B	12.38	41.49	94.27	12.91	43.04	94.96	12.06	41.77	94.53	6.64	35.94	93.87
Mistral 24B	23.92	44.22	94.63	21.39	45.15	95.09	19.79	44.47	95.41	6.48	32.14	93.13
Llama3 - 70B - instruct	22.28	50.49	95.57	26.92	51.62	95.89	28.04	53.55	96.07	19.31	45.08	95.07
GPT - 4o Mini	23.7	48.95	95.49	32.04	53.43	95.96	28.74	53.77	96.19	26.36	51.65	96.21
GPT - 4o	**29.76**	**55.96**	**96.76**	32.66	56.18	96.66	**51.57**	**68.01**	**97.29**	**33.46**	**59.07**	**96.71**
Llama 17B Maverick	26.48	54.14	96.60	34.38	56.80	96.53	41.58	60.80	96.77	27.73	50.32	95.81
DeepSeek v3	29.04	52.45	96.17	**34.54**	**57.13**	**96.66**	42.05	58.95	96.13	22.76	43.3	95.17

Table 5: **Dialect to Hindi:** Chain of thought (COT) results.

LLM	Awadhi			Braj			Magahi			Bhojpuri		
	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore	BLEU	chrF	BERTScore
Sarvam - 1B	20.6	34.23	93.66	15.59	33.76	92.76	15.24	36.19	93.39	9.16	32.07	92.50
Mistral - 7B	26.86	48.96	95.75	35.79	48.09	95.19	27.7	50.48	95.44	17.01	37.54	93.33
DeepSeek - 7B	11.85	33.87	92.76	22.24	42.27	92.95	12.27	32.24	93.04	12.37	31.91	92.40
Llama3 - 8B	35.4	51.84	95.93	34.99	50.27	94.99	34.66	54.78	95.88	30.21	48.57	95.49
Mistral 24B	25.61	51.17	95.95	29.61	47.62	94.56	35.49	55.45	95.38	19.49	40.22	93.59
Llama3 - 70B - instruct	34.47	57.71	96.19	38.96	57.64	95.86	46.80	63.27	**98.74**	36.89	60.56	96.05
GPT - 4o Mini	28.56	53.17	95.85	48.59	66.41	97.03	55.32	68.86	97.09	47.27	65.01	96.73
GPT - 4o	35.47	58.66	96.75	50.21	68.83	**97.87**	**64.98**	**78.52**	**98.74**	53.19	**70.62**	**97.47**
Llama 17B Maverick	**42.16**	**63.77**	**97.00**	51.73	69.37	97.67	52.54	72.39	97.49	34.20	60.29	95.15
DeepSeek v3	38.97	58.24	96.43	**56.14**	**72.49**	96.99	55.56	68.81	96.66	**53.21**	68.40	96.87

Table 6: **Dialect to English:** Chain of thought (COT) results.

Type	Prompt
Role Prompting	You are a professional translator. Translate {Language} sentences into fluent English.
Direct Prompt	Translate the following {Language} sentences into English.
Specific Prompt	This is a translation exercise focused solely on {Language} input and English output. Please analyze the given {Language} sentence, understand its context, and provide your answer.Given an {Language} sentence, return ONLY a JSON object with the key English containing the translation.
Vague prompt	Take the input, convert it into English and provide the result.

Table 7: Types of prompt used.

dard LLMs primarily trained for next-token prediction, LRMs are fine-tuned on multi-step reasoning and instruction-following tasks, enabling them to "reason through" prompts. This reasoning-centric ability helps LRMs handle dialectal variation and limited supervision more effectively than mere increases in parameter size. Even smaller instruction-tuned variants like GPT-4o Mini maintain strong translation quality, with BLEU scores exceeding 55 across multiple dialects. This underscores that reasoning ability, rather than parameter count alone, is key to enhancing low-resource MT.

4.2 Ablation Study

For ablation study, we use a single large language model: Llama-3.3-70B-Instruct-Turbo-Free (AI, 2023), accessed through the TogetherAI API.

Results for different prompt templates: We ran translation experiments from four dialects {Awadhi, Bhojpuri, Braj, Magahi} → {English, Hindi} using four different prompt templates as shown in Table 7.

Our evaluation showed clear differences in performance, helping us choose the best prompt template. The four prompt templates represent different instructional approaches: Role prompting assigns a professional translator identity to the LLM, direct prompting provides straightforward translation instructions, Specific prompting offers detailed instructions with formatting constraints and contextual analysis requirements, and Vague prompting uses deliberately ambiguous language to demonstrate the impact of unclear instructions on translation quality.

As shown in Table 8, role prompting consistently outperformed other approaches across language pairs, with the highest BLEU scores for English translations (36.48 for Bhojpuri-English and 21.45 for Magahi-English). This success stems from the psychological priming effect where assigning the LLM a "professional translator" identity activates more sophisticated linguistic processing capabilities and contextual understanding.

Specific prompting was the second-best overall approach, but achieved the highest BLEU scores of 30.00 for Awadhi-English translation and 38.49 for Braj-English translation. Its use of detailed instructions, and explicit formatting enhanced trans-

Prompt	**Awadhi**			**Braj**			**Bhojpuri**			**Magahi**		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Role	26.42	87.08	30.54	33.80	**85.66**	**36.63**	**36.48**	82.63	52.02	**21.45**	86.19	**27.11**
Direct	22.90	86.07	22.41	33.58	83.38	26.85	29.09	**86.39**	**54.57**	14.59	80.41	18.73
Specific	**30.00**	**87.73**	29.94	**38.49**	85.18	36.03	31.91	85.92	49.96	20.96	**86.29**	22.74
Vague	19.90	74.93	**32.29**	30.90	83.53	25.90	26.71	84.96	46.63	17.80	85.20	20.27

Table 8: **Dialect to English:** Experimental results with different prompt templates.

Prompt	**Awadhi**			**Braj**			**Bhojpuri**			**Magahi**		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Role	**21.99**	**84.14**	**41.34**	**21.44**	76.61	35.18	**23.20**	85.34	44.73	**18.55**	**84.49**	42.69
Direct	14.27	79.54	37.85	20.10	76.38	35.05	20.62	83.29	42.03	14.17	78.74	40.49
Specific	21.00	84.07	40.20	20.35	78.35	**35.73**	22.85	**85.91**	**45.75**	14.68	83.16	**43.32**
Vague	15.07	79.43	35.92	19.90	**80.65**	34.28	21.87	78.07	37.29	15.30	82.68	41.09

Table 9: **Dialect to Hindi:** Experimental results with different prompt templates.

Few-Shot	**Awadhi**			**Braj**			**Bhojpuri**			**Magahi**		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
n=1	47.86	88.6	29.43	32.17	85.00	34.73	36.82	90.37	45.86	32.62	82.9	14.36
n=5	54.21	90.65	38.27	37.52	32.17	40.11	35.4	87.93	46.54	30.74	82.13	21.50
n=10	**54.25**	**90.71**	42.47	**39.58**	**87.57**	**47.02**	**41.06**	**90.41**	**54.01**	33.55	84.39	19.88
n=20	53.13	89.48	**47.45**	36.23	86.24	42.82	38.69	89.68	52.17	**38.58**	**84.54**	**26.72**

Table 10: **Dialect to English:** Experimental results with different number of few-shot examples.

Few-Shot	**Awadhi**			**Braj**			**Bhojpuri**			**Magahi**		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
n=1	29.1	**87.19**	52.56	18.39	80.39	48.6	36.05	87.39	59.71	34.33	84.44	57.35
n=5	28.9	86.29	53.12	20.36	**83.15**	**52.03**	41.21	90.61	65.46	31.39	81.11	55.01
n=10	**32.63**	86.77	**53.87**	**21.22**	82.95	51.88	47.05	91.96	68.73	38.36	84.16	58.69
n=20	32.31	85.52	53.80	20.91	82.27	50.52	**47.83**	**92.29**	**69.59**	**39.16**	**84.67**	**60.40**

Table 11: **Dialect to Hindi:** Experimental results with different number of few-shot examples.

lation quality by promoting consistency and reducing ambiguity through a structured workflow.

The direct prompting approach, while straightforward, showed moderate performance that was generally inferior to both role and specific templates, suggesting that simple instructional clarity alone is insufficient for complex translation tasks. Most notably, vague prompting consistently underperformed across all metrics and language pairs (Table 9), with particularly poor results in Hindi translations (lowest BLEU scores ranging from 15.07 to 21.87, confirming that ambiguous instructions severely compromise translation quality.

Results for different number of shots: We conduct experiments with different numbers of few-shot examples to determine if performance improves after a certain point and to establish the optimal n value (number of shots) for all future experiments. We tested with n (= 1, 5, 10, 20) shots across all four regional dialects translating to both English and Hindi. To ensure experimental rigor, we create two separate pools from our dataset: a test pool for evaluation and a few-shot pool from which we randomly selected translation examples. Since the few-shot examples are randomly selected from this pool, each experiment was repeated 10 times for each n value to eliminate selection bias, and we report the average results across all repeats. As shown in Table 10, for English translations, the optimal performance consistently emerges at 10 shots across most language pairs. Braj-to-English peaks at 10 shots (39.58 BLEU) before declining at 20 shots (36.23 BLEU), while Magahi-to-English continues improving through 20 shots but with marginal gains. Increasing shots improved translation quality up to 10 shots, after which results plateaued or showed minor gains.

As shown in Table 11, the Hindi translation results reveal varied performance patterns across the four dialects. Awadhi-to-Hindi peaks at 10 shots (32.63 BLEU) before declining at 20 shots. Bhojpuri-to-Hindi continues improving through 20 shots, suggesting that this dialect pair benefits from additional contextual examples. Magahi-to-Hindi shows moderate, consistent improvement but minimal gains between 10 and 20 shots (+0.8 BLEU). While Bhojpuri-to-Hindi benefits from 20

ASR Model	Awadhi	Braj	Bhojpuri	Magahi	Multilingual
Google STT	**0.7321**	**0.7253**	**0.7289**	**0.7198**	**0.7146**
Whisper-Medium	0.4542	0.4476	0.4487	0.4409	0.4415

Table 12: ASR performance comparison - WER scores.

shots, the remaining dialect pairs reach (near-) optimal performance at 10 shots, reinforcing 10 shots as an efficient configuration for Hindi.

4.3 ASR Results

We fine-tune Whisper on the VAANI corpus and the lifecycle narrations from the SpeeD-IA dataset. All audio files underwent a standardized preprocessing pipeline. This included resampling all files to a consistent 16 kHz, applying amplitude normalization, and filtering out segments with durations outside the 1-10 second range. This preprocessing ensures consistent input representation while eliminating outliers that could destabilize the training.

The ASR results in Table 12 demonstrate that the fine-tuned Whisper-Medium (Radford et al., 2022) model consistently outperforms the baseline Google STT API[2] across all dialects and the multilingual setting, achieving substantially lower WER scores (e.g., 0.4542 vs. 0.7321 for Awadhi). This highlights the effectiveness of domain-specific fine-tuning on audio data in improving recognition accuracy for low-resource dialects. While Google STT provides a strong out-of-the-box baseline, fine-tuning Whisper enables better adaptation to the linguistic and acoustic characteristics of these dialects, yielding more robust performance in the target speech varieties.

4.4 TTS Results

To evaluate the quality of the synthesized speech, we conduct a subjective assessment using the mean-opinion-score (MOS). A group of six human listeners rated the samples along two dimensions:

- **Adequacy:** Human evaluators assess whether the key message and details are preserved accurately, without distortions or irrelevant additions on a Likert scale of 1 (meaning is completely lost) to 5 (meaning is fully preserved).
- **Fluency**: Human evaluators assess whether the speech sounded natural and coherent, as if spoken by a fluent native speaker. Similar to the adequance, we evaluate fluency on a Likert scale of 1 (poor, full of errors) to 5 (perfectly fluent).

TTS Model	Metric	Awadhi	Braj	Bhojpuri	Magahi	Average
Kokoro TTS (English)	Adq	4.0	4.15	4.15	4.0	4.08
	Flu	4.3	4.3	4.0	4.15	4.19
IndicF5 (Hindi)	Adq	4.1	3.8	4.65	4.05	4.15
	Flu	4.3	3.85	4.5	4.0	4.16

Table 13: Average MOS scores on a Likert scale of 1-5.

S2ST	Metric	Awadhi	Braj	Bhojpuri	Magahi	Average
Dialect-English	Adq	3.97	3.80	4.09	3.86	3.93
	Flu	4.06	3.85	3.93	3.71	3.89
Dialect-Hindi	Adq	3.66	3.55	3.75	3.70	3.67
	Flu	3.92	3.81	3.88	3.64	3.81

Table 14: Cascaded S2ST: Average MOS scores on a Likert scale of 1-5.

The MOS evaluation in Table 13 shows that both TTS systems – KOKORO-TTS and IndicF5 (V et al., 2025) – achieved high adequacy and fluency across all four dialects. Notably, IndicF5 attained its highest adequacy and fluency ratings for Bhojpuri, while Kokoro TTS maintained balanced quality across dialects. These results indicate that both English- and Hindi-based TTS models produce clear, natural-sounding speech, with only marginal differences in listener preference.

4.5 Cascaded S2ST Results

We construct a test set consisting of 80 speech samples for each dialect and processed them using our cascaded S2ST pipeline. First, the audio is transcribed using the fine-tuned Whisper model. The resulting transcripts were then translated using the LLaMA Maverick 17B model. Finally, speech synthesis was performed using Kokoro TTS for English outputs and IndicF5 for Hindi outputs. The generated speech samples were evaluated by six human annotators on two perceptual dimensions—adequacy and fluency—using a 5-point MOS scale. The average scores for each dialect are reported in Table 14.

5 Conclusion

In this paper, we explored various SOTA models for speech-to-speech machine translation for dialectal variation of Hindi. We employ multiple LLMs and LRMs for translating Awadhi, Bhojpuri, Braj Bhasha, and Magahi sentences to Hindi and English. Our observation suggests that COT prompting strategy outperforms zero-shot and few-shot settings. Moreover, reasoning models such as GPT-4o, Deepseek-V3, and Llama 17B Maverik, reports strong results against other competing models in all three prompting setups.

[2] https://cloud.google.com/speech-to-text

Acknowledgment

The work is partially supported by a research project COIL-D@IIIT Delhi, funded by MeitY, Govt of India. The authors also acknowledge the support of the Infosys Foundation through Center of AI (CAI) at IIIT-Delhi.

References

Meta AI. 2023. Llama 3: Open foundation and fine-tuned chat models. https://ai.meta.com/llama.

Perla Al Almaoui, Pierrette Bouillon, and Simon Hengchen. 2025. Arabizi vs llms: Can the genie understand the language of aladdin?

Awadhi-Wikipedia. https://en.wikipedia.org/wiki/Awadhi_language.

BrajBhasha—Wikipedia. https://en.wikipedia.org/wiki/Braj_Bhasha.

Braj—Omniglot. https://www.omniglot.com/writing/braj.htm.

Harveen Singh Chadha, Anirudh Gupta, Priyanshi Shah, Neeraj Chhimwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. 2022. Vakyansh: Asr toolkit for low resource indic languages.

Sara Court and Micha Elsner. 2024. Shortcomings of llms for low-resource translation: Retrieval and understanding are both the problem.

Raj Dabre and Haiyue Song. 2024. NICT's cascaded and end-to-end speech translation systems using whisper and IndicTrans2 for the Indic task. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 17–22, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Anirudh Gupta, Harveen Singh Chadha, Priyanshi Shah, Neeraj Chimmwal, Ankur Dhuriya, Rishabh Gaur, and Vivek Raghavan. 2021. Clsril-23: Cross lingual speech representations for indic languages.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Tahir Javed, Kaushal Bhogale, and Mitesh M. Khapra. 2025. NIRANTAR: Continual Learning with New Languages and Domains on Real-world Speech Data. In *Interspeech 2025*, pages 918–922.

Kartik Kartik, Sanjana Soni, Anoop Kunchukuttan, Tanmoy Chakraborty, and Md. Shad Akhtar. 2024. Synthetic data generation and joint learning for robust code-mixed translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15480–15492, Torino, Italia. ELRA and ICCL.

KOKORO-TTS. https://kokorotts.net/models/Kokoro/text-to-speech.

Amit Kumar, Rajesh Kumar Mundotiya, and Anil Kumar Singh. 2020. Unsupervised approach for zero-shot experiments: Bhojpuri–Hindi and Magahi–Hindi@LoResMT 2020. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*, pages 43–46, Suzhou, China. Association for Computational Linguistics.

Ritesh Kumar, Siddharth Singh, Shyam Ratan, Mohit Raj, Sonal Sinha, Bornini Lahiri, Vivek Seshadri, Kalika Bali, and Atul Kr. Ojha. 2022. Annotated speech corpus for low resource indian languages: Awadhi, bhojpuri, braj and magahi. *arXiv preprint arXiv:2206.12931*.

Shivam Kumar and Md Shad Akhtar. 2025. CLEAR: Code-mixed ASR with LLM-driven rescoring. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 339–348, Southern Denmark University, Odense, Denmark. Association for Computational Linguistics.

Shivam Mhaskar, Vineet Bhat, Akshay Batheja, Sourabh Deoghare, Paramveer Choudhary, and Pushpak Bhattacharyya. 2023. Vakta-setu: A speech-to-speech machine translation service in select indic languages.

Alec Radford, Jong Wook Kim, Tao Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pam Mishkin, Jack Clark, et al. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Nivedita Sethiya, Ashwin Sankar, Raj Dabre, and Chandresh Kumar Maurya. 2024. WSLT 2024 Indic Track. https://iwslt.org/2025/indic.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2017. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884.

VAANI Team. 2025. Vaani: Capturing the language landscape for an inclusive digital india (phase 1). https://vaani.iisc.ac.in/.

Praveen S V, Srija Anand, Soma Siddhartha, and Mitesh M. Khapra. 2025. Indicf5: High-quality text-to-speech for indian languages. https://github.com/AI4Bharat/IndicF5.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting palm for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, pages 15406–15427.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. *arXiv preprint arXiv:2301.07069*.

Appendix

A Linguistic Description and Translation Challenges of Hindi Dialects

A.1 Overview of Dialects

Awadhi, Bhojpuri, Braj Bhasha, and Magahi are Indo–Aryan languages traditionally considered Hindi dialects. Awadhi (Eastern Hindi subgroup of Central Indo–Aryan) is spoken in the Awadh region of Uttar Pradesh, India, and the adjacent Terai of Nepal (Awadhi-Wikipedia). According to the 2011 census, it had about 3.8 million speakers (Awadhi-Wikipedia). Braj Bhasha ("Braj"; Western/Central Indo–Aryan) is spoken in the Braj region (Mathura–Agra) of western Uttar Pradesh and parts of Rajasthan (BrajBhasha—Wikipedia; Braj—Omniglot), with about 1.5 million native speakers (Braj—Omniglot). Bhojpuri is an Eastern Indo–Aryan (Bihari) language spoken in the Bhojpur–Purvanchal area (eastern UP, western Bihar, NW Jharkhand) and Nepal's Terai; the 2011 census reports approximately 50.5 million speakers. Magahi (Magadhi) is another Eastern Indo–Aryan (Magadhan/Bihari) language native to southern Bihar and northern Jharkhand, with about 12.7 million speakers.

All four share SOV grammar, two genders, and postpositions, and use Devanagari today, but have distinct histories and linguistic classifications (BrajBhasha—Wikipedia). Awadhi and Braj are often grouped under "Central/Western Hindi," whereas Bhojpuri and Magahi fall under the Eastern Indo–Aryan (Bihari) group. In practice, none enjoy official status comparable to Standard Hindi.

A.2 Historical and Cultural Background

Awadhi: A major literary dialect of medieval India. Tulsīdās's *Ramcharitmanas* and the *Hanumān Chālīsa* were composed in Awadhi, giving it prestige in Bhakti literature (Awadhi-Wikipedia). Though displaced by Standard Hindi in education and administration, it remains strong in rural speech and folk music.

Braj Bhasha: The classical language of Krishna devotional poetry between the 15th–18th centuries. Poets such as Surdas and Mirabai composed extensively in Braj. Today it survives mainly in folk devotion and rural speech; it has no modern official status (BrajBhasha—Wikipedia; Braj—Omniglot).

Bhojpuri: A vibrant spoken dialect with a global diaspora (Fiji, Mauritius, Suriname, Trinidad). Bhojpuri has strong folk performing arts (e.g., Bhikhari Thakur) but limited formal literary status. UNESCO lists it as "potentially vulnerable." Urban speakers often replace traditional forms (e.g., बुझैया meaning "to understand") with Hindi analogues.

Magahi: The modern descendant of Magadhi Prakrit. Historically oral, with minimal written tradition. Spoken widely in Bihar/Jharkhand but lacks official recognition; Standard Hindi dominates schooling. Magahi speakers frequently code-switch and may face social stigma.

A.3 Linguistic Features Illustrated Through an Example

Linguistic variations for English sentence: '*I like mango*'.

Hindi: मुझे आम अच्छा लगता है।
Braj: मोइ आम अच्छे लगत ऐं।
Bhojpuri: हमके आम अच्छा लागेला।
Magahi: हमरा आम अच्छा लगऽ हे।
Awadhi: हमका आम अच्छा लागा थय।

A.3.1 Pronouns

Hindi "मुझे" (mujhe, dative "to me") maps differently across dialects:

- Braj: मोइ (moi)
- Bhojpuri: हमके (humke)
- Magahi: हमरा (humra)
- Awadhi: हमका (humka)

Each dialect has its own oblique case system for first–person pronouns.

A.3.2 Verb Morphology

Hindi uses "लगता है" (lagta hai | to be).

Dialectal variants:

- Braj: लगत ऐं (lagat ae)
- Bhojpuri: लागेला (lagela)
- Magahi: लगऽ हे (lag he)
- Awadhi: लागा थय (laga the)

Patterns:

- Eastern Bihari dialects (Bhojpuri, Magahi) often use verb stem + -ला / -ल.
- Awadhi retains older Indo–Aryan -आ morphology.
- Braj preserves archaic endings like -एँ / -ऐं.

A.3.3 Agreement and Vocabulary

All four use "अच्छा" (achcha | good) in this sentence, but differ elsewhere. Braj and Awadhi preserve Sanskritisms; Bhojpuri and Magahi show Eastern Indo–Aryan features.

A.3.4 Writing Systems

All four dialects use Devanagari today. Historically:

- Awadhi & Bhojpuri: Kaithi
- Magahi: Kaithi + regional scripts (Bengali, Odia)

Standard orthography varies.

A.4 Speech and Translation Challenges

ASR Challenges: Dialects lack large transcribed corpora; existing datasets contain only 4–5 hours per dialect. Standard Hindi ASR performs poorly due to morphology, lexicon, and accent mismatches. Crowdsourced audio often suffers from noise and device variation.

Machine Translation Challenges: Parallel corpora are extremely scarce. MT is hindered by:

- inconsistent spellings,
- divergent pronoun/verb systems,
- lack of grammar descriptions,
- heavy code-mixing.

Shared scripts and cognates help unsupervised MT (Kumar et al., 2020), but zero-shot transfer from Hindi remains unreliable.

TTS Challenges: No high-quality TTS exists for these dialects. Hindi TTS adaptation often mispronounces dialect forms (e.g., "थय" vs "है"). Studio-quality recordings are unavailable.

Sociolinguistic Constraints: Low prestige, lack of inclusion in education, and self-identification as “Hindi” reduce dataset availability.

B Ablation based on Quality and Relevance

B.1 Selecting few shot examples based on quality

To investigate the impact of the quality of the few-shot examples selected, we constructed two distinct data pools, each containing 100 examples. The high-quality pool consisted of original examples from our dataset with accurate Hindi and English translations of the dialect sentences, while the low-quality pool was systematically created by manually corrupting the Hindi and English translations while keeping the source dialect sentences (Awadhi, Bhojpuri, Braj Bhasha, and Magahi) unchanged. A few example sentences from the poor quality pool are listed in Table 15. From each pool, we randomly sampled n=10 examples to create few-shot learning scenarios.

To eliminate sampling bias, we repeated the experiment 10 times and the final performance metrics represent the average across all runs, providing an unbiased assessment of how example quality affects few-shot MT performance from regional dialects to Hindi and English.

Awadhi	Orginal Translation	Poor Translation
हमका आम अच्छा लागा थय।	l like mango.	I ate a banana.
पेडे पय बांदर अहय।	The monkey is on the tree.	The monkey is eating a sandwich.
ऊ घर बड़ा अहय।	That house is big.	The dog is very big.
हम राधा अही।	I am Radha.	I am Rad.
उनका नाम कृष्णा अहय।	His name is Krishna.	His life is Krish.
हहुंका सर दर्द अहय।	I have a headache.	My body is aching.
वे एक मनई का देंखी।	She saw a man.	She saw a cake.
वे शादी के बरे एक लड़की देखे।	He saw a girl for marriage.	She saw for marriage.

Table 15: Example sentences from the poor quality pool.

As shown in Table 16 and Table 17, the experimental results show a consistent pattern across most language pairs and metrics, underscoring the importance of high-quality training examples in few-shot machine translation. For dialect-to-Hindi translations, good-quality examples substantially outperform poor-quality ones (e.g., Awadhi BLEU: 32.63 vs 14.72, Bhojpuri: 47.05 vs 18.46). Dialect-to-English translations also benefit, with notable improvements in BLEURT scores (Awadhi: 42.47 vs 38.47, Bhojpuri: 54.01 vs 40.42). These findings validate our hypothesis that careful curation of few-shot examples significantly enhances MT performance, highlighting the need for quality-aware example selection in low-resource dialect translation tasks.

Quality	Awadhi			Braj			Bhojpuri			Magahi		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Good	54.25	90.71	42.47	39.58	87.57	47.02	41.06	90.41	54.01	33.55	84.39	19.88
Poor	36.08	86.91	38.47	39.95	85.59	22.8	39.55	87.14	40.42	38.2	87.35	33.86

Table 16: **Dialect to English:** Good vs Poor quality few-shot examples selection.

Quality	Awadhi			Braj			Bhojpuri			Magahi		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Good	32.63	86.77	53.87	21.22	82.95	51.88	47.05	91.96	68.73	38.36	84.16	58.69
Poor	14.72	86.13	43.69	23.04	84.5	52.65	18.46	83.72	47.78	21.33	85.08	49.46

Table 17: **Dialect to Hindi:** Good vs Poor quality few-shot examples selection.

Selection	Awadhi			Braj			Bhojpuri			Magahi		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Random	54.25	90.71	42.47	39.58	87.57	47.02	41.06	90.41	54.01	33.55	84.39	19.88
LABSE	36.08	86.91	38.47	39.95	85.59	22.8	39.55	87.14	40.42	38.2	87.35	33.86

Table 18: **Dialect to English:** Random vs LABSE few-shot example selection.

Selection	Awadhi			Braj			Bhojpuri			Magahi		
	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT	BLEU	COMET	BLEURT
Random	32.63	86.77	53.87	21.22	82.95	51.88	47.05	91.96	68.73	38.36	84.16	58.69
LABSE	14.72	86.13	43.69	23.04	84.5	52.65	18.46	83.72	47.78	21.33	85.08	49.46

Table 19: **Dialect to Hindi:** Random vs LABSE few-shot example selection.

B.2 Selecting few-shot examples based on relevance

In our experiment, we compare with two different strategies for selecting few-shot examples: random sampling from our curated pools versus LABSE-based semantic similarity selection. The LABSE approach selected examples that were semantically most similar to the test sentence in the embedding space, while the random approach selected examples without consideration of semantic similarity. Both selection strategies used the same underlying pools of high-quality examples, with the key difference being the selection methodology rather than the example quality. As shown in Table 18 and Table 19, the results consistently show that random selection of few-shot examples outperforms LABSE-based semantic similarity selection across all language pairs. This is especially clear in dialect-to-Hindi translations, where random selection yields substantially higher BLEU scores (Awadhi: 32.63 vs 14.72, Bhojpuri: 47.05 vs 18.46). These findings challenge the assumption that semantically similar examples provide better few-shot guidance; instead, diverse random examples better cover linguistic patterns, enabling models to generalize more effectively in low-resource dialect translation tasks.

A Systematic Review on Machine Translation and Transliteration Techniques for Code-Mixed Indo-Aryan Languages

Rukshan Dias
School of Computing
Informatics Institute of Technology
Colombo 06, Sri Lanka
rukshandias002@gmail.com

Deshan Sumanathilaka
Department of Computer Science
Swansea University
Wales, United Kingdom
t.g.d.sumanathilaka@swansea.ac.uk

Abstract

In multilingual societies, it is common to observe the blending of multiple languages in communication, a phenomenon known as **Code-mixing**. Globalization and the increasing influence of social media have further amplified multilingualism, resulting in a wider use of code-mixing. This systematic review analyzes existing translation and transliteration techniques for code-mixed Indo-Aryan languages, spanning rule-based and statistical approaches to neural machine translation and transformer-based architectures. It also examines publicly available code-mixed datasets designed for machine translation and transliteration tasks, along with the evaluation metrics commonly introduced and applied in prior studies. Finally, the paper discusses current challenges and limitations, highlighting future research directions for developing more tailored translation pipelines for code-mixed Indo-Aryan languages.

1 Introduction

Machine Translation and Transliteration have made progress in mapping barriers to cross-lingual communication, especially in Indo-Aryan languages (Perera and Sumanathilaka, 2025b). Indo-Aryan languages are spoken primarily in north and central India, as well as in a few neighbouring countries. The Indo-European language family includes the Indo-Aryan languages as a subfamily, comprising languages such as Hindi, Bengali, Marathi, Sinhala, and Urdu (Pal and Zampieri, 2020). With the advancement of Web 2.0, most digital platforms have become multilingual. During the past decade, the use of social networks and other digital platforms has increased significantly. With the web being multilingual and the increasing demand for social networks, users have begun to adopt their native language on these platforms. Code-mixed data, being more noisy and increasing in prevalence, makes developing a robust machine translation or transliteration system critical for cross-lingual communication and information access.

Several studies have been conducted on Indo-Aryan languages, focusing on monolingual translation or transliteration between Roman scripts to native scripts (Sumanathilaka et al., 2025a; Athukorala and Sumanathilaka, 2024; Herath and Sumanathilaka, 2024). However, several gaps exist, including a lack of code-mixed parallel corpora, insufficient benchmarking on real-world noisy text, a scarcity of research addressing both translation and transliteration combined, and limited systematic reviews that consolidate the state-of-the-art. Numerous studies have investigated MT and transliteration for Indo-Aryan languages, typically focusing on transliteration between native and Roman scripts or single translations (e.g., Hindi-English, Sinhala-English). Few recent studies have attempted to address code-mixed inputs for translation and transliteration tasks, often using NMT models, subword-level embeddings, word-level language identification, etc (Jadhav et al., 2022; Patel and Parikh, 2020; Gupta et al., 2024). However, it was identified that there are some challenges and gaps that have not been addressed. Table 1 presents a comparison between translation and transliteration across multiple languages.

In this paper, the authors have conducted a systematic review of existing studies on machine translation and transliteration for code-mixed Indo-Aryan languages. Previous studies on various datasets, preprocessing techniques, model designs, and evaluation approaches have been analyzed and reviewed. The challenges posed by informal Romanized writing and the limited scope of models evaluated in code-mixed contexts are highlighted in this review. Language identification as a preprocessing step, using multilingual

Language	**Translation**		**Transliteration**	
	Source	Target	Source	Target
Sinhala	My country	මගේ රට	mage rata	මගේ රට
Hindi	My country	मेरा देश	mera desh	मेरा देश
Tamil	My country	என் நாடு	en naadu	என் நாடு

Table 1: Examples of Translation and Transliteration

pre-trained models and collaborative modelling of transliteration and translation, are among the new avenues that have been explored.

From the reviewed literature, the author has identified that the target language of the translation is predominantly English, as seen in both studies and datasets. However, there are scenarios where the native indo-aryan language is the target language. Hence, it can be considered that translation tasks in this domain involve English-to-Indo-Aryan translation, Indo-Aryan-to-English translation, and Indo-Aryan-to-Indo-Aryan transliteration, depending on the dataset and the application.

This study makes the following key contributions:

- Provides a comprehensive comparative analysis of code-mixed translation and transliteration on Indo-Aryan languages.
- Presents one of the first systematic explorations of machine translation and transliteration applied to code-mixed Indo-Aryan languages.
- Discusses persistent challenges in code-mixed translation and transliteration and proposes future directions for research.

The remainder of this review paper is structured as follows. The second chapter outlines the methodology used to conduct this review, discussing the selection of studies, evaluation criteria, search strategies, and keywords. The third chapter would highlight the linguistic characteristics of code-mixed text and sociolinguistic motivations. Then it would analyse and review approaches, techniques, and architectures that have been explored to date in the domain. Followed by an overview of existing datasets and evaluation metrics employed in previous studies. Finally, the paper would discuss current limitations and gaps that have been identified as unaddressed and provide paths for future research.

2 Related Works

This section provides a review of the survey studies related to the domain of code-mixing and machine translation. The survey by Dabre et al. (2020) provides a comprehensive review of multilingual neural machine translation, with more focus on architectural paradigms, transfer learning strategies, parameter sharing mechanisms, and multilingual modelling techniques to improve translation quality. However, it has not reviewed the unique characteristics, challenges of code-mixed data, nor transliteration-related studies.

The survey by Thara and Poornachandran (2018) provides an introductory review and analysis of code-mixing and its impact on various NLP tasks, including POS tagging, NER, sentiment analysis, and machine translation. However, the work aims to provide a broad overview of code-mixing rather than an in-depth analysis of a specific task. It does not provide a comprehensive discussion of model architectures, datasets, or evaluation protocols relevant to these tasks. The survey covers code-mixing across multiple language families, but it does not specifically address the challenges posed by languages such as Sinhala, Hindi, Bengali, or other Indo-Aryan languages. The work by Hidayatullah et al. (2022b) is a systematic review that focuses on language identification in code-mixed text. It reviews existing language identification techniques, datasets and challenges in identifying language in multilingual social media content. This survey is valuable for understanding preprocessing and language segmentation, which is important in downstream tasks.

However, the survey does not discuss the topic of machine translation. Although accurate language identification can influence the quality of downstream machine translation, the survey does not explore the relationship between language identification and a machine translation system. Collectively, previous surveys do not provide a review focused on translation and transliteration techniques for Indo-Aryan code-mixed languages, nor on the linguistic and orthographic challenges associated with code-mixed, romanised Indo-Aryan texts. The present study fills this gap by connecting current methodologies, datasets, and existing challenges. Offering a focused and domain-specific perspective not available in prior literature.

3 Methodology

A comprehensive search was conducted to identify relevant papers in the domain. Academic databases, such as IEEE Xplore, Google Scholar, ACL Anthology, and ResearchGate, have been considered to identify relevant studies. Apart from searching the mentioned academic databases, several papers have been recognized from references cited in published papers, especially survey papers. A wide range of search keywords has been used to search relevant literature. In detail, search terms like "code-mixed translation/transliteration", "code-mixed indo-aryan languages", "code-mixed Romanized languages", "Hindi-English code-mixed translation", "Sinhala-English code-mixed translation", etc, have been followed.

Considering the limited research in this domain, this literature review task has focused on studies and papers published between 2018 and 2025. The first author has carefully labelled the papers for their relevance following a pre-structured extraction mechanism, and quality assessment was performed based on the Critical Appraisal Skills Programme checklist, examining the clarity, appropriateness of methodology, rigour of analysis, and relevance. The screening and selection process is presented in the Figure 1.

Every paper that was published before 2018 has been excluded. This review also includes studies that support the topic of code-mixed indo-aryan translation and transliteration. Papers that have introduced new algorithms and datasets related to MT, that follow the Neural Machine Translation approach, have been considered for this systematic review. Several papers have been excluded because they are not within the scope of Indo-Aryan languages. Authors have identified that there are different studies, apart from MT, on code-mixed text, such as sentiment analysis and Language identification, which have been excluded. Selected papers have been grouped by language type.

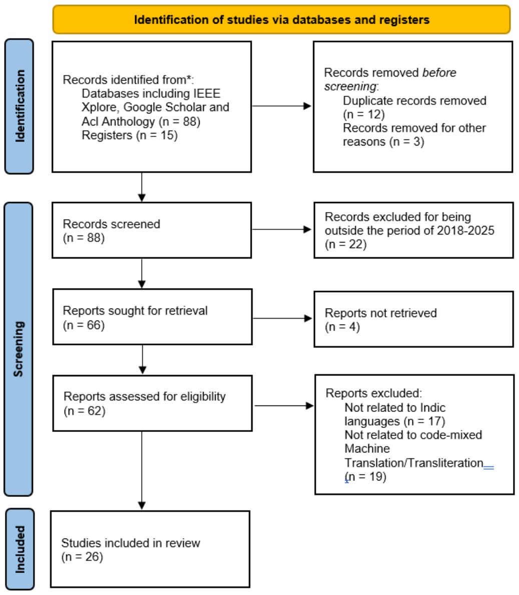

Identification of studies via databases and registers
Identification
Records identified from*:
Databases including IEEE Xplore, Google Scholar and Acl Anthology (n = 88)
Registers (n = 15)
Records removed *before screening*:
Duplicate records removed (n = 12)
Records removed for other reasons (n = 3)
Screening
Records screened (n = 88)
Records excluded for being outside the period of 2018-2025 (n = 22)
Reports sought for retrieval (n = 66)
Reports not retrieved (n = 4)
Reports assessed for eligibility (n = 62)
Reports excluded:
Not related to Indic languages (n = 17)
Not related to code-mixed Machine Translation/Transliteration (n = 19)
Included
Studies included in review (n = 26)

Figure 1: PRISMA flow of literature selection process

4 Code-Mixing background in Indic languages

Language mixing is a result of multilingual language usage across people. This behaviour is more common in multicultural and multilingual societies, such as most countries that use Indo-Aryan languages. Code-mixing is the practice of mixing multiple languages in a single instance (Thara and Poornachandran, 2018). There are two main types of Code-mixing. The first one is Inter-Sentential Switching, which occurs at the boundaries of sentences. One sentence would have a single language type, but multiple sentences would have multiple language types. For example, "*Mujhe abhi jaana hai. I'll call you later*". '*Mujhe abhi jaana hai*' is Hindi and "*I'll call you later*" is English. The second one is Intra-Sentential Switching, which occurs within the same sentence. Therefore, borrowings from different languages can be found in a single sentence (Thara and Poornachandran, 2018). As an example in Hindi-English, "*Main kal office meeting attend karunga*". When considering the languages in code-mixed text Myers-Scotton (1997) has provided a theory called the Linguistic Matrix Language Frame (MLF) theory. In this concept, the dominant language is defined as the Matrix language, and the secondary language is defined as the Embedded language. This is applicable for both code-mixed and code-switched text (Iakovenko and Hain, 2024). The example on Sinhala can be found in Figure 2.

The main reason people tend to use code-mixed language in their day-to-day conversation is that it can express feelings easily and effectively (Sumanathilaka et al., 2023). Use of code-mixing is more common among younger demographics and urban populations (Senaratne, 2009). Due to globalization, most people have adapted to En-

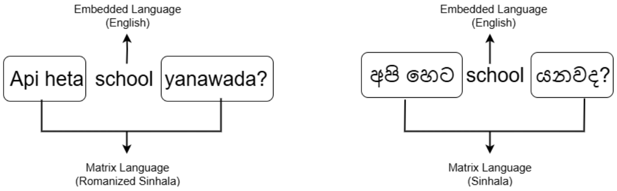


Figure 2: Matrix and Embedded language in code-mixed

glish as the universal language. This increases the number of people who are bilingual or multilingual, with some level of understanding in English. For this reason, it is more common to see people code-mix their native language with English. In the same way code-mixed language is used in verbal communication, it is also common to see this in written communication, especially on informal communication platforms like social media. In social media communication, it has been found that people tend to use their native language in a Romanized way. Despite the availability of native Unicode keyboards, many users prefer Romanized for its ease and convenience (De Silva, 2021).

5 Current state of Code-Mixed Indo-Aryan language translation and transliteration

This section contains a summary of the current state of Code-mixed translation and transliteration in Indo-Aryan languages. The language pairs presented have been carefully selected for discussion, with the choice driven by both the availability of resources and the identified importance of these pairs for advancing research in code-mixed translation and transliteration.

5.1 Hindi - English

Hindi, being one of the most popular languages in India, has several studies that have attempted to translate and transliterate code-mixed Hindi-English content. Bhowmick et al. (2023) have proposed a model to translate Roman Hindi codemixed sentences to monolingual English. First, to train the translation model, they have performed synthetic code-mixed sentence generation by training a mT5 model. The translation model training has been conducted on both augmented and manually scraped data, with 120000 and 4000 sentences. The overall translation pipeline would contain two models: the first is the correction model, which converts Roman Hindi to Devanagari and outputs a Mixed-Script sentence. The second one is the Translation model, which receives the output of the correction model and converts it to monolingual English. (Nair and Gupta, 2024) has proposed a study on exploring the capabilities of different LLMs in translating code-mixed Hindi-English data to English. While evaluating against state-of-the-art decorder-only models/LLMs like GPT-4, Gemini, GEMMA 2, BLOOMZ-3B, and Navarasa 2.0, the zero-shot prompting technique has been followed across all selected models. The Gemini model has outperformed other evaluated models on Roman Hindi code-mixed translation, achieving a BLEU score of 20.82%.

Another study(Gahoi et al., 2022) has been conducted on code-mixed Roman Hindi to English translation. It has utilised the PHINC dataset (Srivastava and Singh, 2020) to train the model, which contains 13,738 parallel code-mixed Hinglish and English sentences. The model has been trained by fine-tuning Salesken AIs pre-trained model mentioned in Huggingface Transformers. Upon evaluating the results, it was identified that the system struggled to translate long sentence inputs. This task returned a results score of 0.41493 for ROUGE-L and 0.80804 for WER Metrics. This paper Jadhav et al. (2022) also has suggested a novel solution to code-mixed Hinglish to English. It will first undergo language identification using an LSTM-based neural method with a dataset of 25,000 Hindi-English words. The identified English words would be tagged as 'en', and Hindi words would be tagged as 'hi'. The identified Hindi words would be transliterated using the Google Transliteration API. Words tagged as English would translate to Hindi using an NMT model. Finally, it would get concatenated to produce the final Hindi output. This has achieved a BLEU score of 0.737 and WER 0.238. This study has demonstrated that incorporating a language identification model into a code-mixed translation task enhances the accuracy of the output. However, it has been stated that the Language Identification model failed to identify language in ambiguous words and based on context.

5.2 Sinhala - English

Sinhala is the most used language in communication in Sri Lanka. Although there are keyboards for Unicode Sinhala, people tend to use the romanized version of Sinhala because of ease (De Silva, 2021; Perera et al., 2025). This creates the need for a translation system.

A recent study (Senanayaka et al., 2024) has been conducted on processing code-mixed Singlish text with RAG implementation for document retrieval. This model could translate English-Singlish code-mixed sentences to English by fine-tuning the LLaMA-2 7B parameter model. This is the first study to develop an LLaMA-based RAG framework tailored for code-mixed Singlish. A Synthetic parallel corpus has been generated using Claude-3-Sonnet. It has been stated that the transformer's attention mechanism enables it to process larger sentences more accurately than other models. This model achieves scores of BLEU 0.1347, ROUGE-1 0.3732, and METEOR 0.5923.

(Kugathasan and Sumathipala, 2022) has done a study on translating code-mixed Singlish to Sinhala. The authors have manually created a dataset comprising 5,000 code-mixed sentences with relevant Sinhala translations. The model is an LSTM Seq2Seq model with a Teacher Forcing mechanism. One of the main novelties of this code-mixed translation task is that, in preprocessing, it utilises the Levenshtein Edit Distance to address ambiguous words in Singlish to some extent. Overall, the study has received a BLEU score of 0.3389. Although a wide range of studies have been conducted on Romanized Sinhala transliteration (Sumanathilaka et al., 2025b; Dharmasiri and Sumanathilaka, 2024), these works exclusively focus on non code-mixed language data. Nevertheless, they underscore the significance of addressing code-mixed scenarios, given their prevalence and practical importance in real-world applications.

5.3 Gujarati - English

(Patel and Parikh, 2020) has proposed an approach to translate code-mixed romanized Gujarati sentences to Gujarati script. This approach would leverage a language identification model that tags the language before performing the translation or transliteration. Authors have identified that Gujarati-English code-mixed data can create ambiguity, for example, the word 'mate' in English, but in Gujarati, it means 'for' in a contextual sense. To overcome that problem, the Hidden Markov Model approach has been used to predict accurate language based on the context.

5.4 Bengali - English

A study has been conducted (Bhowmick et al., 2023) to translate Bengali roman codemixed sentences into monolingual English, utilising mT5 and integrating a correction model and a translation model into the pipeline.

(Shibli et al., 2023) addressed the task of automatic back-transliteration of code-mixed Romanized Bengali into native script Bengali. Their approach generates multiple candidate forms through rule-based phonetic mappings. It applies statistical language models for ranking, similar in spirit to similarity-based scoring and graph-ranking techniques. By resolving ambiguities in noisy romanized input, their system enables more accurate input for subsequent translation models.

5.5 Other Code-Mixed transliteration

Amin et al. (2023) focuses on generating Marathi-English code-mixed text, addressing the lack of code-mixed resources. The proposed method is based on Matrix Language Frame theory, which extracts English phrases identified and transliterated into the Devanagari script, ensuring that they phonetically blend with the surrounding text.

The paper (Wisal et al., 2022) proposes an approach to translating and transliterating code-mixed Roman Urdu-English into Urdu. It utilizes the "g2p-multilingual-byT5-small" deep learning pre-trained model and fine-tunes it with a corpus of code-mixed, romanized Urdu and Urdu translations, which was created by the authors. The system has achieved a BLEU score of 66.73%, providing a strong foundation for noisy low-resource language translations. It has been identified that the model struggles when handling rare vocabulary, culture-specific words, and short sentences.

6 Datasets and Evaluation metrics

6.1 Code-Mixed Datasets

To produce a robust machine translation and transliteration system for code-mixed Indic languages, the quality of the data on which the model was trained was heavily dependent. This section describes publicly available datasets that could be used for code-mixed translation and transliteration tasks on different Indic languages.

6.1.1 Hindi - English

The PHINC dataset (Srivastava and Singh, 2020) is a large parallel dataset with more than 13,000

Study	Languages	Task	Model Type	Dataset	Evaluation	Notes
Hindi						
Bhowmick et al. (2023)	Mixed Roman Hindi/Bengali to English	Translation	mT5 Seq2Seq (Two-step pipeline: Correction + Translation)	5,100 Hindi-English CM	BLEU, METEOR, ROUGE	Mixed Script Augmentation (Roman + native script) improves MT performance.
Nair and Gupta (2024)	Mixed Hindi to English	Translation	LLMs (BLOOMZ-3B) with LoRA and prompting	Not specified	BLEU, chrF++	Demonstrates LLM performance for Indic CM translation.
Gahoi et al. (2022)	Mixed roman Hindi to English	Translation (CM → Monolingual)	mBART (Transformer-based)	Train 8,060, Val 942, Test 960	ROUGE 0.41493; WER 0.80804	Transliteration to Devanagari + Hindi parallel text improves MT.
Jadhav et al. (2022)	Mixed Hinglish to English	Translation	LID layer + GNMT pipeline	25,000 Hindi-English word for LID	BLEU 0.737; TER 0.256; WER 0.238	Uses intermediate Hindi translation via LID for improved accuracy.
Sinhala						
Senanayaka et al. (2024)	Mixed Singlish to English	Bidirectional Translation (RAG)	LLaMA-2 7B + LoRA	100 annotated data + synthetic data	BLEU 0.1347; ROUGE-1 0.3732	Synthetic corpus generation; perplexity reduced to 11.95.
Kugathasan and Sumathipala (2022)	Mixed Roman Sinhala to Sinhala	Normalization; Transliteration; Translation	Seq2Seq LSTM with teacher forcing	5,000 SCM sentences	BLEU 0.3389	Handles OOV words, slang, and inconsistent Romanization.
Gujarati						
Patel and Parikh (2020)	Mixed Roman Gujarati to Gujarati	LID + Normalization + Translation	Naive Bayes + HMM + Dictionary methods	1200 manually created sentences	Manual / API comparison	Hidden Markov Model to predict language.
Bengali						
Shibli et al. (2023)	Roman Banglish to Bengali	Back-Transliteration	Nine transliteration models + BERT similarity	5,000 collected; 1,000 for evaluation	BLEU, ROUGE, WER, WIL	Addresses varied romanization; Google Translate performed best.
Marathi						
Amin et al. (2023)	Marathi-English (Minglish)	CM Text Generation	Linguistic code-mixed generation algorithm	Uses parallel EN-MR corpus	CMI = 0.2; DCM = 7.4	Generates realistic Marathi-English CM sentences.
Urdu						
Wisal et al. (2022)	Mixed Roman Urdu to Urdu	Translation	T5-based multilingual Transformer	17,689 manually created	BLEU 66.73	HuggingFace g2p_multilingual_byT5_small used; dataset created by volunteers.

Table 2: Summary of reviewed methods for Code-Mixed translation and transliteration

code-mixed Romanized Hindi paired with English translations. These sentences have been scraped from social media platforms and utilise the support of six existing corpora that were created for other NLP tasks. Different preprocessing tasks were conducted when creating the corpus to ensure its quality. This contains data from various domains, including Bollywood, sports, politics, and social events.

The Dakshina dataset (Roark et al., 2020) contains resources for 12 different South Indian languages, including Hindi, Bengali, Telugu, Tamil, and Sinhala. It has over 12 million pairs of Romanized to their native script forms. Unlike the PHINC, the Dakshina dataset is more generic and is used in a wider range of NLP tasks, including machine translation, which makes it more suitable for Indic languages.

The L3Cube-HingCorpus (Nayak and Joshi, 2022) is considered the largest code-mixed Hindi-English corpus when compared with other state-of-the-art datasets. It consists of 52.93 million sentences(1.04 billion tokens) collected from Twitter to ensure broader domain coverage and to address the lack of large scale code-mixed Hinglish resources. Unlike the previous two datasets mentioned, L3Cube-HingCorpus does not include parallel translations. Hence, it is not a dataset that could be directly used in translation tasks. However, several studies have outperformed the state-of-the-art results using this corpus.

The HinGe dataset (Srivastava and Singh, 2021) has been introduced to address the scarcity of quality Hindi-English code-mixed resources. The foundation of the HinGe dataset is sourced from the Hindi-English parallel corpus from IIT Bombay (Kunchukuttan et al., 2018). This dataset is structured into two components: Human-generated and Machine-generated sentences. It contains a high-quality collection of 4,803 human-generated sentences, translations annotated with five expert annotators. On the machine-generated side, a total of 3,952 Hinglish sentences have been synthetically generated using two rule-based algorithms: Word-aligned Code-Mixing (WAC) and Phrase-aligned Code-Mixing (PAC). It has been mentioned that this corpus can be used for NLP tasks, such as language identification and POS tagging, in addition to machine translation.

This study (Sheth et al., 2025) has identified that synthetically generated data fails to capture the nuances of real language usage, and human annotation is crucial for creating a high-quality, natural code-mixed text resources. COMI-LINGUA (Sheth et al., 2025) has been developed to address this gap by providing the largest manually annotated dataset for code-mixed text. The translation annotation was performed by three expert annotators who are fluent in both Hindi and English. The dataset has been validated using Fleiss' Kappa measure. The COMI-LINGUA dataset is mainly structured for five different NLP tasks: word-level language identification, sentence-level language identification, part-of-speech tagging, name entity recognition, and machine translation with sentences in Romanized Hindi, Devanagari Hindi, and English.

6.1.2 Sinhala - English

Sinhala, being a low-resource language for Sinhala-English code-mixed, has very limited datasets available. Though large transliteration datasets exist (Sumanathilaka et al., 2024), the availability of code-mixed and properly annotated corpora is limited. This work by Kugathasan and Sumathipala (2022) has provided a corpus that could be used for translating code-mixed Singlish (Sinhala-English) to Sinhala. This corpus consists of over 5,000 parallel code-mixed sentences with their relevant Sinhala translations. There are some other datasets that contain code-mixed Romanized Sinhala for other NLP tasks like language identification, sentiment analysis, etc (Uthpala and Thirukumaran, 2024; Smith and Thayasivam, 2019). But for machine translation tasks, there are extremely limited datasets that could be used.

6.1.3 Bengali - English

The BNSENTMIX dataset (Alam et al., 2025) comprises diverse Bengali-English code-mixed sentences, totalling 20,000. The data has been collected from social media platforms and manually annotated the translations. While this is not a direct translation dataset, it could enhance machine translation pipelines for code-mixed Bengali-English.

6.1.4 Urdu - English

The work by Wisal et al. (2022) has attempted to translate romanised code-mixed Urdu-English text to monolingual Urdu. The authors have annotated 17689 code-mixed Roman Urdu sentences with their relevant translation, with the help of a few annotators.

6.2 Evaluation metrics for code-mixed

Evaluating code-mixed text has its own challenges due to the informal nature and diversity of language. There are several standard matrices for code-mix tasks that have been in use for decades, as well as other matrices that have evolved from these standards. In this section, different evaluation matrices could be used for code-mixed machine translation and transliteration tasks.

6.2.1 BLEU

This is considered the most widely used evaluation metric for machine translation tasks. It calculates the score by measuring the precision of n-grams in candidate translation against the reference translation, with Brevity Penalty to address translations that are short (Papineni et al., 2002). In this review, it has been identified that BLEU often correlates poorly when compared against human judgment.

6.2.2 METEOR

METEOR or Metric for Evaluation of Translation with Explicit Ordering is an improvement done on BLEU by calculating the score not just based on exact match, but stem and synonym (Banerjee and Lavie, 2005).

6.2.3 chrF++

The chrF++ is an enhanced version of chrF, which combines character-level matching with the lexical accuracy of word-level matching(Popovi, 2017). Since this benefits both word-level and character-level analysis, some recent code-mixed studies have utilised this approach for evaluation (Nair and Gupta, 2024).

6.2.4 Word and Character Error rate

Both of these evaluation metrics are logically similar and are based on the concept of Levenshtein distance, which measures the number of edits required to transform one string into another. WER compares the generated text with the reference text on the number of substitutions, deletions, and insertions to make them identical. Similar to WER, the CER would operate on a character level instead of a word level (Gohider and Basir, 2024). The equations for WER and CER would operate as follows:

$$\mathrm{WER} = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{\mathrm{N}} \qquad \mathrm{CER} = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I}}{\mathrm{N}}$$

6.2.5 Translation Edit rate

The Translation Edit rate is an extended version of WER and CER. It would also consider the word shift when measuring the score. Word shift indicates the movement of the location of particular text. A lower TER score indicates a better translation (Snover et al., 2006).

$$\mathrm{TER} = \frac{\mathrm{S} + \mathrm{D} + \mathrm{I} + \mathrm{H}}{\mathrm{N}}$$

6.2.6 Human Evaluation

Because code-mixing admits many sentimentally correct forms that other metrics, like n-gram, fail to capture, human judgment would still be the most accurate method of evaluation. Recent studies have demonstrated that standard metrics, such as BLEU, can be misleading for code-mixed outputs, and that human assessments better reflect fluency and the preservation of code-mixing patterns (Gupta et al., 2024; Vavre et al., 2022). Case studies comparing automatic and human evaluations similarly show that human evaluations detect semantic faithfulness and nuanced phenomena introduced by code-mixing that automatic metrics would miss (Nguyen et al., 2023).

7 Gaps and Challenges in Machine Translation and Transliteration for Code-mixed Indo-Aryan Languages

Although recent advancements have been made in the domain of machine translation and transliteration for code-mixed Indo-Aryan languages, several gaps and challenges remain that can be identified. In this section of the review, we will discuss those identified gaps and challenges in this domain.

7.1 Limited Datasets

When it comes to machine translation and transliteration tasks in code-mixed Indo-Aryan languages, datasets play a significant role in the system's output. There are very limited datasets available for code-mixed Indo-Aryan languages, particularly those that can be utilised for machine translation and transliteration tasks. Through this review, it has been identified that there are more parallel corpus for code-mixed Hindi-English rather than other Indo-Aryan languages. Languages like Sinhala, Gujarati, and Bengali have an extremely limited number of datasets that can be used for translation and transliteration tasks. Hence, ensuring a gold standard parallel corpus is essential, especially for languages with limited datasets.

7.2 Transliteration Ambiguity

Transliteration ambiguity refers to a word that has multiple senses in the context of translation and transliteration (Perera and Sumanathilaka, 2025a). Identifying the correct meaning of the word is significantly important to process code-mixed language NLP tasks, including machine translations (Hidayatullah et al., 2022a). As an example in the Sinhala-English sentence *"Ape rate weather eka"*, the word *'rate'* in the romanised Sinhala format refers to the country. Hence, in this context, the word *'rate'* cannot be considered an English word which has the sense of a *"measure, quantity, or frequency"*. Most of the papers reviewed acknowledge addressing Transliteration ambiguity as a challenge, and only a limited number of studies have attempted to provide solutions for this issue in machine translation and transliteration tasks.

7.3 Non-Standard words

Since code-mixing is more commonly associated with social media or informal communication, it is more likely to contain non-standard words. (Hidayatullah et al., 2022a) has categorized the non-standard words into four main types: non-standard spelling, mixing words and numeric or special characters, word exaggeration or wordplay, and abbreviated words. Table 3 describes the types of non-standard words with examples (Barik et al., 2019).

7.4 Code-mixing lexical patterns

When communicating in code-mixed languages, people maintain a lexical pattern unique to each

Non-Standard word type	Example
Non-standard spelling	Prends(friends), plz(please)
word, numbers, special characters mixing	2morrow(tomorrow), 3wheel (Sinhala language meaning trishow in English)
Word exaggeration	gooood(good), woooow(woow), helloooo(hello)
Abbreviated words	TC(take care), tkt(ticket)

Table 3: Types of non-standard words

Pattern type	Sinhala Example	English translation
Present tense	'act krnawa', 'Drive karanawa'	'Acting', 'Driving'
Past tense	'act kara', 'Drive kara'	'Acted', 'Drove'
Indefinite article	'voice ekak', 'chapter ekak'	'A Voice', 'A Chapter'
Definite article	'voice eka', 'chapter eka'	'The Voice', 'The Chapter'
Suffixes	'studentsla', 'teacherla'	Plural form of 'student' and 'teacher'

Table 4: Types of code-mix lexical patterns in Sinhala-English

code-mixed language pair. This is not something that was agreed on formally, but something that could be identified when analyzing the code-mixed language patterns. For example, in Sinhala-English code-mixed language, the word "eka" would be used after most English words. Like "Computer eka" and "Vehicle eka" (Smith and Thayasivam, 2019). Table 4 describes more lexical patterns of Sinhala-English code-mixing. Although these patterns are unique to each language, some of the patterns remain unsolved when analysing the recent study on the domain.

7.5 Compatible evaluation metrics

Traditional machine translation metrics, such as BLEU, METEOR, and TER, are commonly used for compatibility; however, these metrics often fail or are insufficient for handling code-mixed translation and transliteration. The primary limitation of existing evaluation metrics is their inability to handle multiple valid translation outputs. When translating the embedding language to matrix language, the translation could have used an accurate synonym that matches the reference text. However, existing metrics lack the ability to understand contextual meaning.

7.6 Pre-processing and Language Identification Issues

Preprocessing is considered an important step, as code-mixed data tends to be noisier compared to standard text data. When code-mixing is involved with Romanized text, it becomes challenging to perform certain preprocessing tasks, such as spelling correction. A simple spelling correction system would not be able to succeed in a Romanized code-mixed setting since an 'incorrect' token may belong to the other mixed language. Applying a normal spelling correction model risks introducing further errors than normalizing them. Hence, it creates the need for a context-aware spelling correction system.

In this review, it has been identified that some proposed systems have implemented a Language Identification model in the pre-processing pipeline to address ambiguity. But state-of-the-art Language Identification models could only address word-level language identification. Sub-word level language identification is needed to address code-mixed words like 'studentsla'(plural form of Student), where 'Student' is English and 'la' is Sinhala. These challenges need to be addressed, as even small misclassifications propagate into major quality degradation.

8 Conclusion

This review paper has provided a comprehensive analysis of the current advancements, datasets, evaluation methods, and challenges in Machine Translation and transliteration techniques, with a specific focus on code-mixed Indo-Aryan languages. These studies are important to ensure effective communication across different code-mixed indo-aryan languages. In this review, it is evident that the code-mix translation and transliteration accuracies have significantly improved when combined with recent discoveries in the domain of NLP. This marks a promising direction for addressing future research gaps and producing products that solve real-world problems.

Limitation

This review contains several limitations that should be acknowledged. The review primarily focuses on literature published between 2018 and 2025. This ensures that recent advancements are reviewed, but it may have excluded earlier foundational studies. A few studies were excluded due to accessibility issues. Finally, this study focuses on academic studies rather than doing a systematic analysis of industrial or applied systems, which could offer additional insights into the practical difficulties of dealing with code-mixed Indo-Aryan text.

Acknowledgments

This paper has been conducted in compliance with the ethical standards of the Informatics Institute of Technology, Sri Lanka. Generative AI tools were used solely to enhance the clarity and readability of the manuscript.

References

S. Alam and 1 others. 2025. Bnsentmix: A diverse bengali-english code-mixed dataset for sentiment analysis. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 68–77, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dhiraj Amin, Sharvari Govilkar, Sagar Kulkarni, Yash Shashikant Lalit, Arshi Ajaz Khwaja, Daries Xavier, and Sahil Girijashankar Gupta. 2023. Marathi-english code-mixed text generation. *arXiv preprint arXiv:2309.16202*.

Maneesha U Athukorala and Deshan K Sumanathilaka. 2024. Swa bhasha: Message-based singlish to sinhala transliteration. *arXiv preprint arXiv:2404.13350*.

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

A.M. Barik, R. Mahendra, and M. Adriani. 2019. Normalization of indonesian-english code-mixed twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, Hong Kong, China. Association for Computational Linguistics.

R.S. Bhowmick and 1 others. 2023. Improving indic code-mixed to monolingual translation using mixed script augmentation, generation & transfer learning. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

A.D. De Silva. 2021. Singlish to sinhala converter using machine learning. Master's thesis, University of Colombo School of Computing. [Accessed 25 August 2025].

Sachithya Dharmasiri and TGDK Sumanathilaka. 2024. Swa bhasha 2.0: Addressing ambiguities in romanized sinhala to native sinhala transliteration using neural machine translation. In *2024 4th International Conference on Advanced Research in Computing (ICARC)*, pages 241–246. IEEE.

A. Gahoi and 1 others. 2022. Gui at mixmt 2022: English-hinglish: An mt approach for translation of code mixed data. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1126–1130, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nada Gohider and Otman A. Basir. 2024. Recent advancements in automatic disordered speech recognition: A survey paper. *Natural Language Processing Journal*, 9:100110.

Ayushman Gupta, Akhil Bhogal, and Kripabandhu Ghosh. 2024. Multilingual controlled generation and gold-standard-agnostic evaluation of code-mixed sentences. *Preprint*, arXiv:2410.10580.

HM Anuja Dilrukshi Herath and TG Deshan K Sumanathilaka. 2024. Tamzhi: Shorthand romanized tamil to tamil reverse transliteration using novel hybrid approach. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 17(1).

A.F. Hidayatullah and 1 others. 2022a. A systematic review on language identification of code-mixed text: Techniques, data availability, challenges, and framework development. *IEEE Access*, 10:122812–122831.

Ahmad Fathan Hidayatullah, Atika Qazi, Daphne Teck Ching Lai, and Rosyzie Anna Apong. 2022b. A systematic review on language identification of code-mixed text: Techniques, data availability, challenges, and framework development. *IEEE Access*, 10:122812–122831.

O. Iakovenko and T. Hain. 2024. Methods of automatic matrix language determination for code-switched speech. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5791–5800, Miami, Florida, USA. Association for Computational Linguistics.

I. Jadhav and 1 others. 2022. Code-mixed hinglish to english language translation framework. In *2022 International Conference on Sustainable Computing* *and Data Communication Systems (ICSCDS)*, pages 684–688.

A. Kugathasan and S. Sumathipala. 2022. Neural machine translation for sinhala-english code-mixed text. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 15(3):60–71.

A. Kunchukuttan, P. Mehta, and P. Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association.

C. Myers-Scotton. 1997. *Duelling Languages: Grammatical Structure in Codeswitching*. Clarendon Press.

A.R. Nair and D. Gupta. 2024. Evaluating performance and accuracy of large language models in translating code-mixed hindi to english: A comparative study. In *2024 IEEE 21st India Council International Conference (INDICON)*, pages 1–6.

R. Nayak and R. Joshi. 2022. L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Li Nguyen, Christopher Bryant, Oliver Mayeux, and Zheng Yuan. 2023. How effective is machine translation on low-resource code-switching? a case study comparing human and automatic metrics. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14186–14195, Toronto, Canada. Association for Computational Linguistics.

S. Pal and M. Zampieri. 2020. Neural machine translation for similar languages: The case of indo-aryan languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 424–429. Association for Computational Linguistics.

K. Papineni, S. Roukos, T. Ward, and W.-J. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

D. Patel and R. Parikh. 2020. Language identification and translation of english and gujarati code-mixed data. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–4.

Sandun Sameera Perera, Lahiru Prabhath Jayakodi, Deshan Koshala Sumanathilaka, and Isuri Anuradha. 2025. Indonlp 2025 shared task: Romanized sinhala to sinhala reverse transliteration using bert. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 135–140.

Sandun Sameera Perera and Deshan Sumanathilaka. 2025a. Evaluating transliteration ambiguity in adhoc romanized sinhala: A dataset for transliteration disambiguation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2025)*.

Sandun Sameera Perera and Deshan Koshala Sumanathilaka. 2025b. Machine translation and transliteration for indo-aryan languages: A systematic review. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 11–21.

M. Popovi. 2017. chrf++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

B. Roark and 1 others. 2020. Processing south asian languages written in the latin script: the dakshina dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association.

S.M.M.R.J. Senanayaka, A.W.A.D.N.D. Abeysekara, and M.G.N. Premadasa. 2024. Singrag: A translation-augmented framework for code-mixed singlish processing. In *2024 9th International Conference on Information Technology Research (ICITR)*, pages 1–6.

C. Senaratne. 2009. *Sinhala-English code-mixing in Sri Lanka: A sociolinguistic study*. LOT Publications.

R. Sheth, H. Beniwal, and M. Singh. 2025. Comilingua: Expert annotated large-scale dataset for multitask nlp in hindi-english code-mixing. ArXiv preprint.

G.M.S. Shibli and 1 others. 2023. Automatic back transliteration of romanized bengali (banglish) to bengali. *Iran Journal of Computer Science*, 6(1):69–80.

I. Smith and U. Thayasivam. 2019. Language detection in sinhala-english code-mixed data. In *2019 International Conference on Asian Language Processing (IALP)*, pages 228–233.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

V. Srivastava and M. Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49. Association for Computational Linguistics.

V. Srivastava and M. Singh. 2021. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208. Association for Computational Linguistics.

Deshan Sumanathilaka, Isuri Anuradha, Ruvan Weerasinghe, Nicholas Micallef, and Julian Hough. 2025a. Indonlp 2025: Shared task on real-time reverse transliteration for romanized indo-aryan languages. *arXiv preprint arXiv:2501.05816*.

Deshan Sumanathilaka, Nicholas Micallef, and Ruvan Weerasinghe. 2024. Swa-bhasha dataset: Romanized sinhala to sinhala adhoc transliteration corpus. In *2024 4th International Conference on Advanced Research in Computing (ICARC)*, pages 189–194.

Deshan Sumanathilaka, Sameera Perera, Sachithya Dharmasiri, Maneesha Athukorala, Anuja Dilrukshi Herath, Rukshan Dias, Pasindu Gamage, Ruvan Weerasinghe, and YHPP Priyadarshana. 2025b. Swa-bhasha resource hub: Romanized sinhala to sinhala transliteration systems and data resources. *arXiv preprint arXiv:2507.09245*.

TGDK Sumanathilaka, Ruvan Weerasinghe, and YHPP Priyadarshana. 2023. Swa-bhasha: Romanized sinhala to sinhala reverse transliteration using a hybrid approach. In *2023 3rd International Conference on Advanced Research in Computing (ICARC)*, pages 136–141. IEEE.

S Thara and Prabaharan Poornachandran. 2018. Code-mixing: A brief survey. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2382–2388.

D. K. Uthpala and S. Thirukumaran. 2024. Sinhala-english code-mixed language dataset with sentiment annotation. In *2024 4th International Conference on Advanced Research in Computing (ICARC)*, pages 184–188.

Aditya Vavre, Abhirut Gupta, and Sunita Sarawagi. 2022. Adapting multilingual models for code-mixed translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7133–7141, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Muhammad Wisal, Abbas Mustafa, and Umair Arshad. 2022. Cmrutu: Code mixed roman urdu (roman urdu and english) to urdu translator. In *2022 24th International Multitopic Conference (INMIC)*, pages 1–5.

CycleDistill: Bootstrapping Machine Translation using LLMs with Cyclical Distillation

Deepon Halder[1,4] **Thanmay Jayakumar**[1,2] **Raj Dabre**[1,2,3*]

[1]Nilekani Centre at AI4Bharat [2]Indian Institute of Technology, Madras
[3]Indian Institute of Technology, Bombay
[4]Indian Institute of Engineering, Science and Technology, Shibpur

Abstract

Large language models (LLMs), despite their ability to perform few-shot machine translation (MT), often lag behind dedicated MT systems trained on parallel corpora, which are crucial for high quality machine translation (MT). However, parallel corpora are often scarce or non-existent for low-resource languages. In this paper, we propose CycleDistill, a bootstrapping approach leveraging LLMs and few-shot translation to obtain high-quality MT systems. CycleDistill involves iteratively generating synthetic parallel corpora from monolingual corpora via zero- or few-shot MT, which is then used to fine-tune the model that was used for generating said data for MT. CycleDistill does not need parallel corpora beyond 1 to 4 few-shot examples, and in our experiments focusing on three Indian languages, by relying solely on monolingual corpora, it can achieve high-quality machine translation, improving upon a few-shot baseline model by **20-30 chrF points** on average in the first iteration. We also study the effect of leveraging softmax activations during the distillation process and observe mild improvements in translation quality. We publicly release the source code associated with this work[1].

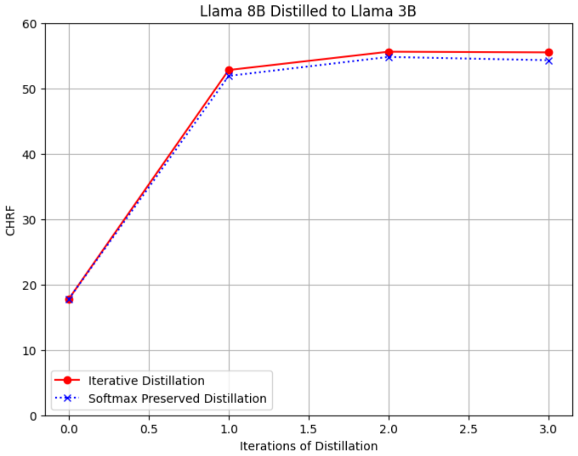


Figure 1: chrF scores over distillation cycles for LLaMA 8B → 3B using Iterative and Softmax-Preserved Distillation under a zero-shot Hindi setting. Marginal gains observed across iterations.

1 Introduction

Machine translation (MT) for low-resource languages poses persistent challenges due to the limited availability of bilingual corpora and the linguistic variation these languages exhibit. Although large language models (LLMs) can perform translation with minimal supervision, their effectiveness in low-resource settings is typically inferior to systems trained with substantial parallel data (Koehn et al., 2017; Gu et al., 2018).

This paper introduces *CycleDistill*, a resource-efficient framework for improving translation quality without requiring extensive parallel data. The approach begins with a small set of example translations and utilizes LLMs to generate synthetic parallel corpora from monolingual text. These corpora are then used to iteratively fine-tune the translation model, enabling progressive performance gains with each cycle.

The framework incorporates two key techniques. First, *Iterative Synthetic Data Distillation* leverages repeated cycles of data generation and model training to enhance translation performance over time (Kim et al., 2021). Second, *Soft Distribution-Preserving Distillation* transfers detailed token-level probability distributions from teacher to student models, allowing for more comprehensive knowledge retention (Tan et al., 2019). Building on previous work in self-training (He et al., 2020), sequence-level and soft-target knowledge distillation (Kim and Rush, 2016; Hinton et al., 2015), *CycleDistill* offers a practical and scalable solution for MT in low-resource scenarios.

The main contributions of this work are:

[*]Corresponding Author: raj.dabre@cse.iitm.ac.in
[1]Code : Github

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 78–92
December 24, 2025 ©2025 Association for Computational Linguistics

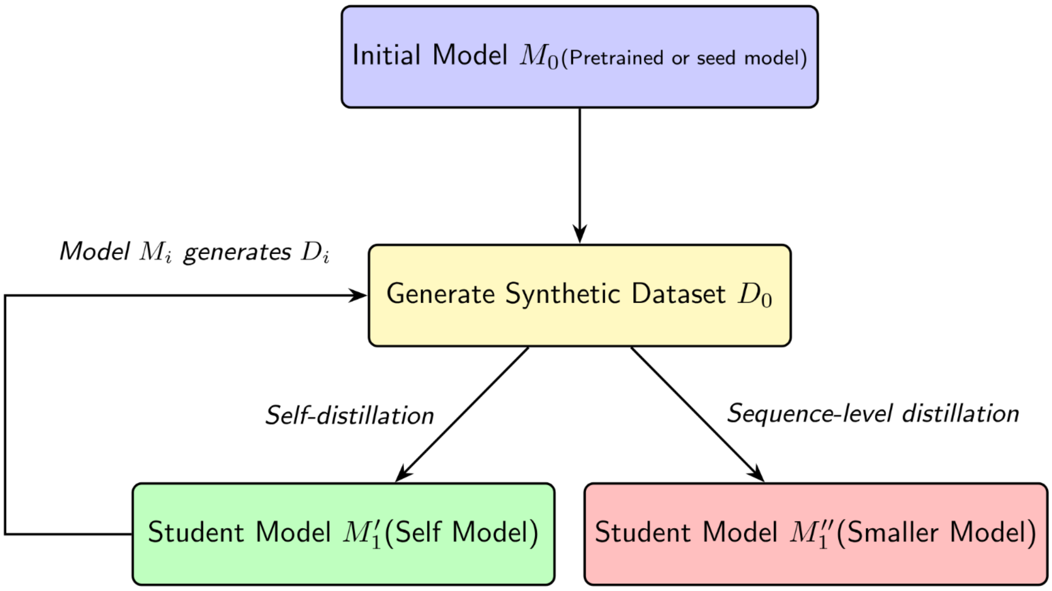

Initial Model M_0(Pretrained or seed model)
Model M_i generates D_i
Generate Synthetic Dataset D_0
Self-distillation
Sequence-level distillation
Student Model M_1'(Self Model)
Student Model M_1''(Smaller Model)

Figure 2: An overview of the CycleDistill framework, which iteratively generates synthetic parallel data from monolingual corpora and refines translation models through cyclic self distillation.

- We present *CycleDistill*, a self-supervised MT framework that improves translation quality using only monolingual corpora and minimal supervision.
- We propose a token-level soft distillation strategy to facilitate richer and more effective learning from teacher models.
- We demonstrate that our method achieves substantial improvements of 20-30 chrF points over few-shot translation baselines, with consistent chrF score gains across three Indian low-resource languages.

2 Related work

Low-resource machine translation (MT) remains a significant challenge due to the scarcity of parallel corpora and high linguistic diversity (Koehn et al., 2017; Gu et al., 2018). Knowledge distillation (KD) has become a popular approach for addressing these issues, transferring knowledge from large teacher models to smaller student models (Hinton et al., 2015). Sequence-level KD (Kim and Rush, 2016) and iterative or self-training strategies (Kim et al., 2021; Furlanello et al., 2018) have demonstrated improvements in low-resource and multilingual MT (Tan et al., 2019). Recent advances include continual KD, which sequentially distills knowledge from multiple existing models (Zhang et al., 2023), and encoder-aware KD for better transfer in compute-constrained and low-resource settings (Velayuthan et al., 2025).

Back-translation and its iterative variants are also highly effective for low-resource MT, as they leverage monolingual data to generate synthetic parallel corpora (Edunov et al., 2018; Hoang et al., 2018). These methods have shown strong gains in extremely low-resource and Indic language scenarios, especially when combined with transfer learning and data filtering (Luo et al., 2020; Tars et al., 2021; Ahmed et al., 2023; Krishnamurthy et al., 2024).

While both KD and back-translation have advanced the field, their integration and comparative effectiveness, particularly in settings with minimal parallel supervision, remain active areas of research. Our proposed **CycleDistill** framework is novel in that it bootstraps high-quality MT systems using only monolingual corpora and a handful of few-shot examples, without relying on large-scale parallel data. Unlike prior work, CycleDistill combines cyclical iterative synthetic data generation with token-level soft distribution-preserving distillation, enabling progressive model refinement and compression.

3 Methodology

This work aims to enhance low resource languages to English machine translation through the adoption of two iterative distillation strategies: cyclic synthetic data generation and an advanced distilla-

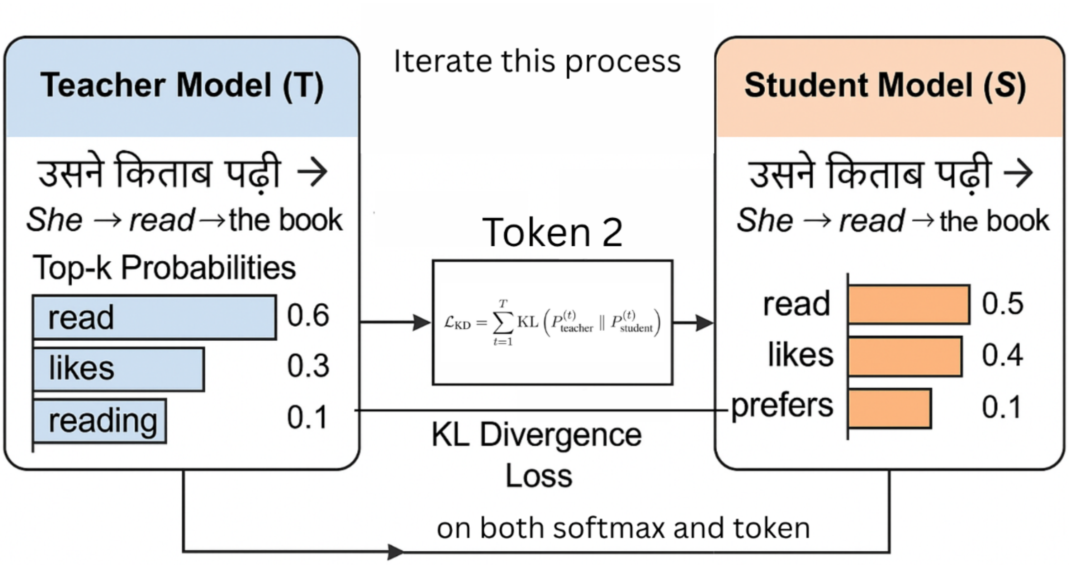


Figure 3: An Overview of the *Soft Distribution Preserving Distillation*. Unlike standard distillation, this method preserves top-k token distributions at each position. The student model learns not only from final outputs but also from the richer probability landscape, encouraging finer-grained generalization.

tion approach that preserves detailed token-level information, such as softmax distributions and subword structures. Our methodology is grounded in recent developments in knowledge distillation and self-training for neural machine translation (Kim and Rush, 2016; Gou et al., 2021).

3.1 Iterative Synthetic Data Distillation

Our first approach enables the base translation model to iteratively improve by generating and learning from its own synthetic data. The procedure is as follows:

- **Base Model Initialization:** The process begins with a pretrained base translation model, denoted as M_0, which is capable of translating from an Indic language to English.

- **Synthetic Data Generation:** The model M_0 is employed to generate a synthetic dataset D_0 comprising translation pairs. This step is inspired by self-training methodologies that have demonstrated efficacy in low-resource scenarios (He et al., 2020).

- **Self-Distillation:** Utilizing the generated synthetic data, knowledge distillation is performed in two ways:
  - The same model architecture is further refined, resulting in an updated model M_1.
  - Additionally, knowledge is distilled into a smaller student model, M_1', via sequence-level knowledge distillation, whereby the student learns from the teacher's generated translations (Kim and Rush, 2016).

- **Iterative Refinement:** This procedure is repeated for three cycles. In each iteration i (where $i = 1, 2, 3$):
  - The distilled model M_i (or M_i') produces a new dataset D_i comprising additional translations.
  - Subsequently, M_i is distilled into M_{i+1} and a new student model M_{i+1}'.

The underlying principle is that, by iteratively learning from its own outputs, the model can progressively improve its performance. This iterative process benefits both the primary and the student models, enhancing their generalization capabilities and, in certain cases, enabling model size reduction with minimal compromise in performance.

3.2 Soft Distribution-Preserving Distillation

The second strategy extends the distillation process by capturing more granular information from the teacher model:

- **Enhanced Data Extraction:** During synthetic translation generation, for each token position t, we record:

- The top-k token predictions ($\{y_1^{(t)}, \ldots, y_k^{(t)}\}$) (Fan et al., 2018)
- The corresponding softmax probabilities ($\{p_1^{(t)}, \ldots, p_k^{(t)}\}$), where $\sum_{j=1}^{k} p_j^{(t)} \leq 1$

This comprehensive information set is motivated by the demonstrated effectiveness of soft-target distillation in capturing the teacher model's knowledge (Hinton et al., 2015).

- **Logit-Based Distillation:** The student model is trained to match not only the final output sequences but also the softmax distributions over the top-k tokens at each position. This is achieved by minimizing the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) loss:

$$\mathcal{L}_{\text{KD}} = \sum_{t=1}^{T} \text{KL}\left(P_{\text{teacher}}^{(t)} \parallel P_{\text{student}}^{(t)}\right)$$

  where T denotes the sequence length, and $P^{(t)}$ represents the softmax distributions. This approach enables the student model to more accurately approximate the teacher's behavior, as suggested in prior research (Hinton et al., 2015; Mukherjee and Khapra, 2021).

- **Iterative Distillation:** This process is also conducted over three iterations. In each cycle, the student from the previous round assumes the role of the new teacher, and a fresh synthetic dataset is generated, ensuring the transfer of rich token-level distributions.

4 Experiments

This section outlines the experimental framework designed to investigate the efficacy of iterative knowledge distillation in enhancing machine translation quality. Our approach involves distilling knowledge from larger language models into smaller counterparts, followed by comprehensive performance evaluation across multiple metrics and languages.

4.1 Models and Languages

Our study employs four language models of varying sizes from the LLaMA (Meta, 2024) and Gemma (Google, 2024) families:

- **Gemma 2 9B** (G_{9B})
- **Gemma 2 2B** (G_{2B})
- **LLaMA 3.1 8B** (L_{8B})
- **LLaMA 3.2 3B** (L_{3B})

Each larger model undergoes distillation to produce both a refined same-size model and a compressed smaller model, adhering to established Sequence Distillation principles (Kim and Rush, 2016). Our evaluation encompasses three Indic languages:

- **Hindi** (HIN)
- **Bengali** (BEN)
- **Malayalam** (MAL)

4.2 Distillation Process

For a given teacher model T, distillation is performed to produce two student models:

- Same-size student ($S_{\text{same}} \leftarrow T$)
- Smaller student ($S_{\text{small}} \leftarrow T$)

The distillation relationships are formally expressed as:

$$G_{9B} \rightarrow \{G'_{9B}, G_{2B}\}, \quad L_{8B} \rightarrow \{L'_{8B}, L_{3B}\}$$

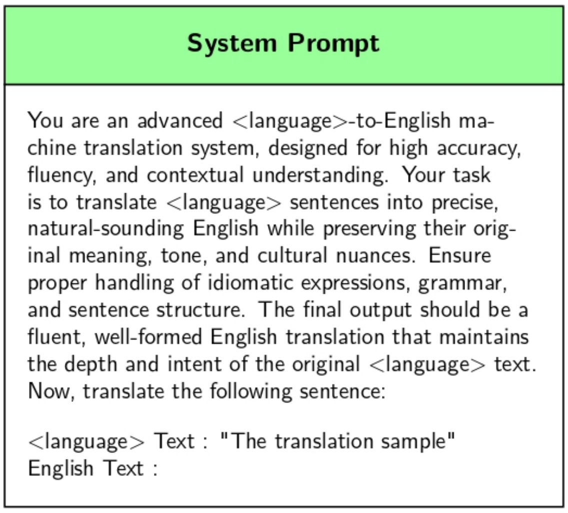
where the refined large models (G'_{9B}, L'_{8B}) are subsequently utilized for synthetic data generation. We select $k = 20$ after empirical evaluation of the teacher models' output distributions revealed that the probability mass beyond the 20 highest-scoring tokens is negligible. We perform the experiments only upto three iterations ($n = 3$). This limit was set because we observed that the performance gains stabilized after the third iteration. Further iterations yielded negligible improvements, indicating that the models were approaching a performance plateau, making additional computational cycles inefficient.

4.3 Training Data

Models are fine-tuned using the **BPCC seed corpus** (Gala et al., 2023), a parallel Indic-to-English dataset. Consistent with established practices in low-resource translation research (Kunchukuttan et al., 2023), we randomly sample 20,000 sentence pairs for training and distillation. We use a fixed prompt format for all of the language and model pair, discussed in Figure 4.

4.4 Synthetic Data Generation

Following each distillation iteration, the most recent large model generates synthetic English translations for the original 20,000 source sentences. This synthetic data generation process is repeated for three complete cycles to enable progressive model refinement.



System Prompt

You are an advanced <language>-to-English machine translation system, designed for high accuracy, fluency, and contextual understanding. Your task is to translate <language> sentences into precise, natural-sounding English while preserving their original meaning, tone, and cultural nuances. Ensure proper handling of idiomatic expressions, grammar, and sentence structure. The final output should be a fluent, well-formed English translation that maintains the depth and intent of the original <language> text. Now, translate the following sentence:

<language> Text : "The translation sample"
English Text :

Figure 4: Example of the general prompt used for the translation task.

4.5 Prompt Used

The prompt utilized for the translation task described in Section 4.3 is shown in Figure 4.

In 1-shot and 4-shot settings, example translation pairs are inserted into the middle section of the prompt prior to the final instruction.

4.6 Evaluation

Model performance is assessed using the **IN22 Gen corpus** (Gala et al., 2023), the standard evaluation benchmark coupled with the BPCC seed corpus. The translation quality is quantified through chrF scores (Popović, 2015). This metric provides standardized measurement of n-gram translation accuracy, aligning with current best practices in machine translation evaluation.

5 Results and Analyses

We first describe our main results on CycleDistill (iterative self distillation) and then analyze its various effects.

5.1 Main Results

Zero-Shot Setting We observe a consistent performance trend across iterations of distillation. The first iteration results in a substantial performance increase. The second and third iteration usually have similar values with the first iteration, but we notice a small increase of 1-2% of chrF with each iteration.

This pattern holds true for both *iterative distillation* and *soft distribution-preserving distillation*, with no significant differences observed between the two. However there are some notable results –

- For the Gemma 2B model with Bengali and the LLaMA 3B model with Malayalam, iterative distillation outperforms soft distribution-preserving distillation.

- In contrast, for the LLaMA 8B model with Hindi and the LLaMA 3B model with Bengali, soft distribution-preserving distillation demonstrates superior performance compared to iterative distillation.

One-Shot Setting The one-shot setting yields the best overall performance, with the highest chrF scores observed exclusively in this configuration. The performance trend across iterations closely resembles that of the zero-shot setting. We observe some crossover between the two distillation methods, where one approach outperforms the other depending on the iteration count. Notable observations include:

- For the LLaMA 3B model on the Malayalam dataset, iterative distillation surpasses soft distribution-preserving distillation in performance.

- Conversely, for the LLaMA 3B model on the Bengali dataset, soft distribution-preserving distillation outperforms iterative distillation.

Four-Shot Setting Performance declines slightly in the four-shot setting compared to earlier configurations, though iteration-wise differences remain minimal. Both iterative and soft distribution-preserving distillation exhibit similar gradual improvements and overall trends. This drop is primarily attributed to reduced contextual clarity due to increased input length, the four-shot prompt is approximately 60% longer than the one-shot, placing greater demands on the model's context window. Maintaining coherence across multiple examples becomes harder as prompts grow longer. The degradation is more pronounced in linguistically complex languages, suggesting that context dilution disproportionately affects grammatically rich targets. These results highlight the need to balance shot count and context efficiency in multilingual distillation, especially under limited model capacities.

5.2 Impact of Language Morphology on chrF

To further investigate the observed decline in 4-shot performance, particularly for morphologically

Model	Iter	chrF (0-shot)			chrF (1-shot)			chrF (4-shot)		
		BEN	HIN	MAL	BEN	HIN	MAL	BEN	HIN	MAL
G_{9B}	Base	41.4	47.9	39.9	42.7	49.2	38.8	24.2	44.6	14.5
	DD_1	61.1	64.4	60.2	60.8	64.2	60.0	53.1	63.8	37.0
	SD_1	60.9	64.7	60.4	60.1	64.5	57.9	49.3	63.7	18.2
	DD_2	61.4	64.5	60.7	60.5	64.6	60.2	52.4	63.7	37.2
	SD_2	60.5	64.7	60.7	64.8	64.9	59.1	49.3	64.3	32.9
	DD_3	61.0	60.4	61.1	60.6	59.0	60.4	52.8	57.7	37.8
	SD_3	61.4	64.4	61.0	60.9	63.3	58.4	45.0	64.1	48.1
L_{8B}	Base	29.2	33.6	22.8	26.6	36.0	8.5	13.5	24.1	14.0
	DD_1	44.9	29.8	42.6	39.6	26.8	17.6	16.7	18.9	17.4
	SD_1	42.1	40.3	40.6	32.0	39.6	21.2	16.7	29.3	17.4
	DD_2	48.3	50.3	46.2	42.0	55.5	26.4	16.5	51.1	17.4
	SD_2	46.2	54.1	44.5	38.3	39.4	23.5	15.1	33.4	17.4
	DD_3	38.9	37.3	17.8	30.0	27.6	15.0	18.3	21.0	17.4
	SD_3	38.9	50.8	38.0	38.7	40.7	22.3	17.0	27.3	17.4
L_{3B}	Base	24.2	14.5	2.9	18.4	17.8	5.0	13.4	14.5	14.0
	DD_1	46.0	52.7	38.9	39.3	52.8	27.4	27.0	36.3	17.4
	SD_1	49.4	53.1	33.5	37.5	51.9	18.2	17.2	34.5	17.3
	DD_2	34.3	55.0	37.5	28.0	55.6	24.5	12.8	42.7	17.3
	SD_2	52.3	54.4	29.4	39.3	54.8	17.5	16.6	44.4	17.2
	DD_3	26.1	55.1	27.1	16.4	55.5	18.7	13.4	42.6	17.4
	SD_3	45.2	53.9	25.3	37.5	54.3	17.4	13.5	42.8	17.3
G_{2B}	Base	24.6	28.8	23.8	28.7	33.4	27.8	19.0	31.2	13.4
	DD_1	50.9	58.4	48.3	50.3	58.7	46.6	27.7	54.1	25.4
	SD_1	40.1	58.3	48.2	58.3	56.9	47.1	23.8	55.5	23.0
	DD_2	50.0	58.1	48.2	50.1	58.4	47.1	29.0	53.8	25.8
	SD_2	43.0	58.4	49.0	48.8	58.1	47.4	28.6	51.2	21.4
	DD_3	49.9	57.8	47.4	49.4	57.2	46.9	34.9	54.9	25.3
	SD_3	49.1	56.8	48.5	45.4	56.8	47.0	32.8	53.3	21.0
Average		**44.4**	**51.5**	**40.9**	**39.8**	**49.6**	**31.0**	**26.8**	**42.5**	**21.6**

Table 1: chrF scores for all models and methods across three languages and shot settings, with column averages.

rich languages, we visualize language-specific sensitivity to increasing shot settings. As shown in Table 1, we find a notable and steeper decline from 1-shot to 4-shot for Bengali and Malayalam, compared to Hindi, which supports the hypothesis that context dilution disproportionately impacts morphologically complex languages.

5.3 Effectiveness in Extremely Low Resource Languages

Study on Nepali To assess the robustness and generalizability of our proposed method in low-resource or moderately known language settings, we conducted experiments using Meta's LLaMA 3.1 8B and LLaMA 3.2 3B models. We selected Nepali, written in the Devanagari script, as the target language. This language has partial representation in the model's pretraining corpus, which means the models possess a basic understanding of it and are capable of generating reasonable outputs, although it is not extensively covered. Despite this limited exposure, the models were able to produce useful distillation data. When we applied our method, we observed consistent improvements over baseline methods, as shown in Table 2. These results suggest that our method remains effective even when the target language has minimal presence in the training data. This demonstrates the potential of our approach to enhance performance in low-resource and cross-lingual generalization scenarios.

Study on Manipuri The investigation included preliminary experiments on the Manipuri (Meitei script) to English translation task, utilizing several prominent large language models, specifically GPT-4, LLaMA 3.1 8B, and Gemma 2 9B. These models were evaluated for their ability to generate synthetic distillation data, which is the first step for the proposed CycleDistill framework.

Results indicated a significant limitation: none of the evaluated models were capable of producing usable distillation data for Manipuri. This suggests

Model	Iter	Nepali (Devanagari Script)			Manipuri (Meitei Script)		
		0-shot	1-shot	4-shot	0-shot	1-shot	4-shot
L_{8B}	Base	12.47	13.95	–	16.88	17.45	17.45
	DD_1	38.59	38.08	–	18.51	17.74	17.75
	SD_1	54.44	36.19	–	16.97	17.61	17.43
	DD_2	35.23	30.45	–	18.52	17.02	17.17
	SD_2	54.31	35.19	–	18.84	17.82	18.08
	DD_3	33.24	20.38	–	17.87	15.97	15.98
	SD_3	54.74	34.35	–	18.04	16.98	16.93
L_{3B}	Base	17.16	17.15	–	17.13	17.44	17.45
	DD_1	48.55	48.75	–	18.58	16.82	17.41
	SD_1	47.31	25.51	–	18.70	16.77	16.81
	DD_2	40.48	38.23	–	17.88	14.74	14.57
	SD_2	47.31	25.67	–	17.35	15.11	14.81
	DD_3	41.15	39.34	–	17.49	15.73	15.59
	SD_3	47.08	31.11	–	17.08	13.64	13.47

Table 2: chrF scores for Nepali (Devanagari script) and Manipuri (Meitei script) over the Llama model family.

that the process is inherently constrained in environments where the base large language model cannot effectively perform few-shot translation for the target low-resource language. Further detailed experiments were conducted on Manipuri (Meitei script) using the LLaMA 3.1 8B and LLaMA 3.2 3B models within the iterative distillation framework. As presented in Table 2, these results consistently showed no improvement in chrF scores across successive iterations.

5.4 Further Analyses

Teacher Quality vs. Student Gain

To examine the correlation between teacher model performance and student gains within our CycleDistill framework, we analyzed the relevant data as depicted in Figure 5, where the x-axis indicates teacher performance (measured by the chrF score of models such as G'^{*}_{9B} or L'^{*}_{8B} when generating synthetic data), and the y-axis represents student gain (ΔchrF, denoting the improvement over the baseline, e.g., $\text{chrF}^{*}_{G^{*}_{2B}\text{distilled}} - \text{chrF}^{*}_{G^{*}_{2B}\text{base}}$).

Our analysis reveals that this relationship varies by shot setting. In zero-shot, a positive correlation holds, with higher teacher scores driving greater gains, validating distillation's reliance on data quality in example-free scenarios. In one-shot, correlation vanishes, as a single example anchors learning, making gains independent of teacher quality. In four-shot, gains are suppressed overall, due to context dilution and error propagation in longer prompts, positioning one-shot as the optimal for effective distillation.

Error Propagation and Recovery

A key limitation observed during our experiments is the susceptibility of the iterative framework to error propagation. Specifically, if an error such as the use of incorrectly generated or misaligned synthetic data is introduced at any iteration (for example, the second cycle), it can lead to a substantial degradation in performance, with declines of up to 30 to 40 chrF points observed in certain settings. These errors are compounded across subsequent iterations, as the model continues to self-distill based on flawed data, making recovery increasingly difficult. However, we also find that corrective interventions such as fine-tuning with accurately generated synthetic data can effectively mitigate such errors in subsequent iterations. This underscores the importance of early detection and correction of distillation errors, as well as the need for robust validation mechanisms during each cycle to prevent error amplification.

Performance of CycleDistill over Model Families

A key finding is the divergence in performance between LLaMA and Gemma models under CycleDistill, as shown in Figure 6. Gemma exhibits superior, robust learning, as compared to LLaMA.

These results emphasize that the choice of base model architecture critically influences the stability and effectiveness of iterative distillation strategies.

Efficiency of Knowledge Absorption across Model Families

The analysis of knowledge absorption rates reveals that the LLaMA 3B model exhibits a signifi-

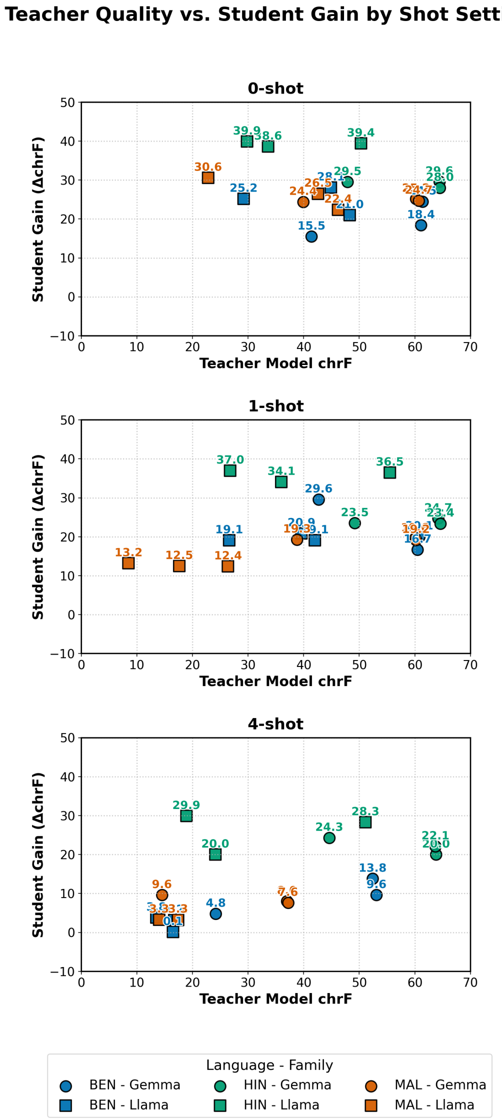


Figure 5: Scatter plot illustrating the relationship between teacher model performance and student model gain across zero-shot, one-shot, and four-shot settings in the CycleDistill framework.

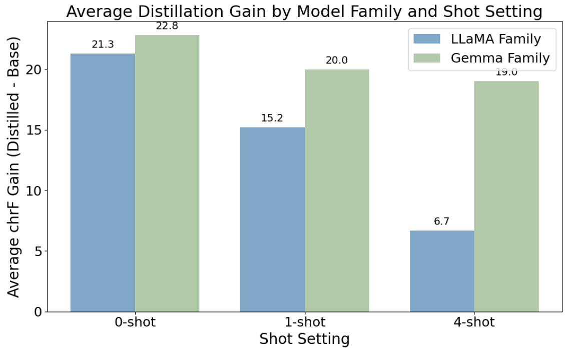


Figure 6: chrF gains for Gemma and LLaMA across shot settings.

cantly higher efficiency in learning from its teacher compared to the Gemma 2B model. Specifically, the average absorption rate for LLaMA 3B is 1.190, while Gemma 2B achieves 0.628. This metric is defined as

$$\text{Absorption Rate} = \frac{\text{Student Peak Gain}}{\text{Teacher Base Score}}$$

where Student Peak Gain is the maximum chrF improvement over the student's base score across distillation iterations and Teacher Base Score is the teacher's initial chrF score, is averaged across nine evaluation conditions (three languages and three shot settings). Although the Gemma family demonstrates superior absolute chrF scores, supported by a stronger teacher (Gemma 9B), the LLaMA 3B's higher absorption rate suggests it is a more efficient learner, particularly beneficial in resource-constrained distillation scenarios.

6 Conclusion

This work presents *CycleDistill*, a structured and data-efficient framework for enhancing machine translation from low-resource languages to English. By leveraging iterative synthetic data generation and token-level soft distillation, CycleDistill improves translation performance without reliance on large-scale parallel corpora. Experimental results across multiple low-resource Indian languages confirm consistent gains in chrF scores, demonstrating the effectiveness of the approach under varying linguistic and architectural conditions.

The integration of iterative self-distillation with soft distribution-based learning reveals complementary benefits, though performance improvements taper beyond the second iteration, and translation quality remains sensitive to error accumulation, particularly in morphologically rich languages and limited supervision settings. Nevertheless, *CycleDistill* enables both model refinement and compression without relying on large-scale parallel corpora, making it an efficient and scalable solution for low-resource MT and a meaningful contribution to multilingual NLP research.

7 Limitations

Despite the effectiveness of CycleDistill in enhancing translation performance through iterative and soft distribution-preserving distillation, the approach exhibits several notable limitations. Firstly, empirical results demonstrate diminishing marginal improvements beyond the second iteration, with performance frequently plateauing or deteriorating by the third cycle. Secondly, the method relies on synthetic data generated by teacher models, which may introduce compounding translation errors over successive iterations due to self-reinforcement effects. Thirdly, in few-shot scenarios, particularly involving morphologically rich languages such as Malayalam and Bengali, the system suffers significant performance degradation, up to 30 chrF points, largely attributable to increased prompt lengths and consequent loss of contextual coherence. Finally, the current evaluation is limited to three Indic languages and specific model families (Gemma and LLaMA), thereby restricting the generalizability of the findings to other language pairs and model architectures.

8 Acknowledgements

We would like to express our sincere gratitude to Dr. Anoop Kunchukuttan for his unwavering support, invaluable guidance, and insightful contributions throughout the course of this work. We also extend our thanks to EkStep Foundation and Nilekani Philanthropies for their generous grant, which has supported research efforts at AI4Bharat. We acknowledge the use of the Unsloth (Han et al., 2023), which was instrumental in enabling rapid experimentation initially and proof-of-concept development.

References

Mazida Akhtara Ahmed, Kishore Kashyap, Kuwali Talukdar, and Parvez Aziz Boruah. 2023. Iterative back translation revisited: An experimental investigation for low-resource english assamese neural machine translation. In *ICON*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*. Published 12/2023, reviewed on OpenReview: https://openreview.net/forum?id=vfT4YuzAYA.

Google. 2024. Gemma 2: Next-generation open models from google. https://ai.google.dev/gemma/. Accessed: 2025-05-17.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1700–1722.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Daniel Han, Michael Han, and Unsloth team. 2023. Unsloth.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations (ICLR)*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. *arXiv preprint arXiv:1806.04402*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yunsu Kim, Jaesong Lee, Jooyoul Lee, and Hermann Ney. 2021. Improving low-resource neural machine translation with iterative back-translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Parameswari Krishnamurthy, Ketaki Shetye, and Abhinav PM. 2024. MTNLP-IIITH: Machine translation for low-resource indic languages. In *WMT*.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Anoop Kunchukuttan and 1 others. 2023. The indicnlp corpus: A large-scale multilingual corpus for indic languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gong-Xu Luo, Ya-Ting Yang, Rui Dong, Yan-Hong Chen, and Wen-Bo Zhang. 2020. A joint back-translation and transfer learning method for low-resource neural machine translation. *Scientific Programming*, 2020.

Meta. 2024. Llama 3: Open foundation and instruction models. `https://llama.meta.com/llama3`. Accessed: 2025-05-17.

Subhajit Mukherjee and Mitesh M. Khapra. 2021. Distilling large-scale teacher models into compact student models for neural machine translation. *Transactions of the Association for Computational Linguistics*, 9:459–474.

Maja Popović. 2015. chrf: Character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT)*.

Xinyi Tan, Longyue Wang, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

Maali Tars, Andre Tättar, and Mark Fišel. 2021. Extremely low-resource machine translation for closely related languages. In *EAMT*.

Menan Velayuthan, Nisansa De Silva, and Surangika Ranathunga. 2025. Encoder-aware sequence-level knowledge distillation for low-resource neural machine translation. In *Proceedings of the Eighth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2025)*, pages 161–170, Albuquerque, New Mexico, U.S.A. Association for Computational Linguistics.

Yuanchi Zhang, Peng Li, Maosong Sun, and Yang Liu. 2023. Continual knowledge distillation for neural machine translation. In *ACL*.

A Appendix A: Visualization of the Effects of Our Methods Across Shot Settings

This appendix presents visual analyses illustrating the effect of our proposed methods under different shot configurations. Figures 7–11 show how performance characteristics evolve as the number of shots increases, providing a clearer understanding of the behavior and effectiveness of our approach.

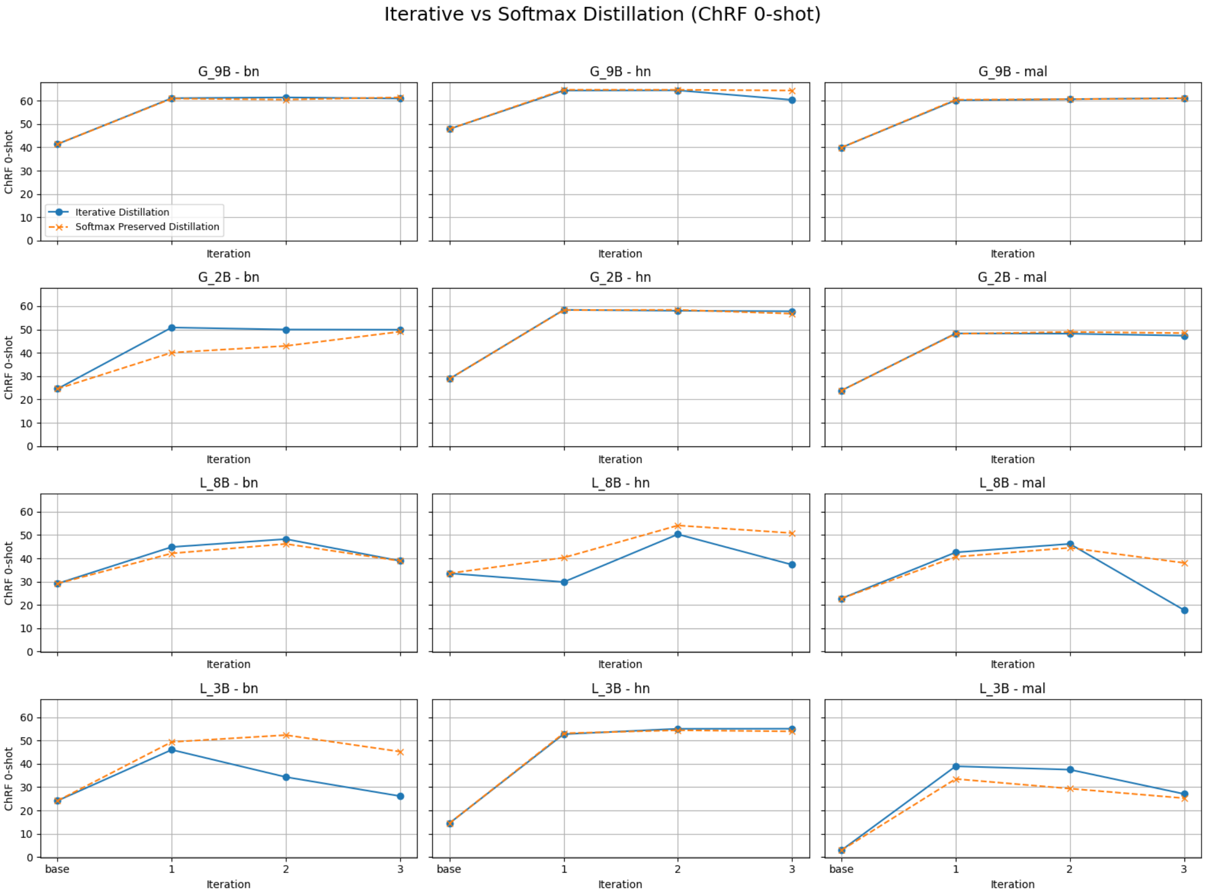


Figure 7: Comparison of methods in the 0-shot setting.

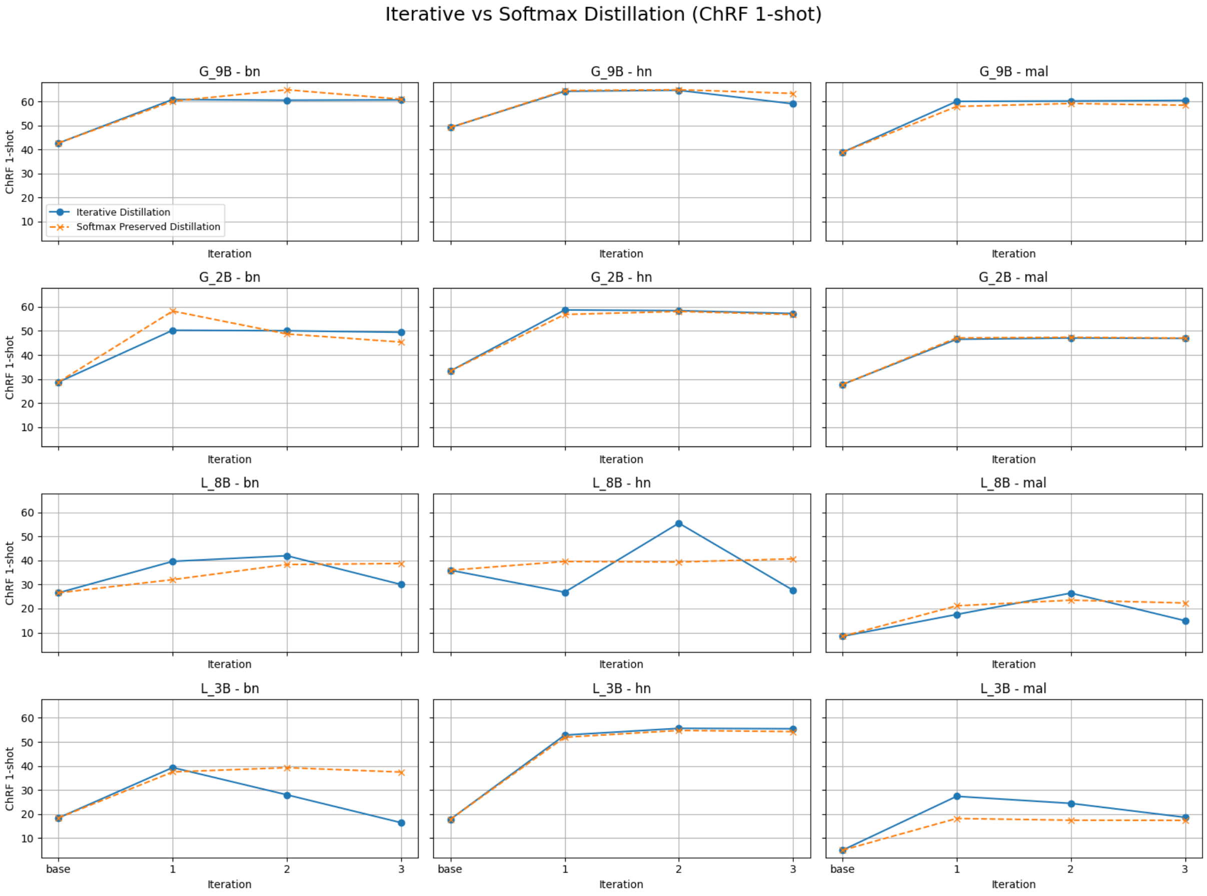


Figure 8: Comparison of methods in the 1-shot setting.

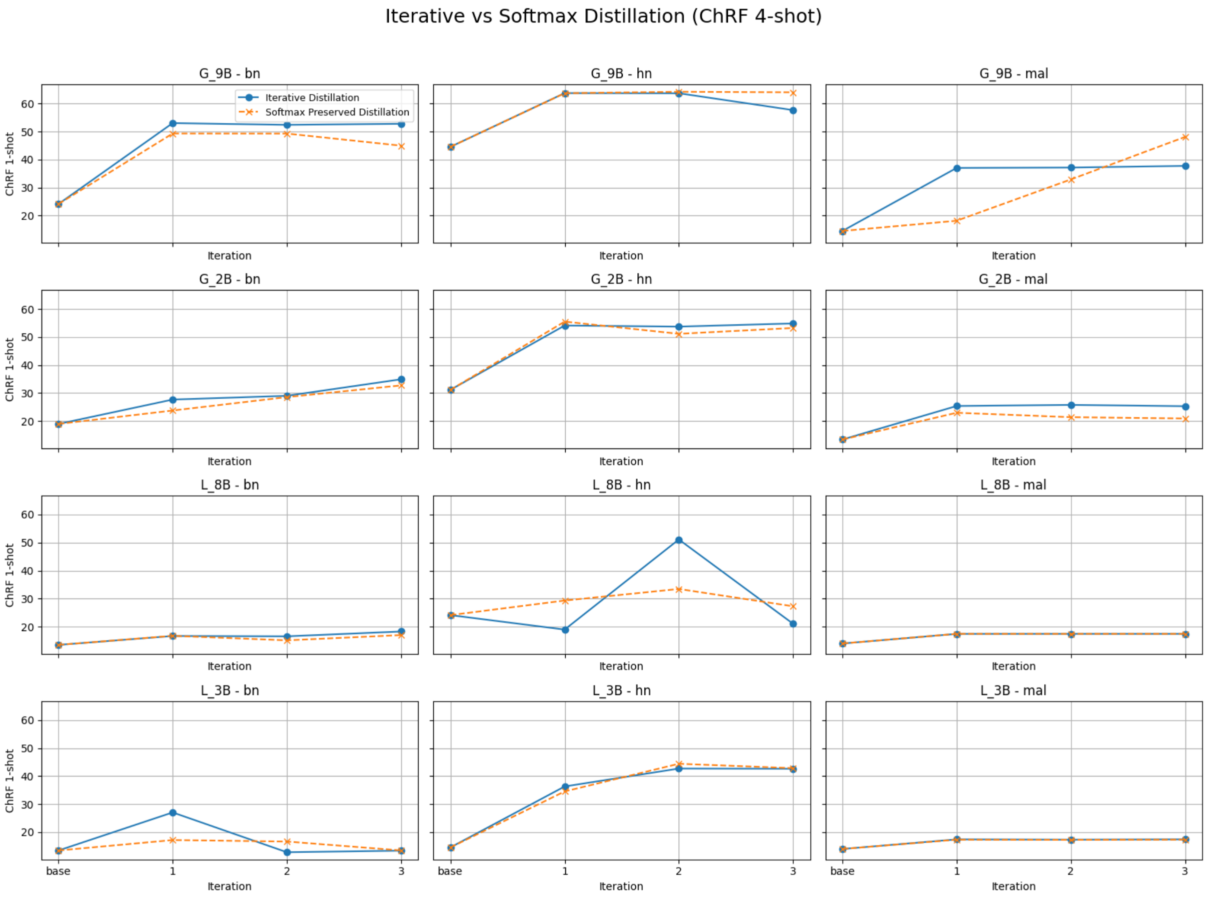


Figure 9: Comparison of methods in the 4-shot setting.

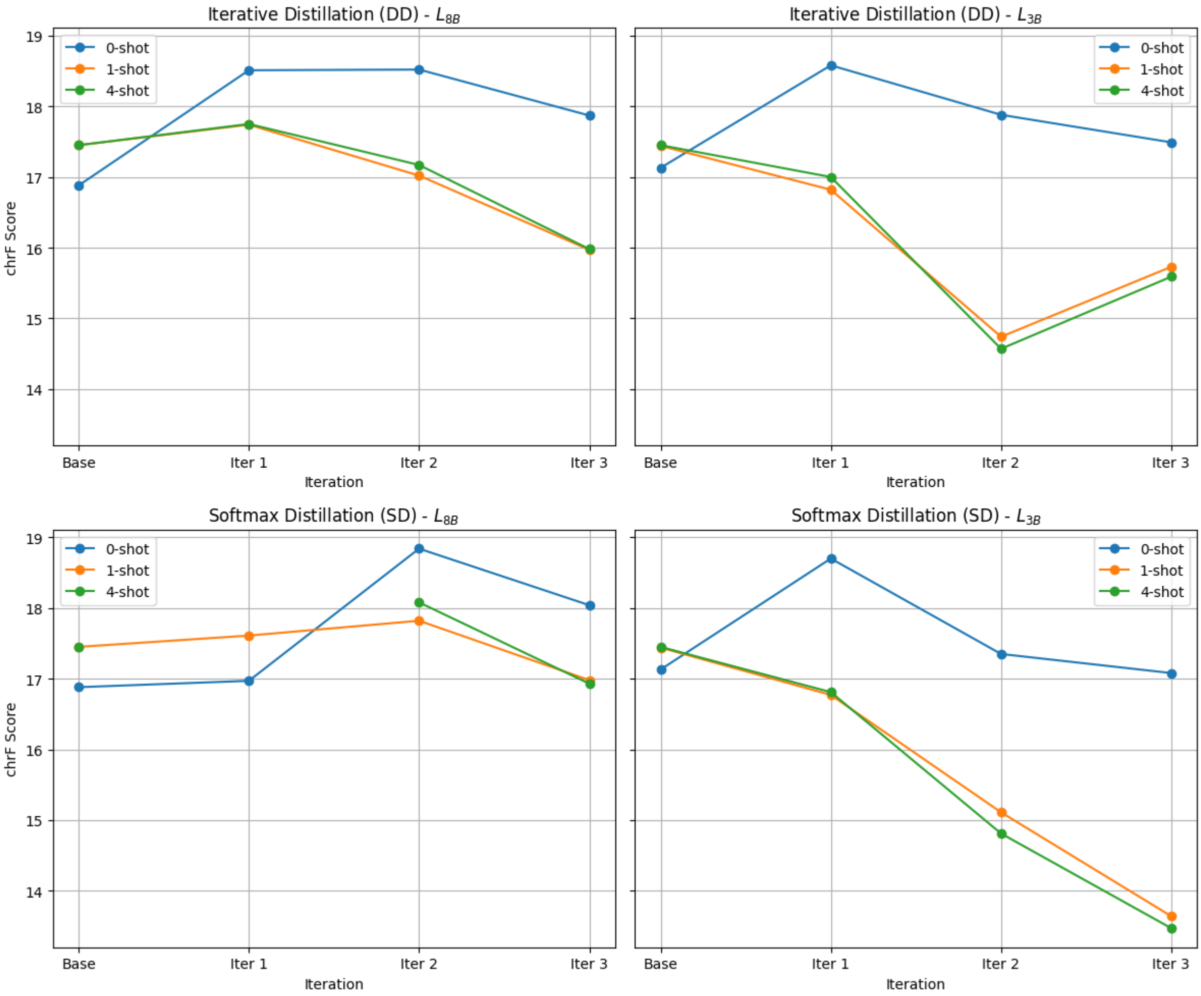


Figure 10: Comparison of methods on the Manipuri dataset.

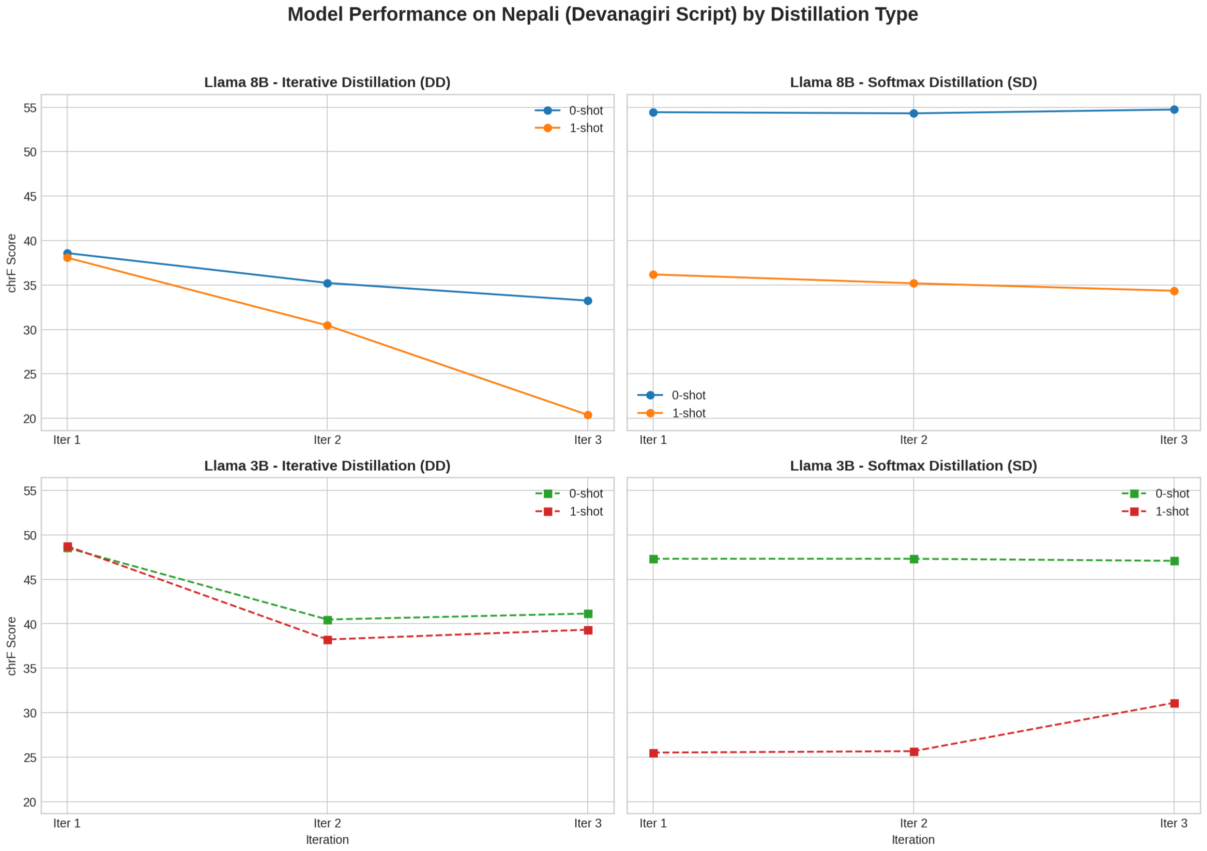


Figure 11: Comparison of methods on the Nepali dataset.

Findings of the WAT 2025 Shared Task on Japanese–English Article-level News Translation

Naoto Shirai, Kazutaka Kinugawa, Hitoshi Ito, Mino Hideya and Yoshihiko Kawai
NHK Science and Technology Research Laboratories
1-10-11 Kinuta, Setagaya-ku, Tokyo, Japan
{shirai.n-hk, kinugawa.k-jg, itou.h-ce, mino.h-gq, kawai.y-lk}@nhk.or.jp

Abstract

We present the preliminary findings of the WAT 2025 shared task on document-level translation from Japanese to English in the news domain [1]. This task focuses on translating full articles with particular attention to whether translation models can learn to produce expressions and stylistic features typical of English news writing, with the aim to generate outputs that resemble original English news articles. The task consists of three translation styles: (1) literal translation, (2) news-style translation, based on English articles edited to match Japanese content, and (3) finalized translation, the primary goal of this shared task. Only one team participated and submitted a system to a single subtask. All tasks were evaluated automatically, and one task was also evaluated manually to compare the submission with the baseline.

1 Introduction

Recent advances in large language models (LLMs) have shown strong potential to improve document-level translation (Wang et al., 2023). Several methods have been proposed to ensure document-level consistency, including context-aware prompting (Cui et al., 2024), fine-tuning (Wu et al., 2024), and agent based approaches (Wang et al., 2025).

In the domain of news translation, translators may need to move beyond fidelity to the source and consider the needs of the target readership (Schäffner, 2012). In practice, this can involve adapting context and expressions to improve clarity for local audiences. In our previous study (Nakazawa et al., 2020; Kinugawa et al., 2024), we constructed sentence-level and document-level news translation data using Japanese and English articles. Building on this foundation, the current shared task explores translation quality at the article level.

In this study, we set up an evaluation using Japanese and English articles published by a Japanese news agency to assess how well existing language models can produce translations that reflect consideration for the target reader. Our task consists of three translation styles: (1) literal translation, (2) news-style translation that preserves the content of Japanese articles while translating them into natural English, and (3) finalized translation, which refers to translation into an original English article. For the third task, which is the main focus of the shared task, we received a submission from one participating team. We evaluated both the baseline and submitted outputs to understand current challenges. The baseline model demonstrated improved performance in document-level BLEU (Papineni et al., 2002) scores when fine-tuned on each dataset. In the third task, a model fine-tuned and optimized with direct preference optimization (DPO) (Rafailov et al., 2023) by the submitting team achieved the highest performance in document-level BLEU scores. This suggests that using original translations as preference data may help to produce translations that are better aligned with reader expectations. However, human evaluation ranked GPT-4o (OpenAI et al., 2024) highest overall, which indicates that challenges remain in translating long articles with deep contextual understanding.

2 Task and Dataset

2.1 Task

This shared task focuses on evaluating the performance of document-level translation models in producing translations that reflect such reader-oriented adaptations when translating Japanese news articles into English. To achieve this, we defined three subtasks:

- **Task 1 Literal Translation**: Literal translation from the Japanese article.

[1] We hosted a shared task titled "Japanese → English: Article-level News Translation Tasks": https://lotus.kuee.kyoto-u.ac.jp/WAT/jiji-corpus/2025/.

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 93–97
December 24, 2025 ©2025 Association for Computational Linguistics

Table 1: Dataset statistics: number of articles ($|D| = 377$), tokens ($|T|$), and sentences ($|S|$), with per-article averages. We consider each headline as a single sentence for each article.

Data Name	$\|T\|$	$\|S\|$	$\|T\|/\|D\|$	$\|S\|/\|D\|$
Original Japanese Article	142,353	4,682	377.59	12.42
Original English Article	129,553	4,475	343.64	11.87
Literal English Translation	137,321	4,747	364.25	12.59
News-style English Translation	144,211	4,888	382.52	12.97

- **Task 2 News-style Translation**: Translation into natural English that preserves the content of the Japanese article.
- **Task 3 Finalized Translation**: Translation into the original English article, which serves as the main objective of this shared task.

Task 2 focuses on producing natural English while preserving the content of the Japanese article, whereas Task 3 aims to match the original English article written by the news agency. For Tasks 2 and 3, we would produce translations that include the dateline (e.g., location and date at the beginning of the article), in line with Jiji Press's English news writing style.

2.2 Dataset Construction

We constructed a dataset consisting of 377 Japanese–English article pairs published by Jiji Press [2] in 2024, each covering the same event. For each Japanese article, we provided two types of English translations: a literal version and news-style version. The dataset included:

- **Original Japanese Article**: Japanese article published by the news agency.
- **Original English Article**: English articles by the news agencies written for an international readership. Task 3 Reference Translation.
- **Literal English Translation**: Translation prioritizing lexical and syntactic fidelity to the original Japanese articles by the translator. Task 1 Reference Translation.
- **News-style English Translation**: Edited translation that reflects English news writing conventions while preserving the content and reporting intent of the Japanese original by the translator. Task 2 Reference Translation.

Hyperparameter	Value
Optimizer	adamw_torch
Learning rate	5e-5
Weight decay	0.01
LR scheduler	cosine
Warmup steps	20
Micro batch size	1
Gradient accumulation steps	1
Epochs	5

Table 2: Hyperparameters used for the SFT models of each task.

The literal and news-style translations were newly created for this task. Translators were instructed to maintain either strict fidelity or stylistic adaptation, depending on the target version. The dataset was split randomly into three subsets: 227 articles for training data, 50 for development data, and 100 for test data. Statistical information [3] for the dataset is shown in Table 1. In addition to the newly constructed data, we also distributed the Jiji2020 dataset [4], which we proposed previously.

3 Approach

3.1 Baseline Models

As baseline systems, we used GPT-4o [5] and Qwen3-8B [6] (Yang et al., 2025), a multilingual model with strong performance in Japanese. For Qwen3-8B, we evaluated both zero-shot inference and supervised fine-tuning (SFT) using 227 training pairs aligned with the expected outputs for each task. The hyperparameters and prompts used in each setting are shown in Tables 2 and 3, respectively. This experiment used four NVIDIA A100 GPUs.

[2] https://www.jiji.com/

[3] We used SpaCy: https://spacy.io/

[4] https://lotus.kuee.kyoto-u.ac.jp/WAT/jiji-corpus/2020/

[5] Version "gpt-4o-2024-11-20" provided by Azure OpenAI.

[6] https://huggingface.co/Qwen/Qwen3-8B

Table 3: Prompts used for each task.

Task 1 Literal Translation Translate the following Japanese news article into English. The output should consist of a headline, followed by a newline, then a body. Do not use extra line breaks or markdown symbols. + [Original Japanese Article]
Task 2 News-style Translation Translate and edit the following Japanese news article into English. The output should consist of a headline, followed by a newline, then a body starting with an appropriate dateline (e.g., “Tokyo, Jan. 1 (Jiji Press)–”). Rephrase and restructure the article. Do not use extra line breaks or markdown symbols. + [Original Japanese Article]
Task 3 Finalized Translation Translate and edit the following Japanese news article into English. The output should consist of a headline, followed by a newline, then a body starting with an appropriate dateline (e.g., “Tokyo, Jan. 1 (Jiji Press)–”). Rephrase and restructure the article, adjusting the amount of information as needed to match English news style. Do not use extra line breaks or markdown symbols. + [Original Japanese Article]

3.2 Submission: NHK-system for Task 3

NHK-system (Mino et al., 2025) is the only submitted model for Task 3. It was trained with SFT and further optimized using DPO with Low-Rank Adaptation (LoRA) (Hu et al., 2021). In this setup, translations resembling literal or news-style outputs were considered as negative examples, whereas Original English Article were preferred. This approach aimed to improve alignment with English news writing. The system was implemented using the Qwen3-8B model.

Task 1: Literal Translation		
Model	**Method**	**d-BLEU**
GPT-4o	Zero-shot	24.87
Qwen3-8B	Zero-shot	22.46
	SFT	**27.45**
Task 2: News-style Translation		
Model	**Method**	**d-BLEU**
GPT-4o	Zero-shot	17.40
Qwen3-8B	Zero-shot	17.15
	SFT	**21.74**

Table 4: Results of Task 1 (top) and Task 2 (bottom).

4 Evaluation

4.1 Automatic Evaluation

We evaluated all tasks using document-level BLEU (d-BLEU) (Liu et al., 2020), which is based on n-gram matches across the whole document [7].

4.2 Human Evaluation

For Task 3, which received system submissions, we additionally conducted human evaluation. Two evaluators were assigned to each criterion. The model's outputs were compared through blind pairwise evaluation based on the perspectives of Adequacy and Fluency. Each submitted translation was compared with the baseline outputs, and assessed as a win, tie, or loss. The final score for each system was computed as the average of these outcomes across all comparisons.

5 Result

5.1 Results of Tasks 1 and 2

Table 4 shows the results of the automatic evaluation of the baseline for literal translation and news-style translation. Results indicate that learning from the corresponding parallel data improved d-BLEU scores by over 4.5 points. Furthermore, GPT-4o achieved higher scores than Qwen3-8B's zero-shot model.

[7] We used SacreBLEU (Post, 2018): https://github.com/mjpost/sacrebleu.

system	model	method	d-BLEU
Baseline	GPT-4o	Zero-shot	13.33
	Qwen3-8B	Zero-shot	14.09
		SFT	19.54
NHK-system	Qwen3-8B	SFT and DPO	**22.72**

Table 5: Results of Task 3 (finalized translation).

NHK-system	Win	Tie	Lose
vs GPT-4o	5.5 / 13.5	27 / 38.5	**67.5 / 48**
vs Qwen3-8B Zero-shot	14.5 / 19	**43 / 42**	42.5 / 39
vs Qwen3-8B SFT	**47** / 22	40 / **51.5**	13 / 26.5

Table 6: Human evaluation results (Adequacy/Fluency) showing Win/Tie/Lose ratios against different baselines.

5.2 Results of Task 3

Table 5 shows the results of the automatic evaluation for Task 3 of the baseline and NHK-system. The submitted system achieved the highest d-BLEU score against all baselines, outperforming the SFT-only baseline by 3.18 points. This suggests that DPO may help models better align with news-specific style and terminology.

Table 6 shows the human evaluation results for Task 3. This table indicates whether the submitted system outperformed (defined as a win) each baseline. The submitted model achieved scores higher than the SFT-only baseline in Adequacy and obtained comparable results in Fluency. These results also demonstrate the effectiveness of DPO. However, zero-shot models such as GPT-4o and Qwen3-8B significantly outperformed the submitted system. Notably, although GPT-4o achieved lower d-BLEU scores than fine-tuning models, it excelled at producing translations tailored to the target audience. These findings highlight the importance of multifaceted evaluation in document-level translation, especially human evaluation.

6 Conclusion

This paper reports preliminary findings from the Japanese–English news article translation task at WAT2025. The task was designed to evaluate document-level translation capabilities through three subtasks: literal translation, news-style translation, and finalized translation, focusing on whether LLMs can produce translations that resemble articles intended for English-speaking readers. SFT improved performance by approximately 4.5 document-level BLEU points in the literal and news-style subtasks. For the finalized translation, applying DPO in addition to SFT achieved a 3.18-point BLEU improvement over an SFT-only model. Human evaluation indicated that GPT-4o outperformed the baseline, thereby highlighting that improvements in BLEU did not consistently align with human assessments, particularly in adequacy and fluency. Overall, the findings indicate potential benefits of LLM tuning and, in specific cases, DPO for improving certain aspects of translation accuracy, while raising open questions about evaluation criteria and alignment with human assessments in news translation. In future work, we will investigate learning strategies and evaluation frameworks that better capture the requirements of document-level news translation.

Limitations

This study has several limitations. First, the experiments used Japanese–English news articles from a single news agency, restricting the findings to this dataset. Second, the evaluation relied on a single reference, which restricted the ability to capture diverse valid translations and may have biased evaluation metrics toward particular stylistic choices. Third, we used BLEU as an automated evaluation method, but it may not be an appropriate substitute for human evaluation (Mathur et al., 2020). In addition, human evaluation was conducted only in a pairwise manner, so absolute evaluations are also needed. Further exploration is required to understand the relationship between human and automatic evaluations, and to establish appropriate criteria for document-level translation assessment.

Ethical Statements

This study used news articles that were originally published by Jiji Press, Ltd. To protect privacy, all personal names were anonymized through pseudonymization, except for those of public figures.

Acknowledgments

We are deeply grateful to Hidehiro Asaka and Takayuki Kawakami for providing the valuable data used in this research. These research results were obtained from commissioned research (No. 225) by the National Institute of Information and Communications Technology (NICT), Japan. We thank Edanz (`https://jp.edanz.com/home`) for editing a draft of this manuscript.

References

Menglong Cui, Jiangcun Du, Shaolin Zhu, and Deyi Xiong. 2024. Efficiently exploring large language models for document-level machine translation with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10885–10897, Bangkok, Thailand. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Kazutaka Kinugawa, Hideya Mino, Isao Goto, and Naoto Shirai. 2024. Findings of the WMT 2024 shared task on non-repetitive translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 715–727, Miami, Florida, USA. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Hideya Mino, Rei Endo Endo, and Yoshihiko Kawai. 2025. NHK submission to WAT 2025: Leveraging preference optimization for article-level news translation tasks. In *Proceedings of the 12th Workshop on Asian Translation*, Mumbai, India.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, pages 1–44, Suzhou, China. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Christina Schäffner. 2012. Rethinking transediting. *Meta*, 57(4):866–883.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Yutong Wang, Jiali Zeng, Xuebo Liu, Derek Wong, Fandong Meng, Jie Zhou, and Min Zhang. 2025. Delta: An online document-level translation agent based on multi-level memory. In *International Conference on Representation Learning*, volume 2025, pages 15708–15731.

Minghao Wu, Thuy-Trang Vu, Lizhen Qu, George Foster, and Gholamreza Haffari. 2024. Adapting large language models for document-level machine translation. *Preprint*, arXiv:2401.06468.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

NHK Submission to WAT 2025: Leveraging Preference Optimization for Japanese–English Article-Level News Translation Tasks

Hideya Mino, Rei Endo, and Yoshihiko Kawai
NHK Science and Technology Research Laboratories
1-10-11, Kinuta, Setagaya-ku, Tokyo, Japan
{mino.h-gq, endou.r-mm, kawai.y-lk}@nhk.or.jp

Abstract

This paper describes our submission to the Japanese → English: Article-level News Translation shared task as part of WAT 2025. In this shared task, participants were provided with a small but high-quality parallel corpus along with two intermediate English translations: a literal translation and a style-adapted translation. To effectively exploit these limited training data, our system employs a large language model trained via supervised fine-tuning followed by direct preference optimization (DPO), a preference learning technique for aligning model outputs with professional-quality references. By leveraging literal and style-adapted intermediate translations as negative (rejected) samples and human-edited English articles as positive (chosen) samples in DPO training, our model achieved notable improvements in translation quality. We evaluated our approach using BLEU scores and human assessments.

1 Introduction

We describe the system submitted by Team NHK as part of the the Japanese → English Article-level News Translation shared task at WAT 2025 (Shirai et al., 2025). The three shared tasks that were part of this shared task were as follows: Task 1 involved literal English translation of the Japanese articles, Task 2 involved style-adopted translation of the Japanese articles, and Task 3 involved translation into the actually published English articles from the Japanese articles. We participated in Task 3, which focused on article-level translation and required maintenance of coherence, consistency, and stylistic appropriateness beyond individual sentence-level translation. In addition to a limited amount of high-quality parallel data, two supplementary English translations for each Japanese article were provided: a literal translation, and a news-style translation, which contained edits of the literal version adapted for readability and stylistic naturalness. These two versions can be viewed as intermediate drafts that reflect different stages of the editorial translation process. Our approach leveraged these intermediate translations to improve model alignment and translation quality. We adopted a two-stage training process:

1. Supervised fine-tuning (SFT) of a large language model (LLM) on the article-level parallel corpus.

2. Direct preference optimization (DPO) (Rafailov et al., 2023) using preference pairs constructed from the provided translation variants. In DPO training, the reference English articles served as the "chosen" responses, while the literal and news-style translations acted as "rejected" responses.

We report both automatic and human evaluations showing the effectiveness of this approach.

2 System Overview

A unique aspect of this task was the availability of intermediate translation drafts alongside the official reference translations. Given the limited parallel data, we explore methods to leverage this auxiliary information to enhance translation accuracy.

Since the training corpus was too small to build a conventional neural machine translation system, we adopted an LLM fine-tuning approach. We also explored preference-based optimization methods such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and DPO. These alignment methods adjust LLM behavior to better reflect human preferences and have been shown to improve performance across various natural language generation tasks (Ziegler et al., 2019).

Because the literal and news-style translations often contained lexical or syntactic deviations from the final references, they served as ideal "negative examples" for preference-based learning. We em-

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 98–102
December 24, 2025 ©2025 Association for Computational Linguistics

	Article	Token (English)		
		Original	Literal	News-style
Train	227	78,064	82,199	86,970
Development	50	15,713	17,135	17,788
Test	100	35,776	37,987	39,453

Table 1: Corpus statistics.

Learning rate schedule	cosine
Learning rate warmup	50
Sequence length	2048
Optimizer	adamW
Learning rate	5e-5
Weight decay	0.05
Micro batch size	1
Gradient accumulation steps	1
Precision	bfloat 16
Gradient clipping	1.0

Table 2: Hyperparameters for SFT with Qwen3-8B.

Learning rate schedule	cosine
Learning rate warmup	20
Sequence length	2048
Optimizer	adamW
Learning rate	1e-5
Weight decay	0.05
Micro batch size	1
Gradient accumulation steps	1
Precision	bfloat16
Gradient clipping	1.0
LoRA rank	2
LoRA alpha	4
Attention modules	q, v

Table 3: Hyperparameters for DPO with Qwen3-8B.

ployed DPO for its simplicity and training stability, constructing preference data from these preliminary translations.

3 Experimental Setup

3.1 Dataset

We used the article-level corpus provided by the WAT 2025 organizers. The dataset models the editorial workflow for translating Japanese news into English and contains 377 pairs of Japanese and English articles from Jiji Press[1], each accompanied by literal and news-style English translations. We define the following abbreviations:

- **ja_orig.**: Original Japanese article published by Jiji Press.
- **en_orig.**: Original English article also published by the same Jiji Press. This is an English version of the original Japanese article and is intended for an international audience. The content of this article may differ from the Japanese version.
- **en_literal**: Literal English translation of the Japanese article.
- **en_news-style**: A translation of the original English article edited to match the content of the original Japanese article. The order in which information is presented, vocabulary, and number of lines may differ from those of the original Japanese article.

The literal and news-style translations were newly created for this shared task. The literal translations prioritized fidelity, while the news-style versions prioritized fluency and natural English expression. Of these 377 articles, 227 belonged to the training set, 50 belonged to the development set, and 100 belonged to the test set. Table 1 shows the statistics of the corpus.

For SFT, we used (ja_orig., en_orig.) pairs. For DPO, we constructed preference tuples (x, y_r, y_c) defined as follows:

$$(x, y_r, y_c) = \begin{cases} (\text{ja\_orig.}, \text{en\_literal}, \text{en\_orig.}) \\ (\text{ja\_orig.}, \text{en\_news-style}, \text{en\_orig.}), \end{cases}$$

where x is the source article, y_r is the rejected translation, and y_c is the chosen translation.

3.2 Model and Training

We employed Qwen3-8B (Yang et al., 2025) as the base LLM. The training process consisted of two stages. First, we performed full-parameter SFT using the 227 article-level parallel pairs in the

[1] https://lotus.kuee.kyoto-u.ac.jp/WAT/jiji-corpus/2025/

Model	BLEU
GPT-4o	13.33
Zero-shot LLM	14.09
Fine-tuned LLM (SFT only)	19.54
Proposed (SFT + DPO)	**22.72**

Table 4: Official automatic evaluation results.

training set. Second, we applied DPO with *low-rank adaptation* (LoRA) (Hu et al., 2022) using 454 preference pairs constructed as described in Section 3.1.

The hyperparameters used in both stages are summarized in Tables 2 and 3. These configurations were determined based on hyperparameter tuning conducted on the development set. All experiments were carried out on a single NVIDIA A100 GPU.

3.3 Evaluation

Our system was evaluated using both automatic and human evaluations. Based on the official evaluation framework, we compared our system against three baseline systems: GPT-4o[2], zero-shot LLM, and fine-tuned LLM with (ja_orig., en_orig.) parallel data (i.e. SFT Qwen3-8B model).

For the automatic evaluation, the task organizers calculated case-sensitive BLEU (Papineni et al., 2002) scores using SacreBLEU (Post, 2018).

For the human evaluation, the task organizers employed two bilingual evaluators to assess the translation outputs of our system and the three baselines. Evaluation was conducted on 100 test articles through pairwise comparisons, separately measuring *adequacy* (semantic faithfulness) and *fluency* (linguistic naturalness). Each pair of system outputs was judged as a win, tie, or loss for our proposed model.

4 Results

4.1 Automatic Evaluation

Table 4 presents the official BLEU scores for all systems. Our proposed method (SFT + DPO) achieved the highest BLEU score, outperforming both the zero-shot and SFT-only models, thereby demonstrating the effectiveness of preference optimization in improving translation quality.

vs Baseline	Win	Tie	Lose
vs GPT-4o	5.5 / 13.5	27 / 38.5	67.5 / 48
vs Zero-shot LLM	14.5 / 19	43 / 42	42.5 / 39
vs Fine-tuned LLM (SFT only)	47 / 22	40 / 51.5	13 / 26.5

Table 5: Official human evaluation results for adequacy/fluency. Win, tie, and loss indicate the number of evaluations our proposed method won against, tied with, or lost against the baseline method.

4.2 Human Evaluation

Table 5 summarizes the official human evaluation results for adequacy and fluency. The values indicate the number of cases (out of 100) that our proposed model *won* against, *tied* with, or *lost* against each baseline, averaged across two evaluators.

Our proposed model demonstrated mixed performance in human evaluation. While it outperformed the SFT-only baseline in adequacy (47 wins vs 13 losses), indicating that DPO training improved semantic faithfulness, it underperformed in all other assessments. The model was particularly weak in fluency compared to GPT-4o (13.5 wins vs 48 losses) and zero-shot LLM (19 wins vs 39 losses), suggesting that maintaining stylistic naturalness remains a significant challenge with the current approach.

This discrepancy between BLEU and human evaluations aligns with prior observations (Sulem et al., 2018; Mathur et al., 2020) that automatic metrics often poorly capture human-perceived quality, particularly in article-level translation tasks where coherence and stylistic appropriateness play important roles.

5 Related Work

DPO (Rafailov et al., 2023) simplifies RLHF by eliminating reward modeling and directly training on preference pairs. Because of its efficiency and stability, this approach has been widely adopted in various NLP domains (Grattafiori et al., 2024; Wu et al., 2024; Sun et al., 2025).

LLMs can perform many zero- or few-shot tasks (Brown et al., 2020), but instruction or preference fine-tuning further enhances task alignment (Ouyang et al., 2022). Since collecting preference data is easier than implementing fully supervised learning, DPO offers a practical approach for adapting LLMs to domain-specific objectives. DPO (Rafailov et al., 2023) directly optimizes LLMs with preference data by removing an ex-

[2] gpt-4o-2024-11-20 version provided by Azure OpenAI.

tra reward model. We utilized DPO in this work since it is both easy to use and highly effective.

6 Conclusion

We have presented our WAT 2025 submission for Japanese→English article-level news translation. Our system leverages DPO to align LLMs using intermediate translation data as preference signals. Experimental results suggest that incorporating editorial-stage translations as negative examples allows model to achieve higher BLEU scores. Future work includes scaling this approach to handle larger datasets and exploring finer-grained document-level alignment.

Acknowledgments

These research results were obtained from the commissioned research (No. 225) by National Institute of Information and Communications Technology (NICT), Japan.

References

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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 *ICLR*. OpenReview.net.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Naoto Shirai, Kazutaka Kinugawa, Hitoshi Ito, Hideya Mino, and Yoshihiko Kawai. 2025. Findings of the wat 2025 shared task on japanese-english article-level news translation. In *Proceedings of the 12th Workshop on Asian Translation*, Mumbai, India.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Haoxiang Sun, Ruize Gao, Pei Zhang, Baosong Yang, and Rui Wang. 2025. Enhancing machine translation with self-supervised preference data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23916–23934, Vienna, Austria. Association for Computational Linguistics.

Qiyu Wu, Masaaki Nagata, Zhongtao Miao, and Yoshimasa Tsuruoka. 2024. Word alignment as preference for machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3223–3239, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Findings of WAT2025 English-to-Indic Multimodal Translation Task

Shantipriya Parida[†], **Ondřej Bojar**[‡]

[†]AMD Silo AI, Finland; [‡]Charles University, MFF, ÚFAL, Czech Republic

correspondence: shantipriya.parida@amd.com

Abstract

This paper presents the findings of the English-to-Indic Multimodal Translation shared task from the Workshop on Asian Translation (WAT2025). The task featured three tracks: text-only translation, image captioning, and multimodal translation across four low-resource Indic languages: Hindi, Bengali, Malayalam, and Odia. Three teams participated, submitting systems that achieved competitive performance, with BLEU scores ranging from 40.1 to 64.3 across different language pairs and tracks.

1 Introduction

The 12th Workshop on Machine Translation (WAT2025), held in conjunction with IJCNLP AACL 2025, hosted a number of shared tasks that covered various aspects of machine translation (MT).

Multi-modal translation, which involves incorporating non-text sources alongside text input for machine translation, has gained attention in recent years (Specia et al., 2016; Elliott et al., 2016). However, research in this area has focused on European languages such as English, German, French, Czech, and mainly used two datasets: Flickr30k (Young et al., 2014) and MS-COCO (Lin et al., 2014), where the text caption corresponds to the content of the associated image.

We organized the WAT2025 English-to-Indic Multimodal Shared Task for Low-Resource Indic languages. One important difference is that in our setting, the text caption is attached to a rectangular region of the picture and not the picture as a whole. This approach provides an interesting opportunity to consider not only the broader image but also the localized visual context surrounding the described region, which may provide additional cues for more accurate translation.

2 Task and Datasets

In this task, participants were provided with corpora from the Visual Genome dataset in four target languages: Hindi, Bengali, Malayalam and Odia. The specific datasets are: Hindi Visual Genome 1.1 (HVG, Parida et al., 2019)[1] for Hindi; Bengali Visual Genome (BVG, Sen et al., 2022)[2] for Bengali; Malayalam Visual Genome (MVG, Parida and Bojar, 2021)[3] for Malayalam; and Odia Visual Genome (OVG)[4] for Odia. The datasets are split into train, test, dev and challenge test in a parallel fashion. The number of sentences in each split is provided in Table 1. Each split contains items consisting of an image, a highlighted rectangular region within the image $(x, y, width, height)$, the original English caption for this region, and the reference translation in the respective target language. These components are illustrated in Figure 1. Depending on the task track, some of these components serve as the source, while others act as references or competing candidate solutions. The specific tracks for this task are listed below.

2.1 Text-Only Translation

Labeled "TEXT" in the WAT official tables, participants translate short English captions into the target language without using visual information. Additional textual resources are allowed but must be documented in the system description paper.

2.2 Captioning

Labeled with the target language code, e.g., "HI," "BN," "ML," "OD", participants generate captions

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3267

[2] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3722

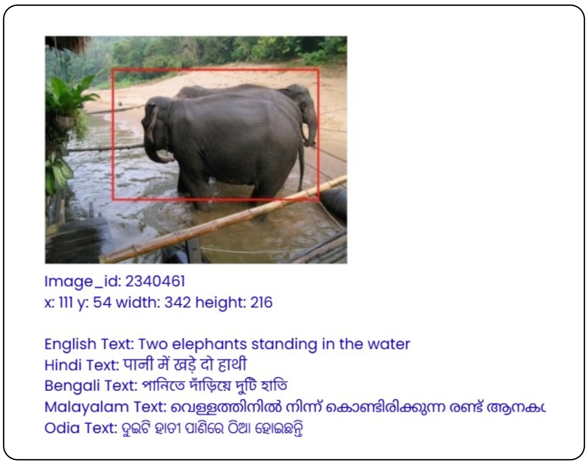
[3] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3533

[4] http://hdl.handle.net/11234/1-5979

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 103–108
December 24, 2025 ©2025 Association for Computational Linguistics

Split	Train	Dev	Test	Challenge
Sentences	28,930	998	1,595	1,400

Table 1: Dataset statistics across all language pairs.



Image_id: 2340461
x: 111 y: 54 width: 342 height: 216

English Text: Two elephants standing in the water
Hindi Text: पानी में खड़े दो हाथी
Bengali Text: পানিতে দাঁড়িয়ে দুটি হাতি
Malayalam Text: വെള്ളത്തിനിൽ നിന്ന് കൊണ്ടിരിക്കുന്ന രണ്ട് ആനക
Odia Text: ଦୁଇଟି ହାତୀ ପାଣିରେ ଠିଆ ହୋଇଛନ୍ତି

Figure 1: Example of a data point showing image ID, region details, source and target languages

in the target language for the highlighted rectangular region in the input image.

2.3 Multi-Modal Translation

Labeled "MM", given an image, a rectangular region within it, and an English caption for that region, participants translate the caption into the target language. Both textual and visual information are available for this task.

3 Evaluation Methods

3.1 Automatic Evaluation

We evaluated translation results by two metrics: BLEU (Papineni et al., 2002), and RIBES (Isozaki et al., 2010). BLEU scores were calculated using SacreBLEU (Post, 2018). RIBES scores were calculated using `RIBES.py` version 1.02.4.[5] All scores for each task were calculated automatically using the corresponding reference translations by the evaluation system through which the participants make their submissions.

Automatic Evaluation System The automatic evaluation system receives translation results by participants and automatically gives evaluation scores to the uploaded results. As shown in Figure 2, the system requires participants to provide the following information for each submission:

- Human Evaluation: whether or not they submit the results for human evaluation;
- Publish the results of the evaluation: whether or not they permit to publish automatic evaluation scores on the WAT2025 web page;
- Task: the task to which the results belong;
- Used Other Resources: whether or not they used additional resources; and
- Method: the type of the method including SMT, RBMT, SMT and RBMT, EBMT, NMT and Other.

Evaluation scores of translation results that participants permit to be published are disclosed via the WAT2025 evaluation web page. Participants can also submit the results for human evaluation using the same web interface. This automatic evaluation system will remain available even after WAT2025.

3.2 Human Evaluation

Due to time constraints, human evaluation was not carried out in WAT2025.

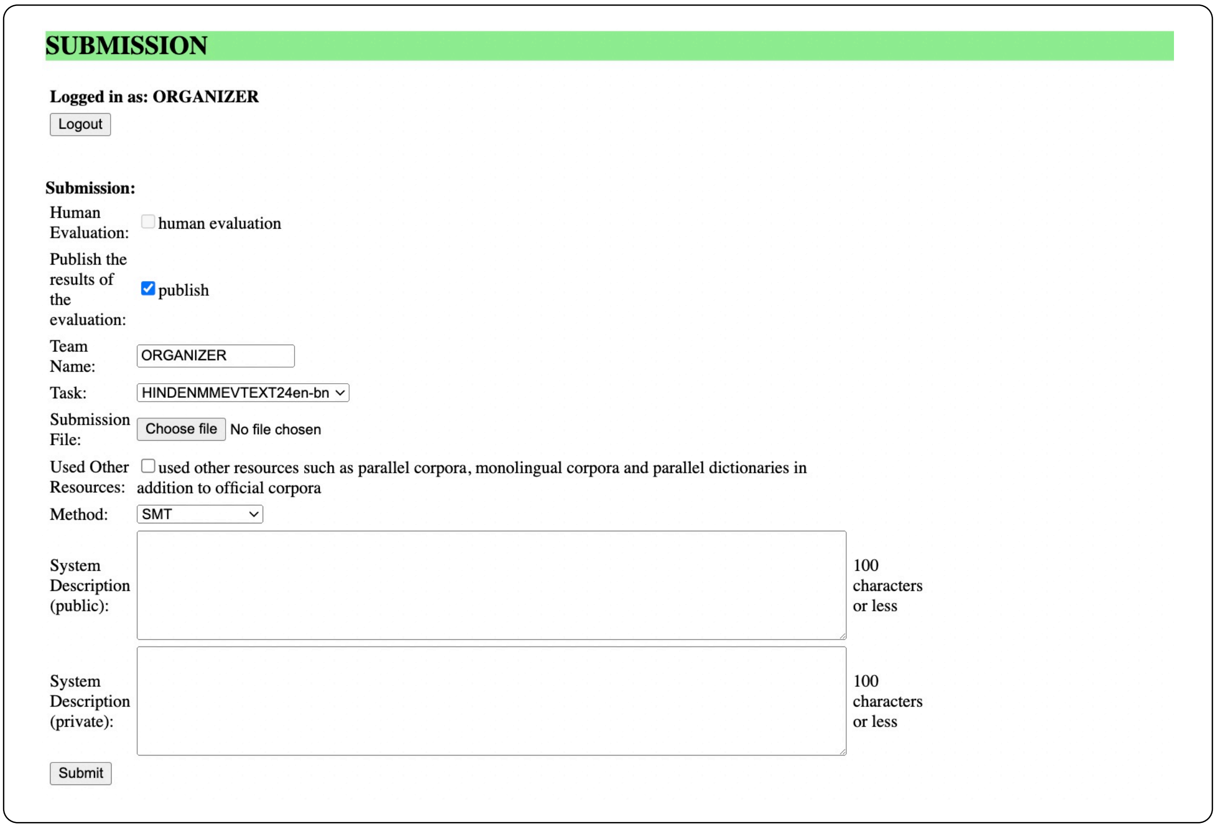
4 Baseline Systems

At WAT2025, we adopted some of the neural machine translation (NMT) as baseline systems. The NMT baseline systems consisted of publicly available software, and the procedures for building the systems and for translating using the systems were published on the WAT web page.

Tokenization The shared task datasets come untokenized, and we did not use or recommend any specific external tokenizer. The standard OpenNMT-py sub-word segmentation was used for pre/post-processing for the baseline system and each participant used what they wanted.

NMT Methods We used the NMT models for all tasks. For the English→Hindi, English→Malayalam, and English→Bengali Multimodal tasks we used the Transformer model (Vaswani et al., 2018) as implemented in OpenNMT-py (Klein et al., 2017) and used the "base" model with default parameters for the multimodal task baseline. We have generated the vocabulary of 32k sub-word types jointly for both the source and target languages. The vocabulary is shared between the encoder and decoder.

[5] `http://www.kecl.ntt.co.jp/icl/lirg/ribes/index.html`



SUBMISSION

Logged in as: ORGANIZER

Logout

Submission:

Human Evaluation:	□ human evaluation	
Publish the results of the evaluation:	☑ publish	
Team Name:	ORGANIZER	
Task:	HINDENMMEVTEXT24en-bn	
Submission File:	Choose file No file chosen	
Used Other Resources:	□ used other resources such as parallel corpora, monolingual corpora and parallel dictionaries in addition to official corpora	
Method:	SMT	
System Description (public):		100 characters or less
System Description (private):		100 characters or less

Submit

Figure 2: The interface for translation results submission

5 Participating Teams and Results

We describe the teams' profiles and submissions as described in their respective description papers. Table 2 shows the team IDs, their respective organizations, and countries.

5.1 Systems' Descriptions

IITP-AI-NLP-ML The IITP-AI-NLP-ML team participated in and reported results for both text-only and multimodal translation tracks. For text-only translation, they fine-tuned the IndicTrans model (Bhat et al., 2015) jointly on all four target languages. In the multimodal track, they enhanced IndicTrans with a CLIP-based visual grounding mechanism that selects the most semantically relevant image regions. By computing cosine similarities between text and full or cropped image embeddings, the system automatically integrates the most aligned visual features into the translation pipeline.

OdiaGenAI team participated in and reported results for all text-only translation tracks. They fine-tuned the NLLB-200 3.3B model (NLLB et al., 2022) to support English-to-multilingual translation, specifically targeting low-resource languages: Hindi, Bengali, Malayalam, and Odia. To enhance training, they applied data augmentation using 100K samples from the Samanantar dataset (Ramesh et al., 2022) provided by AI4Bharat.

BLEU Monday team participated in and reported results for the text-only translation for three language pairs: English-Hindi, English-Bengali, and English-Odia. The proposed system uses a two-stage approach: automated training data correction through a vision-augmented judge-corrector pipeline, followed by LoRA-based fine-tuning. The pipeline employs multimodal models to detect and correct translation errors, replacing ambiguous or mistranslated captions using GPT-4o-mini and IndicTrans2.

5.2 Results and Analysis

Automatic evaluation results Tables 3 to 6 present the automatic evaluation results of the submitted systems, indicating that the systems performed competitively against each other. Despite these promising results, participants expressed a need for human evaluations, as shown in subsequent tables. This reflects a common concern

Team ID	Organization	Country
OdiaGenAI	Odia Generative AI	India
BLEU Monday	Indian Institute of Technology Madras	India
IITP-AI-NLP-ML	Indian Institute of Technology Patna	India

Table 2: List of participants who submitted translations for the WAT2025 English-to-Indic Multimodal Translation Task.

among participants who suspect that their systems may outperform the scores they received, underscoring the importance of qualitative assessments in conjunction with automatic metrics.

Lang.	System	ID	Type	RSRC	BLEU	RIBES
en-hi	IITP-AI-NLP-ML	7461	NMT	Yes	56.60	0.872157
en-ml	IITP-AI-NLP-ML	7463	NMT	Yes	38.90	0.749429
en-bn	IITP-AI-NLP-ML	7462	NMT	Yes	47.00	0.815367
en-od	IITP-AI-NLP-ML	7464	NMT	Yes	55.20	0.915999

Table 3: MMCHMM25 submissions.

Lang.	System	ID	Type	RSRC	BLEU	RIBES
en-hi	OdiaGenAI	7485	NMT	Yes	56.90	0.870254
en-hi	IITP-AI-NLP-ML	7471	NMT	Yes	56.10	0.870914
en-hi	BLEU Monday	7500	Other	Yes	54.00	0.864790
en-ml	OdiaGenAI	7483	NMT	Yes	44.20	0.775824
en-ml	IITP-AI-NLP-ML	7473	NMT	Yes	40.30	0.757277
en-bn	OdiaGenAI	7481	NMT	Yes	50.10	0.830882
en-bn	IITP-AI-NLP-ML	7472	NMT	Yes	47.50	0.819714
en-bn	BLEU Monday	7503	Other	Yes	45.60	0.808860
en-od	OdiaGenAI	7487	NMT	Yes	56.40	0.916177
en-od	IITP-AI-NLP-ML	7474	NMT	Yes	55.40	0.916776
en-od	BLEU Monday	7498	Other	Yes	40.10	0.872698

Table 4: MMCHTEXT25 submissions.

Lang.	System	ID	Type	RSRC	BLEU	RIBES
en-hi	IITP-AI-NLP-ML	7456	NMT	No	44.90	0.765514
en-ml	IITP-AI-NLP-ML	7460	NMT	Yes	50.70	0.780907
en-bn	IITP-AI-NLP-ML	7457	NMT	Yes	48.70	0.799718
en-od	IITP-AI-NLP-ML	7459	NMT	Yes	63.50	0.903624

Table 5: MMEVMM25 submissions.

5.3 Key Findings

The results show that:

- Text-only translation generally outperformed multimodal approaches
- Odia achieved the highest BLEU scores (62.9-64.3)
- Malayalam proved most challenging with lower scores (38.9-51.2)
- Data augmentation strategies proved effective across teams

Lang.	System	ID	Type	RSRC	BLEU	RIBES
en-hi	IITP-AI-NLP-ML	7467	NMT	Yes	45.40	0.834985
en-hi	OdiaGenAI	7484	NMT	Yes	45.10	0.831282
en-hi	BLEU Monday	7494	Other	Yes	42.10	0.814804
en-ml	IITP-AI-NLP-ML	7469	NMT	Yes	51.20	0.760801
en-ml	OdiaGenAI	7482	NMT	Yes	43.20	0.708217
en-bn	OdiaGenAI	7480	NMT	Yes	49.50	0.804158
en-bn	IITP-AI-NLP-ML	7468	NMT	Yes	49.50	0.801714
en-bn	BLEU Monday	7496	NMT	Yes	42.00	0.770437
en-od	IITP-AI-NLP-ML	7470	NMT	Yes	64.30	0.906478
en-od	OdiaGenAI	7486	NMT	Yes	62.90	0.903659
en-od	BLEU Monday	7504	Other	Yes	41.60	0.845874

Table 6: MMEVTEXT25 submissions.

5.4 Cross-Track Performance Comparison

Comparing performance across different tracks reveals interesting patterns:

- **Text-only vs. Multimodal**: Text-only systems achieved comparable or better performance than multimodal systems, indicating room for improvement in visual-textual integration methods
- **Language-specific trends**: Odia consistently performed best across all tracks, while Malayalam showed the most variation between different approaches
- **Team strategies**: Teams employing data augmentation and fine-tuning of large pretrained models (NLLB, IndicTrans) achieved the most competitive results

6 Conclusion and Future Directions

This paper presents an overview of the English-to-Indic Resource Multimodal Translation shared tasks at WAT2025. The task attracted strong participation from numerous teams. Out of these, three teams submitted system description papers detailing their approaches and results. In the future, we aim to expand the range of low-resource languages, with a particular focus on multimodal translation, and encourage greater participation from more teams.

Acknowledgements

The shared tasks were supported by AMD Silo AI, Finland; and Charles University, MFF, ÚFAL, Czech Republic;

Ethical Considerations

The authors do not see ethical or privacy concerns that would prevent the use of the data used in the study. The datasets do not contain personal data. Personal data of annotators needed when the datasets were prepared and when the outputs were evaluated were processed in compliance with the GDPR and national law.

References

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Desmond Elliott, Douwe Kiela, and Angeliki Lazaridou. 2016. Multimodal learning and reasoning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Berlin, Germany. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 944–952, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Team NLLB, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Shantipriya Parida and Ondřej Bojar. 2021. Malayalam visual genome 1.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Shantipriya Parida, Ondřej Bojar, and Satya Ranjan Dash. 2019. Hindi Visual Genome: A Dataset for Multimodal English-to-Hindi Machine Translation. *Computación y Sistemas*, 23(4):1499–1505.

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, Belgium, Brussels. Association for Computational Linguistics.

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

Arghyadeep Sen, Shantipriya Parida, Ketan Kotwal, Subhadarshi Panda, Ondřej Bojar, and Satya Ranjan Dash. 2022. Bengali visual genome 1.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199. Association for Machine Translation in the Americas.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

OdiaGenAI's Participation at WAT 2025

Debasish Dhal[*,$], Sambit Sekhar[*], Revathy V. R.[#],
Akash Kumar Dhaka[†], Shantipriya Parida[†]

[*]Odia Generative AI, Bhubaneswar, India
[$]Aptus Data Labs, Bangalore, India
[#]Department of Computer Science, School of Computational and Physical Sciences,
Kristu Jayanti University, Bengaluru, India
[†]AMD Silo AI, Helsinki, Finland

Abstract

This system description paper presents a detailed overview of the model architecture, training procedure, experimental results, and conclusions of the submission from the OdiaGenAI team to the Workshop on Asian Translation (WAT 2025). For this year, we focus only on text-to-text translation tasks for low-resource Indic languages targeting Hindi, Bengali, Malayalam, and Odia languages specifically. The system uses the large language model NLLB-200-3.3B, fine-tuned on large datasets consisting of over 130k rows for each target language. The entire training dataset consists of data provided by the organizers, as in previous years, and augmented by a much larger 100k sentences of data subsampled from the Samanantar dataset provided by AI4Bharat. Our approach achieved competitive BLEU scores on five of the eight evaluation and challenge test submissions.

1 Introduction

Machine Translation (MT) is a long-standing and well-established sub-field within Natural Language Processing dedicated to creating software capable of automatically translating text or speech between languages. Although substantial progress has been made in achieving human-level translation for languages with extensive training corpora, Indic and Asian languages for which much smaller curated corpuses of training data exist still present significant hurdles to existing MT systems and present sufficient scope for improvement (Popel et al., 2020; Costa-jussà et al., 2022). To overcome these challenges and encourage more fruitful research, WAT has served as an open evaluation platform since 2013 (Nakazawa et al., 2020, 2022). While the challenge is multimodal, this year we decided to focus only on the text-to-text translation for the captions present in the dataset ignoring any visual inputs. Just as in the previous yearly submissions, the evaluation of the given translation tasks is conducted using established metrics like Bilingual Evaluation Understudy (BLEU) and Rank-based Intuitive Bilingual Evaluation Scores (RIBES). In this system description paper, we elaborate on our approach to the tasks that we participated in. In comparison to last year, we have added evaluation for Odia while dropping the Hausa language.

- Task 1: English → Hindi (EN-HI) Text only
- Task 2: English → Bengali (EN-BN) Text only
- Task 3: English → Malayalam (EN-ML) Text only
- Task 4: English → Odia (EN-OD) Text only

2 Task Description and Datasets

In addition to the datasets provided by the organizers, for Hindi, Bengali, Odia, and Malayalam, we also used 100k subsampled translation pairs from Samanantar (Ramesh et al., 2022) in the training set, for each of the four languages. As shown in the results section, this was instrumental in improving the results for the fine-tuned models. The training, evaluation and additional challenge splits are detailed in Table 1.

Task 1: English-to-Hindi Translation The organizers provided the HindiVisualGenome 1.1 (Parida et al., 2019)[1] data set (HVG for short). The training part consists of

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3267

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 109–114
December 24, 2025 ©2025 Association for Computational Linguistics

29k English and Hindi short captions of rectangular areas in photos of various scenes and it is complemented by three test sets: development (D-Test), evaluation (E-Test) and challenge test set (C-Test). Our WAT submissions were for E-Test (denoted "EV" in the official WAT tables) and C-Test (denoted "CH" in the WAT tables).

Task 2: English-to-Bengali Translation For this task, the organizers provided BengaliVisualGenome 1.0 dataset (Parida et al., 2021)[2] (BVG for short). BVG is an extension of the HVG dataset which supports Bengali language. The size of training set and validation set is the same as that for HVG.

Task 3: English-to-Malayalam Translation The organizers provided MalayalamVisualGenome 1.0 dataset[3] (MVG for short). MVG is an extension of the HVG dataset for supporting Malayalam, which belongs to the Dravidian language family (Kumar et al., 2017). The dataset size and images are the same as HVG. MVG contains bilingual English–Malayalam segments, see table 1.

Task 4: English-to-Odia Translation The organizers provided OdiaVisualGenome 1.0 dataset[4] (OVG for short). OVG is a visual genome dataset for Odia language.

3 Modelling and Experimental Details

Identical configurations have been used for all text-to-text translation tasks. For EN-BN, EN-HI, EN-ML, EN-OD text-to-text translation tasks, we individually fine-tuned a large language model (NLLB et al., 2022) separately for all four languages. Similar to Shahid et al. (2023), we used a NLLB-200-3.3B model, but this time chose a much larger 3.3B parameter model, increasing the model size by more than a factor of five. NLLB-200 is a Seq2Seq (Sequence to Sequence) model specifically designed to convert sequences from one domain to sequences in another domain. Bilingual translation (e.g., translating a sequence of words from one language to another) is one of the most prominent applications of Seq2Seq models.

3.1 Evaluation

As in previous years, the quality of the translation task is evaluated by using the BLEU (Papineni et al., 2002) and RIBES (Wołk and Koržinek, 2016). BLEU is perhaps the most widely used evaluation metric and has been an industry standard for a while. It is widely believed to have good correlation with human evaluation for many language pairs while being fast and easy to compute. RIBES is another popular metric for translation between languages with a different word order where BLEU has been reported to struggle. SacreBLEU is a more recent and standardized variant of BLEU having helped industry with easier reproducibility after a widescale call (Post, 2018).

3.2 Finetuning

Since training all parameters of this large 3.3B model is prohibitively expensive, only a small fraction (0.38%) of the parameters are actually allowed to be tunable while the majority are kept frozen, meaning that their values remain the same during optimization. This is achieved by using LoRA fine-tuning made available through the `peft` package from Huggingface using the `PeftModel` API. All the fine-tuning runs were executed on 8×AMD Instinct MI250X/MI250 GPUs. Each such GPU unit offers 128GB HBM2e memory with a peak of 362.1 TFLOPS performance using FP16 precision. This computational capacity enabled us to finish each single-language fine-tuning run in approximately eight hours. The hyperparameters used for the fine-tuning runs are presented in Table 4 to facilitate replication.

The training logs for all four runs are presented in figures 1 and 2. The relatively unstable Malayalam-language run (Figure 2) can be attributed to the inherent grammatical complexity of the Dravidian language family. A similar pattern is observed to a smaller extent for the Hindi-language run (Figure 1). We believe that better and higher quality data can improve the performance of the Hindi language. Odia and Bengali-language runs (Figure 2, 1) demonstrate stable training progres-

[2] `http://hdl.handle.net/11234/1-3722`

[3] `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3533`

[4] `http://hdl.handle.net/11234/1-5979`

Set	**Sentences**	**Tokens**			
		Bengali	Hindi	Malayalam	Odia
Train (Organizer) (Parida et al., 2019)	28930	113978	145448	107133	141647
Train (Additional) (Ramesh et al., 2022)	100000	1019973	1814937	694570	1025677
Dev	998	3936	4978	3620	4907
Evaluation	1595	6408	7852	5689	7734
Challenge	1400	6657	8639	6044	8100

Table 1: Statistics of our data used in the English→Bengali, English→Hindi, English→Malayalam and English→Odia text-to-text translation task: the number of sentences and tokens.

Language	**Visual Genome Source**	**Samanantar Source**	**Visual Genome Target**	**Samanantar Target**
Hindi	4.95	16.42	5.03	18.15
Bengali	4.95	11.53	3.94	10.20
Malayalam	4.95	10.19	3.70	6.95
Odia	4.95	11.33	4.90	10.26

Table 2: Average word count for source (English) and target (Indic) sentences across datasets. The word count is calculated by counting the number of words in a sentence, which serves as a proxy for actual token count.

	WAT BLEU		RIBES	
System and WAT Task Label	OdiaGenAI	Best Comp	OdiaGenAI	Best Comp
English→Hindi				
MMEVTEXT21en-hi	45.10	**45.40**	0.831	**0.834**
MMCHTEXT22en-hi	**56.90**	56.10	0.870	**0.870**
English→Bengali				
MMEVTEXT22en-bn	**49.50**	**49.50**	**0.804**	0.801
MMCHTEXT22en-bn	**50.10**	47.50	**0.830**	0.819
English→Malayalam				
MMEVTEXT21en-ml	43.20	**51.20**	0.708	**0.760**
MMCHTEXT22en-ml	**44.20**	40.30	**0.775**	0.757
English→Odia				
MMEVTEXT21en-od	62.90	**64.30**	0.903	**0.906**
MMCHTEXT21en-od	**56.40**	55.40	0.916	**0.916**

Table 3: WAT2025 Automatic and Manual Evaluation Results for English→Hindi, English→Bengali, English→Malayalam and English→Odia text-to-text translation. For each task, we report the scores of our system (OdiaGenAI) alongside those of the best competing submission. The higher score is highlighted in bold. For both metrics, a higher score indicates better performance.

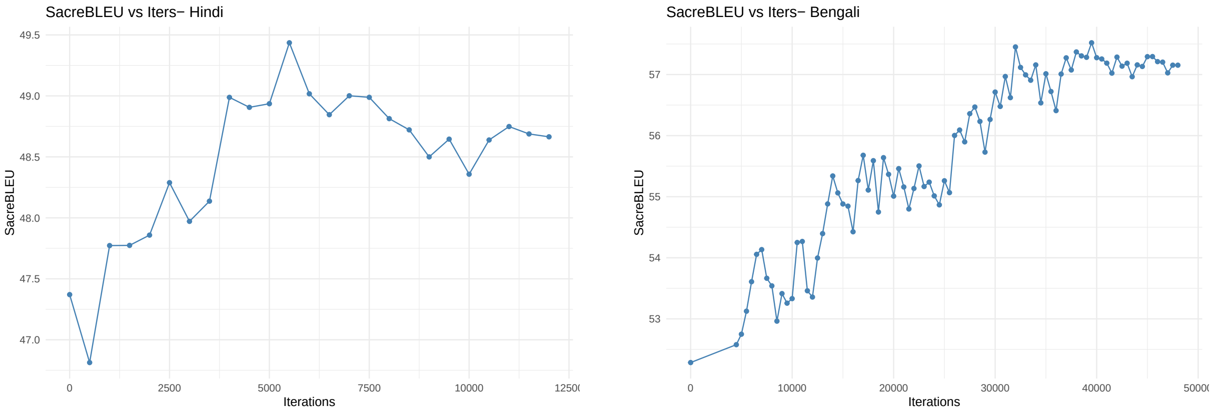


Figure 1: SacreBLEU scores for Hindi and Bengali fine-tuning run.

sion with early convergence, suggesting that extended fine-tuning could yield improved performance. For all four languages, we observe a clear improvement from the starting initial point in the optimization, the highest being for Odia and the lowest for Hindi.

There is still a mismatch in the size of the two components of the final training set. The original dataset provided by the organizers consists of image captions which are short sentences that rarely exceed five words, while the augmented dataset contains many sentences with a higher word count. This case is illustrated in Table 2.

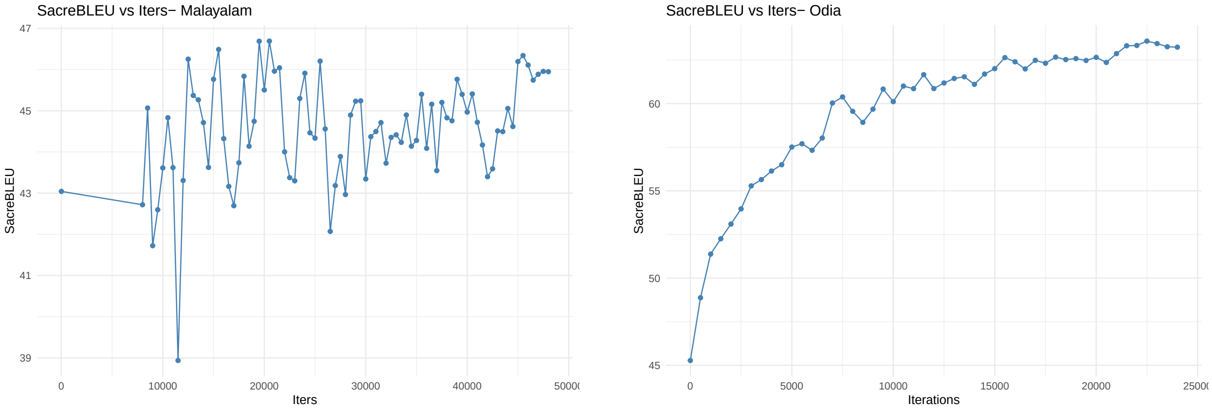


Figure 2: SacreBLEU scores for Malayalam and Odia fine-tuning run.

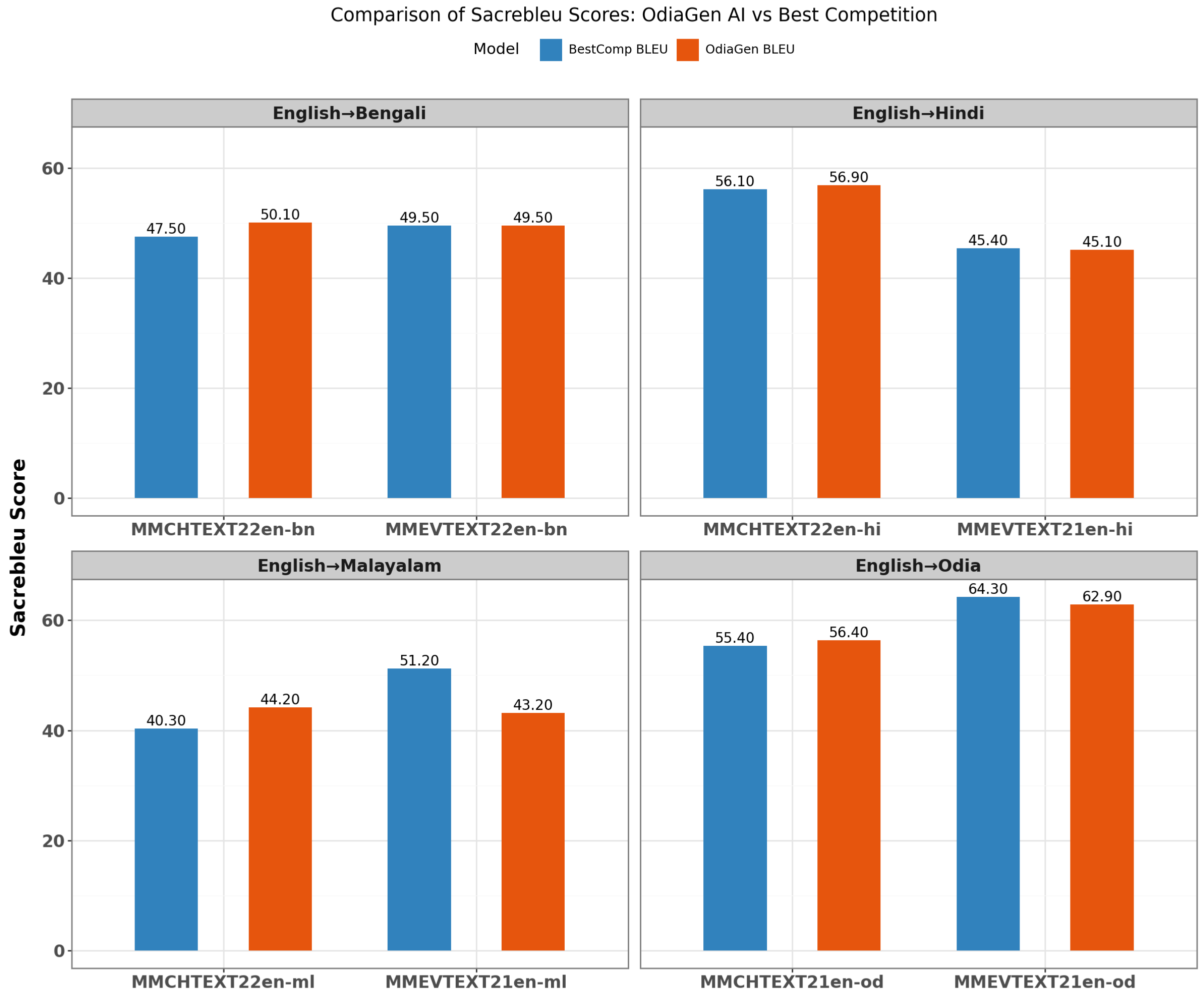


Figure 3: Comparison of our Sacrebleu scores with the best performing team (Source: Table 3).

4 Results

We report the results of the automatic official evaluation after uploading and submitting to the task interface in Table 3, together with the best score attained by the competing submission. Furthermore, we present some selected text samples, translated by our system in Table 5 and do a qualitative analysis. Following the fine-tuning process, these models were used to translate two distinct target test sets for each language: the evaluation set and the challenge set. Translation quality was evaluated using the BLEU score, SacreBLEU,

Hyper Parameter	Value
Learning Rate	$2e^{-4}$
Epochs	3
Cutoff Length	512
Weight Decay	0.01
Warmup Ratio	0.0
max_seq_length	512
LR Scheduler	linear
Lora r	16
Lora α	32
Lora dropout	0.05
use_4bit	False
bnb_4bit_compute_dtype	Not applicable
bnb_4bit_quant_type	None
use_nested_quant	False
per_device_train_batch_size	4 or 8 or 10 or 16
per_device_eval_batch_size	4 or 8 or 10 or 16
gradient_accumulation_steps	1
max_grad_norm	1.0
optim	AdamW
Lora Target Modules	(q_proj, v_proj)

Table 4: Training Hyperparameters.

and RIBES (Ranking by Incremental Bilingual Evaluation System) scores.

For the English-to-Hindi model, a BLEU score of 45.10 was achieved on the evaluation set, while a score of 56.90 was obtained for the challenge set. These results highlight the strong performance of the model and its capacity to handle more complex or unusual translation tasks. The difference between the two scores is 11.8 BLEU points (45.10 vs 56.90) and probably occurs due to a large difference between the two challenge datasets.

In the case of the English-to-Bengali model, a BLEU score of 49.50 and 50.10 were achieved for the evaluation test and challenge sets, respectively. These scores demonstrate strong performance on this task. This indicates a robust overall performance with good generalization and a commendable capability to handle nuanced translations specific to the Bengali language.

BLEU scores of 43.20 and 44.20 were obtained on the evaluation and challenge sets of the Malayalam language, respectively. The best score for the evaluation set of the Malayalam language is 51.20, which is significantly higher than our score.

Our system achieved competitive performance for the Odia language challenge set (56.40), with a BLEU score of 62.90 on the evaluation set. Like the Bengali language, the Odia-language model shows a strong ability for generalized translations.

5 Conclusion

In this system description paper, we presented a system for four text-to-text translation tasks in WAT: (a) English→Hindi, (b) English→Malayalam, and (c) English→Bengali and finally (d) English→Odia text-to-text translation. We released the code through Github for research[5], and the models are released on HuggingFace[6].

These empirical results underscore the effectiveness of the methodology adopted for these MT models. Leveraging a fine-tuned NLLB-200-3.3B model with language-specific Visual Genome datasets provides a robust solution to the MT task for the languages under study: Hindi, Bengali, Malayalam and Odia. The results also pave the way for further enhancements and investigations in the realm of MT.

References

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv e-prints*, pages arXiv–2207.

Arun Kumar, Ryan Cotterell, Lluís Padró, and Antoni Oliver. 2017. Morphological analysis of the Dravidian language family. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 217–222.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Raj Dabre, Shohei Higashiyama, Shantipriya Parida, Anoop Kunchukuttan, Makoto Morishita, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2022. Overview of the 9th workshop on Asian translation. In *Proceedings of the 9th Workshop on Asian Translation (WAT2022)*, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, and 1 others. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, pages 1–44.

[5] https://github.com/shantipriyap/wat2025

[6] https://huggingface.co/collections/OdiaGenAI/wat-2025-finetunedmodels

	Hindi	Bengali	Malayalam	Odia
english-Sentence-1	the orange colored traffic cone	a person wearing a black hat	people on the second level	a water glass on a table
Target-Original	नारंगी रंग यातायात शंकु	একটি কালো টুপি পরা ব্যক্তি	രണ്ടാമത്തെ ലെവലിലെ ആളുകൾ	ଏକ ଟେବୁଲ ଉପରେ ପାଣି ଗ୍ଲାସ |
Target-Translated	नारंगी रंग का यातायात शंकु	একটি কালো টুপি পরা ব্যক্তি	രണ്ടാം നിലയിലെ ആളുകൾ	ଏକ ଟେବୁଲ ଉପରେ ଏକ ପାଣି ଗ୍ଲାସ |
Gloss	the orange colored traffic cone	A person wearing a black hat	people on the second level	a water glass on a table
Remarks (Comparison)	Our translation is more grammatically correct	Both are identical	Our translation is fully translated accurately	Our translation is more grammatically correct
english-Sentence-2	the bird is black	This is a person	the court is dark blue	a person walking on a sidewalk
Target-Original	पक्षी काला है	এটি একজন ব্যক্তি	കോർട്ട് ഇരുണ്ട നീല നിറമാണ്	ରାସ୍ତାରେ ଯାଉଥିବା ଜଣେ ବ୍ୟକ୍ତି |
Target-Translated	पक्षी काला है	এটি একজন ব্যক্তি	കോർട്ട് ഇരുണ്ട നീലയാണ്	ରାସ୍ତାରେ ଯାଉଥିବା ଜଣେ ବ୍ୟକ୍ତି |
Gloss	the bird is black	This is a person	the court is dark blue	A man walking on the road
Remarks (Comparison)	Both are identical	Both are identical	Both are similar	Both are identical
english-Sentence-3	Man wearing military clothes	A stop light	wooden slat that forms back of bench.	Man wearing military clothes
Target-Original	फौजी कपड़े पहने हुए आदमी	একটি স্টপ লাইট	ഒരു വുഡൻ സ്ലാറ്റ് ബെഞ്ചിന്റെ പുറകിൽ രൂപം കൊള്ളുന്നു.	ସାମରିକ ପୋଷାକ ପିନ୍ଧିଥିବା ବ୍ୟକ୍ତି |
Target-Translated	सैन्य कपड़े पहने आदमी	একটি স্টপ লাইট	ബെഞ്ചിന്റെ പുറകിൽ രൂപം കൊള്ളുന്ന മരം സ്ലാറ്റ്.	ସାମରିକ ପୋଷାକ ପିନ୍ଧିଥିବା ବ୍ୟକ୍ତି |
Gloss	Man wearing military clothes	A stop light	Wooden slat that forms the back of the bench.	Man wearing military clothes.
Remarks (Comparison)	Our translation uses a Sanskrit-word for Military, while the target translation uses an Arabic-word.	Both are identical	Our translation is more grammatically correct	Both are identical.

Table 5: Comparison between original translations and our model's translations for English-Malayalam, English-Hindi, English-Bengali, and English-Odia language pairs.

Team NLLB, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Shantipriya Parida, Ondřej Bojar, and Satya Ranjan Dash. 2019. Hindi visual genome: A dataset for multi-modal English to Hindi machine translation. *Computación y Sistemas*, 23(4):1499–1505.

Shantipriya Parida, Subhadarshi Panda, Satya Prakash Biswal, Ketan Kotwal, Arghyadeep Sen, Satya Ranjan Dash, and Petr Motlicek. 2021. Multimodal neural machine translation system for English to Bengali. In *Proceedings of the First Workshop on Multimodal Machine Translation for Low Resource Languages (MMTLRL 2021)*, pages 31–39, Online (Virtual Mode). INCOMA Ltd.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):1–15.

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, Belgium, Brussels. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Sk Shahid, Guneet Singh Kohli, Sambit Sekhar, Debasish Dhal, Adit Sharma, Shubhendra Kushwaha, Shantipriya Parida, Stig-Arne Grönroos, and Satya Ranjan Dash. 2023. OdiaGenAI's participation at WAT2023. In *Proceedings of the 10th Workshop on Asian Translation*, pages 46–52, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Krzysztof Wołk and Danijel Koržinek. 2016. Comparison and adaptation of automatic evaluation metrics for quality assessment of re-speaking. *arXiv preprint arXiv:1601.02789*.

Does Vision Still Help? Multimodal Translation with CLIP-Based Image Selection

Deepak Kumar[*], **Baban Gain**[*], **Kshetrimayum Boynao Singh** and **Asif Ekbal**
Dept. of Computer Science and Engineering, Indian Institute of Technology Patna, India
{deepakkumar1538, gainbaban, boynfrancis}@gmail.com, asif@iitp.ac.in

Abstract

Multimodal Machine Translation aims to enhance conventional text-only translation systems by incorporating visual context, typically in the form of images paired with captions. In this work, we present our submission to the WAT 2025 Multimodal Translation Shared Task, which explores the role of visual information in translating English captions into four Indic languages: Hindi, Bengali, Malayalam, and Odia. Our system builds upon the strong multilingual text translation backbone *IndicTrans*, augmented with a CLIP-based selective visual grounding mechanism. Specifically, we compute cosine similarities between text and image embeddings (both full and cropped regions) and automatically select the most semantically aligned image representation to integrate into the translation model. We observe that overall contribution of visual features is questionable. Our findings reaffirm recent evidence that large multilingual translation models can perform competitively without explicit visual grounding.

1 Introduction

Multimodal Machine Translation (MMT) extends traditional text-only translation by incorporating auxiliary visual information typically an image paired with the source sentence. The motivation behind this integration is that images can provide crucial contextual clues that help resolve linguistic ambiguities and improve translation accuracy. For example, consider the English sentence “The man is standing near the court.” Without additional context, the word “court” could refer to a sports court (e.g., tennis or basketball), or a legal court. A text-only translation model may incorrectly choose one sense based solely on linguistic priors. However, if the corresponding image depicts a tennis court, the visual cue instantly clarifies the intended meaning, guiding the model toward the correct translation in the target language. This exemplifies how visual grounding can disambiguate polysemous words that textual context alone may not fully resolve.

Although several studies have shown that incorporating image information improves translation performance, most prior work trains their MMT models from scratch, learning both textual and visual representations jointly. These models often report improvements over text-only Neural Machine Translation (NMT) systems trained under similar conditions. However, while the relative gains appear significant, the absolute translation scores remain low compared to strong pretrained text-only baselines. Moreover, in many benchmark datasets, intra-sentence textual context is already sufficient to produce correct translations, reducing the actual necessity of visual input. Consequently, it remains unclear whether the observed improvements truly arise from visual grounding or from differences in model training setups.

Another source of debate in MMT lies in the choice of visual input. Given an image, its caption, and a cropped version of the image focused specifically on the captioned region, should the model use the full image or only the cropped area? The full image may offer richer contextual information but might also introduce irrelevant details. Conversely, the cropped image may better correspond to the caption but risk losing broader scene semantics.

To address this challenge, we propose a selective visual alignment approach that automatically chooses the most relevant visual representation for translation. Specifically, we extract CLIP embeddings from both the full and cropped versions of each image and compute their cosine similarity with the corresponding text embedding. The image version that exhibits higher textual similarity is selected and passed to the translation system. Our MMT model integrates these CLIP-based features through a Selective Attention mechanism, which performs cross-attention between the image and text representations, allowing the model to focus on visually aligned information.

We use IndicTrans as our base model a strong pretrained multilingual translation system covering multiple Indic languages such as Hindi, Bengali, Malayalam, and Odia. Interestingly, while our ap-

[*] Equal contribution.

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 115–123
December 24, 2025 ©2025 Association for Computational Linguistics

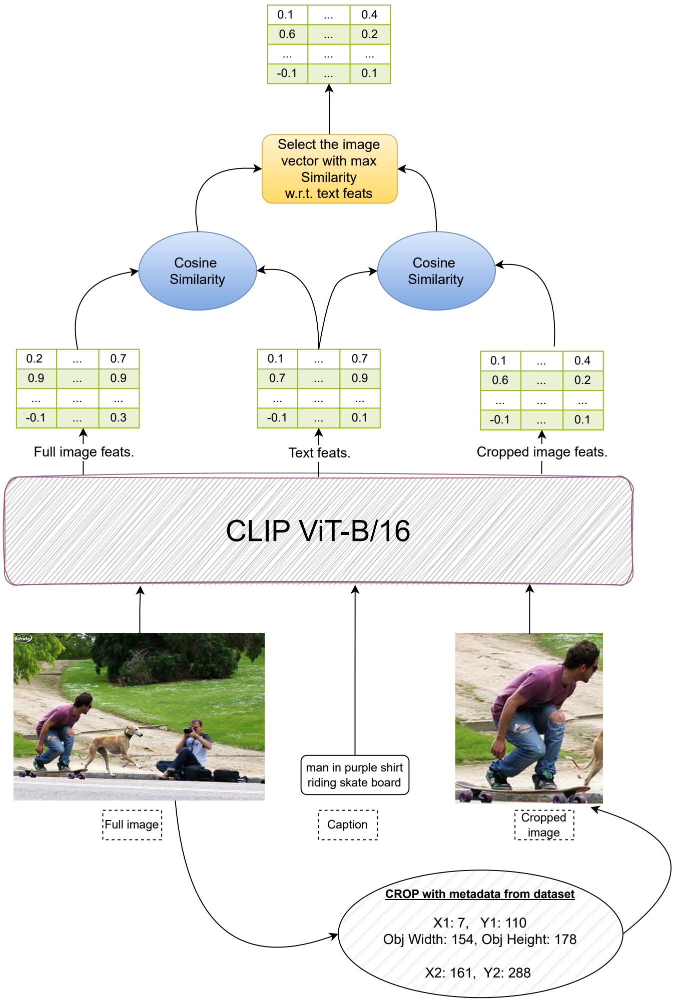


Figure 1: Flow diagram illustrating the image–text similarity computation using CLIP ViT-B/16. The model compares full-image and cropped-image features with text embeddings via cosine similarity, selecting the image vector with the highest similarity score; it is further forwarded to the translation model

proach achieves high absolute translation quality, we find that incorporating visual features does not consistently improve results compared to the text-only baseline. This observation is consistent with previous findings in the literature. Prior work (Li et al., 2022) has also questioned the real contribution of visual information in multimodal translation systems. In this paper, we present the observations in the WAT 2025 Multimodal Translation Task, aiming to further investigate this phenomenon in a competitive shared-task setting. Our system builds upon the Selective Attention architecture (Li et al., 2022; Gain et al., 2025), which effectively integrates visual features into a Transformer-based translation framework. We extend this system in two key ways: (i) we retrain the model to additionally support the *Odia* language, thereby expanding its coverage, and (ii) we incorporate an image selection mechanism that compares the cosine similarity between text embeddings and CLIP features extracted from both the full and cropped versions of each image, forwarding the representation with higher textual alignment. This selective image integration allows the model to better exploit visual cues when relevant, while avoiding unnecessary noise from less informative image regions.

2 Related Works

Multimodal Neural Machine Translation (MMT) seeks to integrate both textual and visual modalities in order to improve translation quality-particularly by helping to disambiguate linguistic phenomena or provide grounding beyond the source text. Early pioneering studies investigated the use of image features (often extracted from convolutional neural networks) alongside an encoder–decoder architecture with attention over both text and image features (Elliott et al., 2016),(Calixto et al., 2017). It was also been observed thaty that MMT systems could leverage visual input under conditions of degraded textual context, but that gains were modest when textual input alone was sufficient (Caglayan et al., 2019).

Subsequent research questioned the actual utility of the visual modality in standard benchmarks, noting that when images were replaced by mismatched or random images, model performance often did not degrade significantly. The authors in (Li et al., 2021) highlighted that existing MMT datasets and architectures might encourage models to ignore the image input altogether. Related work also explored the integration of visual features via fused or hierarchical attention mechanisms (Yao and Wan, 2020) and in low-resource scenarios where the textual signal is weaker.

Multimodal translation in Indian languages has been underexplored, with most studies focusing on the English–Hindi pair. The majority of these works are adaptations of architectures originally designed for high-resource settings.

The earliest work on integrating visual information into Indian language translation can be traced to the approach proposed in (Laskar et al., 2020), which utilized a doubly attentive decoder capable of simultaneously attending to both textual and visual modalities. This model was later refined in (Laskar et al., 2021) through additional text-only pre-training on the IITB parallel corpus (Kunchukuttan et al., 2018) and data augmentation using phrase pairs generated with the Giza++ tool (Marchisio et al., 2022). The visual representations were obtained using a pre-trained VGG19 network (Simonyan and Zisserman, 2015). The same framework was subsequently extended to the English–Bengali language pair in (Laskar et al., 2022), where the model achieved BLEU scores of 43.90 and 28.70 on the Test and Challenge sets, respectively.

Following these early studies, the work presented in (Gupta et al., 2021) introduced an alternative strategy that enriched textual input with object-level visual cues. An object detection model was employed to identify entities within the image, and their class labels were appended to the source sentence to provide additional semantic context. The system, built upon mBART (Liu et al., 2020), achieved state-of-the-art performance for English–Hindi translation; however, the improvement was primarily attributed to large-scale pretraining rather than genuine multimodal fusion. Specifically, the model exhibited a modest gain of +0.52 BLEU on the standard test set while showing a slight decline of 0.06 BLEU on the Challenge set. A subsequent extension of this framework to English–Bengali and English–Malayalam translation was reported in (Parida et al., 2022), yielding comparable trends.

More recent work in (Gain et al., 2021) explored a multimodal transformer architecture for English–Hindi translation. The study demonstrated that focusing on cropped regions of the image corresponding to the textual referents produced more accurate translations than utilizing full-image features. Later, the methodology was revisited in (Shi and Yu, 2022), which introduced refined preprocessing steps such as the removal of duplicate and grayscale images. By employing ResNet50-based features (He et al., 2015) and optimized hyperparameters, this system achieved BLEU scores of 42.29 and 42.70 on the Test and Challenge sets, respectively, highlighting the significant role of data quality and preprocessing in multimodal translation performance.

Overall, while these studies represent significant steps toward integrating visual information in Indian language translation, they collectively indicate that the performance gains from multimodality remain limited. Most improvements appear to arise from better pre-training and data curation rather than from truly leveraging visual grounding.

3 Methodology

3.1 Datasets

The dataset used in this work is part of the WAT 2025 Multimodal Translation shared task (Parida et al., 2024) and is designed to facilitate research on multimodal translation between English and multiple Indic languages. Each data instance comprises an **English caption** paired with its **reference translations** in four target languages: *Hindi* (Parida and Bojar, 2020) , *Bengali* (Sen et al., 2022), *Malayalam* (Parida and Bojar, 2021), and *Odia* (Parida et al., 2025).

In addition to the text pairs, each example is associated with an **image** that visually represents the described scene. To support fine-grained visual

Subset	Sentences	Avg. Src (en)	Avg. Tgt Words			
			hi	bn	ml	or
train	28930	4.95	5.03	3.94	3.70	4.90
test	1595	4.92	4.92	4.02	3.57	4.85
valid	998	4.93	4.99	3.94	3.63	4.92
challenge	1400	5.85	6.17	4.76	4.32	5.79

Table 1: Statistics of the multilingual parallel datasets showing the number of sentences and average word counts for English source and four target languages.

grounding, the dataset also provides **bounding box coordinates** corresponding to the region of interest (ROI) within the image that the caption explicitly refers to.

The multimodal nature of this dataset allows translation models to learn both linguistic mappings and visual alignments, thereby grounding the translation process in contextual image information. Table 1 presents the detailed statistics of the dataset, including the number of sentence pairs and the average word counts for the English source and the four target languages. The corpus contains approximately **29K training examples**, along with dedicated **validation**, **test**, and **challenge** subsets to facilitate comprehensive evaluation.

This multimodal setup provides a valuable benchmark for assessing whether and how visual information contributes to disambiguating textual input during translation, particularly in resource-constrained Indic language settings.

3.2 Experimental Setup

For the multimodal experiments, we enrich the textual input with visual representations extracted from the images paired with the parallel data. To obtain these representations, we use CLIP ViT-B/16 encoder to compute fixed-dimensional embeddings for every image. The encoder outputs a 512-dimensional feature vector that captures high-level semantic attributes relevant for translation. These features are pre-computed offline to avoid additional computational overhead during model training. All models are trained using the Fairseq (Ott et al., 2019) framework and adapted from (Gain et al., 2025) [1], following a consistent configuration across both text-only and multimodal settings to facilitate controlled comparison. Training is carried out using the inverse square root learning rate schedule with 4,000 warm-up steps, Adam optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, and label-smoothed cross-entropy with a smoothing factor of 0.1. We constrain the maximum source and target lengths to 210 tokens and train for up to 20,000 updates with mixed-precision (FP16). A dropout rate of 0.3 is applied uniformly across the model, and early stopping is triggered based on validation loss with a patience of 5 epochs. For the multimodal system, the base Transformer architecture is augmented with an image-aware fusion module, where the pre-extracted CLIP features are injected into the encoder using a shallow self-attention fusion layer. Additional dropout is applied to the image features and the fusion attention to enhance robustness. Apart from this cross-modal extension, all hyperparameters remain consistent with the text-only baseline. Decoding is performed using beam search with a beam width of 5 and a batch size of 64. All predictions undergo standard post-processing, including the removal of subword segmentation markers.

3.3 Visual Feature Extraction

We employ CLIP (Radford et al., 2021), a vision–language model developed by OpenAI, to integrate visual grounding into our translation framework. CLIP learns joint representations of images and text through contrastive learning on large-scale image–caption pairs, mapping both modalities into a shared embedding space where semantic similarity can be effectively measured. This allows the model to capture fine-grained correspondences between visual and linguistic concepts.

In our setup, each data instance contains both a full image and a cropped region specified by bounding box coordinates provided in the dataset. We use CLIP to extract embeddings for both these variants one representing the global context and the other focusing on the region of interest. The text caption accompanying the image is also encoded using CLIP's text encoder, resulting in a dense semantic representation. To determine which visual

[1] https://github.com/babangain/indicMMT/

Model Name	Image Used	Eval. Dataset	Bengali	Hindi	Malayalam	Odia	Average
Textual finetune	✗	Eval Set	49.50	45.40	51.20	64.30	52.60
Multimodal finetune	✓	Eval Set	48.70 (-0.80)	44.90 (-0.50)	50.70 (-0.50)	63.50 (-0.80)	51.95 (-0.65)
Textual finetune	✗	Challenge Set	47.50	56.10	40.30	55.40	49.83
Multimodal finetune	✓	Challenge Set	47.00 (-0.50)	56.60 (+0.50)	38.90 (-1.40)	55.20 (-0.20)	49.43 (-0.40)

Table 2: BLEU Score of our models on different Indic languages from WAT evaluations.

Model Name	Image Used	Eval. Dataset	Bengali	Hindi	Malayalam	Odia	Average
Textual finetune	✗	Eval Set	80.17	83.50	76.08	90.65	82.60
Multimodal finetune	✓	Eval Set	79.97	83.08	76.55	90.36	82.49
Textual finetune	✗	Challenge Set	81.97	87.09	75.73	91.68	84.12
Multimodal finetune	✓	Challenge Set	81.54	87.22	74.94	91.60	83.83

Table 3: RIBES Score of our models on different Indic languages from WAT evaluations.

variant best aligns with the caption, we compute two cosine similarity scores: (a) between the text embedding and the full-image embedding, and (b) between the text embedding and the cropped-image embedding. Since it is not known a priori which of the two visual representations (global or localized) provides more relevant contextual cues, we adopt a simple yet effective heuristic selecting the image feature that yields the higher similarity score with the text. This strategy allows the system to automatically adapt to the most semantically aligned visual cue for each instance, ensuring that the translation model attends to the most meaningful image content while ignoring irrelevant background noise. The overall CLIP-based image selection process is illustrated in Figure 1.

3.4 IndicTrans

IndicTrans (Ramesh et al., 2023) is a multilingual neural machine translation model designed for translation between English and multiple Indic languages. It is trained on large-scale parallel corpora and optimized for high-quality translation across diverse language pairs such as Hindi, Bengali, Malayalam, and Odia. Leveraging a transformer-based architecture and multilingual pretraining, IndicTrans achieves strong performance even in low-resource scenarios, making it a robust baseline for multilingual and multimodal translation research. In this work, we adopt IndicTrans as the underlying translation backbone due to its strong pretraining across Indic languages and its ability to provide robust sentence-level representations and cross-lingual transfer capabilities, making it a suitable foundation for multimodal extensions.

3.5 Model Architecture

We use the Selective Attention architecture (Li et al., 2022) for incorporating visual information into our multimodal translation framework. The model combines the visual features extracted as described in Section 3.3 with the pretrained *IndicTrans* model detailed in Section 3.4. This architecture enables fine-grained alignment between image regions and text tokens through a combination of gated fusion and selective attention mechanisms.

Formally, let the textual input sequence be X_{text} and the corresponding image (either full or cropped, selected via the CLIP-based mechanism) be X_{img}. The *IndicTrans* encoder processes the source text to obtain the hidden representation:

$$H_{\text{text}} = \text{TransformerEncoder}(X_{\text{text}}), \tag{1}$$

while the visual encoder (e.g., ViT) produces image representations:

$$H_{\text{img}} = W \cdot \text{ViT}(X_{\text{img}}), \tag{2}$$

where W is a projection matrix that matches the dimensionality of image and text features.

Following Li et al. (2022), a gated fusion mechanism is used to control the relative contribution of the two modalities:

$$\lambda = \sigma(UH_{\text{text}} + VH_{\text{img}}), \tag{3}$$

$$H_{\text{out}} = (1 - \lambda) \odot H_{\text{text}} + \lambda \odot H_{\text{img}}, \tag{4}$$

where U and V are trainable parameters and σ is the sigmoid activation. The gating variable λ regulates the degree to which visual information influences the textual representation, allowing adaptive fusion based on semantic relevance.

Image			
Source	date when taken in yellow	knife block sitting on counter with knives in it	player running on court
Ground Truth	তারিখ যখন হলুদ নেওয়া হয়	चाकू ब्लॉक में चाकू लेकर काउंटर पर बैठे	कोर्ट पर दौड़ता हुआ खिलाड़ी
Unimodal	তারিখ যখন হলুদ নেওয়া হয়	चाकू ब्लॉक में चाकू लेकर काउंटर पर बैठे	खिलाड़ी कोर्ट पर चल रहा है
Multimodal	হলুদ রঙের সময় তারিখ	इसमें चाकू के साथ काउंटर पर बैठे चाकू ब्लॉक	कोर्ट पर दौड़ता हुआ खिलाड़ी
Explanation	The original text conveys that the date when the photo was taken is depicted in yellow. However, the reference is not reflecting this. Although the “text+image” translation meaning is somewhat accurate, the real improvement is not captured as due to reference.	Unimodal: Awkward and incorrect. Multimodal: Clearer and closer to ground truth, just slightly verbose.	Incorrect verb, says “walking” instead of “running” in case of unimodal

Figure 2: Examples of outputs from unimodal and multimodal model. The major improvements are generally from grammatical issues.

To capture localized visual textual correspondences, the model further applies a Selective Attention layer that correlates textual queries with image patches:

$$H_{\text{img}}^{\text{attn}} = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{5}$$

where Q is derived from H_{text}, and K and V are obtained from H_{img}. The attention weighted visual representation $H_{\text{img}}^{\text{attn}}$ is subsequently used in the gated fusion equations above, ensuring that the model focuses on semantically relevant visual regions while down-weighting background noise.

The resulting fused representation H_{out} is then provided to the *IndicTrans* decoder for translation generation. In summary, our framework directly employs the Selective Attention architecture (Li et al., 2022), integrating it with *IndicTrans* and CLIP-based image selection to ground translation in visually relevant content.

4 Results and Analysis

We evaluated our proposed CLIP-based multimodal translation approach on the English→Indic Multimodal Translation Task using four target languages: Bengali, Hindi, Malayalam and Odia. The results are reported in terms of BLEU and RIBES scores on both the Eval and *Challenge* sets, as shown in Table 2 and Table 3. The *Textual finetune* models correspond to the IndicTrans baseline trained purely on text, while the *Multimodal finetune* models integrate visual features selected through our CLIP-based image–text similarity mechanism.

4.1 Quantitative Evaluation

The BLEU results (Table 2) show that the text-only IndicTrans baseline achieves strong performance across all languages, with average BLEU scores of 52.60 on the *Eval Set* and 49.83 on the *Challenge Set*. Incorporating visual information through CLIP-based multimodal fine-tuning yields small but consistent variations across languages. On the *Eval Set*, multimodal finetuning slightly decreases the average BLEU by 0.65 points, while on the *Challenge Set*, it results in a marginal average reduction of 0.40 points. Interestingly, Hindi demonstrates a minor improvement (+0.50 BLEU) under noisy or out-of-domain conditions, suggesting that visual grounding may be helpful when textual cues are ambiguous or degraded.

For other languages, the observed differences remain within ±1 BLEU, which aligns with prior findings that visual information contributes weakly to translation quality when the text provides sufficient context. Malayalam and Odia, in particular, show small declines, possibly due to the limited correlation between the visual content and sentence semantics in the dataset, leading to minor noise introduction during fusion.

The RIBES results (Table 3) mirror these trends. The textual baseline achieves an average RIBES of 82.60 and 84.12 on the *Eval* and *Challenge* sets, respectively. The multimodal variants record comparable averages of 82.49 and 83.83, indicating no statistically significant degradation. These consistent RIBES values suggest that the inclusion of visual embeddings does not disrupt sentence-level reordering or fluency, even though it provides limited benefits to lexical adequacy.

4.2 Cross-Language Observations

Among all Indic languages, Hindi exhibits the most stable and slightly positive response to multimodal cues, showing improvements in both BLEU (+0.50) and RIBES (+0.13) on the Challenge Set. This is likely due to Hindi's richer contextual grounding in the shared training corpus and its relatively better alignment with English sentence structures. In contrast, Malayalam shows the largest negative shift, consistent with its morphological complexity and looser syntactic alignment, which may hinder effective multimodal fusion.

5 Conclusion

This work presented a systematic investigation of the impact of visual information in multilingual MMT for Indic languages. Building upon the strong text-only IndicTrans model, we proposed a CLIP-based selective visual grounding mechanism that dynamically identifies the most semantically aligned image representation between the full and cropped variants. We observed that visual grounding offers limited gains in translation quality compared to strong text-only baselines. While the absolute BLEU and RIBES scores remain competitive across all languages, the improvements from multimodal finetuning are modest and often lower than text-only model. These findings are consistent with recent studies questioning the necessity of visual input in multimodal translation, particularly when models are pretrained on large-scale textual corpora.

Acknowledgement

The authors express gratitude to the COIL-D Project under Bhashini, funded by MeitY, Govt. of India, for the support and resources that facilitated the successful execution of this research.

References

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4159–4170, Minneapolis, Minnesota. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30K: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74, Berlin, Germany. Association for Computational Linguistics.

Baban Gain, Dibyanayan Bandyopadhyay, and Asif Ekbal. 2021. Experiences of adapting multimodal machine translation techniques for Hindi. In *Proceedings of the First Workshop on Multimodal Machine Translation for Low Resource Languages (MMTLRL 2021)*, pages 40–44, Online (Virtual Mode). INCOMA Ltd.

Baban Gain, Dibyanayan Bandyopadhyay, Samrat Mukherjee, Chandranath Adak, and Asif Ekbal. 2025. Impact of visual context on noisy multimodal nmt: An empirical study for english to indian languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 24(8).

Kshitij Gupta, Devansh Gautam, and Radhika Mamidi. 2021. ViTA: Visual-linguistic translation by aligning object tags. In *Proceedings of the 8th Workshop* *on Asian Translation (WAT2021)*, pages 166–173, Online. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sahinur Rahman Laskar, Pankaj Dadure, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay. 2022. English to Bengali multimodal neural machine translation using transliteration-based phrase pairs augmentation. In *Proceedings of the 9th Workshop on Asian Translation*, pages 111–116, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Darsh Kaushik, Partha Pakray, and Sivaji Bandyopadhyay. 2021. Improved English to Hindi multimodal neural machine translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 155–160, Online. Association for Computational Linguistics.

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, and Sivaji Bandyopadhyay. 2020. Multimodal neural machine translation for English to Hindi. In *Proceedings of the 7th Workshop on Asian Translation*, pages 109–113, Suzhou, China. Association for Computational Linguistics.

Bei Li, Chuanhao Lv, Zefan Zhou, Tao Zhou, Tong Xiao, Anxiang Ma, and JingBo Zhu. 2022. On vision features in multimodal machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6327–6337, Dublin, Ireland. Association for Computational Linguistics.

Jiaoda Li, Duygu Ataman, and Rico Sennrich. 2021. Vision matters when it should: Sanity checking multimodal machine translation models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8556–8562, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Kelly Marchisio, Conghao Xiong, and Philipp Koehn. 2022. Embedding-enhanced giza++: Improving alignment in low- and high- resource scenarios using embedding space geometry. *Preprint*, arXiv:2104.08721.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Shantipriya Parida and Ondřej Bojar. 2020. Hindi visual genome 1.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Shantipriya Parida and Ondřej Bojar. 2021. Malayalam visual genome 1.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Shantipriya Parida, Ondřej Bojar, Idris Abdulmumin, Shamsuddeen Hassan Muhammad, and Ibrahim Said Ahmad. 2024. Findings of WMT2024 English-to-low resource multimodal translation task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 677–683, Miami, Florida, USA. Association for Computational Linguistics.

Shantipriya Parida, Subhadarshi Panda, Stig-Arne Grönroos, Mark Granroth-Wilding, and Mika Koistinen. 2022. Silo NLP‘s participation at WAT2022. In *Proceedings of the 9th Workshop on Asian Translation*, pages 99–105, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Shantipriya Parida, Shashikanta Sahoo, Kalyanamalini Sahoo, Ondřej Bojar, and Satya Ranjan Dash. 2025. Odia visual genome. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2023. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Preprint*, arXiv:2104.05596.

Arghyadeep Sen, Shantipriya Parida, Ketan Kotwal, Subhadarshi Panda, Ondřej Bojar, and Satya Ranjan Dash. 2022. Bengali visual genome: A multimodal dataset for machine translation and image captioning. In *Intelligent Data Engineering and Analytics*, pages 63–70, Singapore. Springer Nature Singapore.

Xiayang Shi and Zhenqiang Yu. 2022. Adding visual information to improve multimodal machine translation for low-resource language. *Mathematical Problems in Engineering*, 2022.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *Preprint*, arXiv:1409.1556.

Shaowei Yao and Xiaojun Wan. 2020. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online. Association for Computational Linguistics.

A Picture is Worth a Thousand (Correct) Captions: A Vision-Guided Judge-Corrector System for Multimodal Machine Translation

Team BLEU Monday

Siddharth Betala[1], Kushan Raj[1], Vipul Betala[2], Rohan Saswade[1]

[1]Indian Institute of Technology (IIT) Madras, [2]Independent

{ betalas5, kushan5711, vipulcbetala, rohansaswade2001 }@gmail.com

Correspondence: betalas5@gmail.com

Abstract

In this paper, we describe our system under the team name *BLEU Monday* for the English-to-Indic Multimodal Translation Task at WAT 2025. We participate in the text-only translation tasks for English-Hindi, English-Bengali, English-Malayalam, and English-Odia language pairs. We present a two-stage approach that addresses quality issues in the training data through automated error detection and correction, followed by parameter-efficient model fine-tuning.

Our methodology introduces a vision-augmented judge-corrector pipeline that leverages multimodal language models to systematically identify and correct translation errors in the training data. The judge component classifies translations into three categories: correct, visually ambiguous (requiring image context), or mistranslated (poor translation quality). Identified errors are routed to specialized correctors: GPT-4o-mini regenerates captions requiring visual disambiguation, while IndicTrans2 retranslates cases with pure translation quality issues. This automated pipeline processes 28,928 training examples across four languages, correcting an average of 17.1% of captions per language.

We then apply Low-Rank Adaptation (LoRA) to fine-tune the IndicTrans2 en-indic 200M distilled model on both original and corrected datasets. Training on corrected data yields consistent improvements, with BLEU score gains of +1.30 for English-Bengali on the evaluation set (42.00 → 43.30) and +0.70 on the challenge set (44.90 → 45.60), +0.60 for English-Odia on the evaluation set (41.00 → 41.60), and +0.10 for English-Hindi on the challenge set (53.90 → 54.00).

1 Introduction

Machine translation (MT) for low-resource languages remains a challenging problem, particularly when dealing with multimodal data where visual context can resolve ambiguities (Specia et al., 2016; Elliott et al., 2016). The Workshop on Asian Translation (WAT) 2025 English-to-Indic Multimodal Translation Task addresses this challenge for four scheduled Indian languages: Hindi, Bengali, Malayalam, and Odia, each with distinct scripts and linguistic characteristics (Parida et al., 2019; Sen et al., 2022; Parida et al., 2024). While recent advances in neural machine translation have shown remarkable progress for high-resource language pairs (Bahdanau et al., 2014; Vaswani et al., 2017), low-resource languages continue to lag behind due to limited parallel corpora, lack of linguistic diversity in training data, and inconsistent translation quality (Sennrich and Zhang, 2019; Costa-Jussà et al., 2022).

Recent research in multimodal machine translation (MMT) has demonstrated that incorporating visual information can significantly improve translation quality, especially for ambiguous terms and culturally-specific content (Ahmed et al., 2025; Elliott et al., 2016; Calixto et al., 2017). The underlying hypothesis is that visual context provides crucial disambiguating cues that align with human cognitive processes of language understanding, which naturally rely on multiple sensory inputs (Beinborn et al., 2018). However, a critical bottleneck in training robust MMT systems for low-resource languages is the quality of parallel training data itself.

Prior work has identified systematic translation errors in the Visual Genome-based datasets used for low-resource MMT tasks (Betala and Chokshi, 2024), where captions often lack proper visual grounding, contain linguistic errors, or exhibit unnatural phrasing that can propagate through trained models. To validate these observations in the context of the WAT 2025 task, one of the authors manually evaluated a sample of the training data. This analysis confirmed numerous quality issues including mistranslations (semantic errors), visual am-

Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025), pages 124–137
December 24, 2025 ©2025 Association for Computational Linguistics

biguities (terms requiring image context for disambiguation), and unnatural expressions that deviate from native speaker conventions (see Figure 2). These findings highlight a fundamental challenge: noisy training data can severely limit the effectiveness of even state-of-the-art neural MT systems.

Building on these findings, we introduce a two-stage approach that systematically addresses data quality before model training. First, we employed a vision-guided judge-corrector system powered by multimodal large language models (LLMs) to automatically identify and fix errors in training captions. Recent research has established the effectiveness of LLM-as-a-judge paradigms for quality assessment across multiple modalities (Zeng et al., 2024; Xiong et al., 2025), demonstrating their ability to provide nuanced evaluations that would traditionally require human annotators. Our judge module leverages visual context to classify each caption into one of three categories: (1) correct translations requiring no modification, (2) incorrect translations where visual context is needed to resolve ambiguities (e.g., distinguishing "dish" as food versus container), or (3) incorrect translations with poor translation quality independent of visual information (e.g., mistranslations, severe grammatical errors, or unnatural phrasing). Based on this classification, we route corrections through specialized mechanisms: a multimodal LLM (GPT-4o-mini[1](Menick et al., 2024)) regenerates captions requiring visual disambiguation, while IndicTrans2 (Gala et al., 2023), a state-of-the-art model for English-to-Indic translation, retranslates cases with pure linguistic errors. This routing strategy enables targeted correction while leveraging the strengths of each approach.

Second, we leveraged the corrected training data to fine-tune IndicTrans2 (Nair et al., 2024) using LoRA (Low-Rank Adaptation) (Hu et al., 2022), a parameter-efficient fine-tuning method (Xu et al., 2023; Han et al., 2024) that has proven effective for adapting large models to specific domains with minimal computational resources (Wong et al., 2024). To rigorously evaluate the impact of data quality on translation performance, we train separate models on both the original and corrected datasets, providing direct evidence of the benefits of automated data cleaning.

Our automated pipeline processes 28,928 training examples across four languages, correcting 19,806 captions in total. On average, 17.1% of captions per language require correction, with rates varying from 12.0% for Odia (3,486 corrections) to 24.0% for Malayalam (6,945 corrections), while Hindi and Bengali show intermediate rates of 16.3% (4,727 corrections) and 16.1% (4,648 corrections), respectively. Of the total corrections, 5,290 (26.7%) require visual context for proper disambiguation, while 14,513 (73.3%) exhibit poor translation quality addressable through retranslation. Experimental results demonstrate that training on corrected data yields consistent improvements across evaluation metrics, with notable BLEU score gains for English-Bengali (+1.30 on evaluation set, +0.70 on challenge set), English-Odia (+0.60 on evaluation set), and English-Hindi (+0.10 on challenge set). To support future research in this area, we make our corrected dataset, judge-corrector pipeline code, and trained models publicly available at `https://github.com/sid-betalol/wat-2025-english2indic-mmt`.

Our main contributions are:

- A vision-guided judge-corrector pipeline that automatically identifies and corrects translation errors in multimodal training data through intelligent routing between visual and linguistic correction strategies
- Comprehensive analysis of error patterns in low-resource MMT datasets, processing 28,928 examples across four languages and revealing that an average of 17.1% of captions per language contain errors, with 26.7% of corrections requiring visual context for proper disambiguation
- Comparative evaluation demonstrating that LoRA finetuning on corrected data yields measurable improvements over training on original data, validating the importance of data quality in low-resource MT

2 Methodology

The overall pipeline of our approach is shown in Figure 1 and the data used for this task is described in Appendix A.

2.1 Preprocessing

We perform two preprocessing steps to prepare the data for our pipeline. First, we combine the language-specific datasets into a unified format where each row contains a unique image identifier,

[1] `https://platform.openai.com/docs/models/gpt-4o-mini`

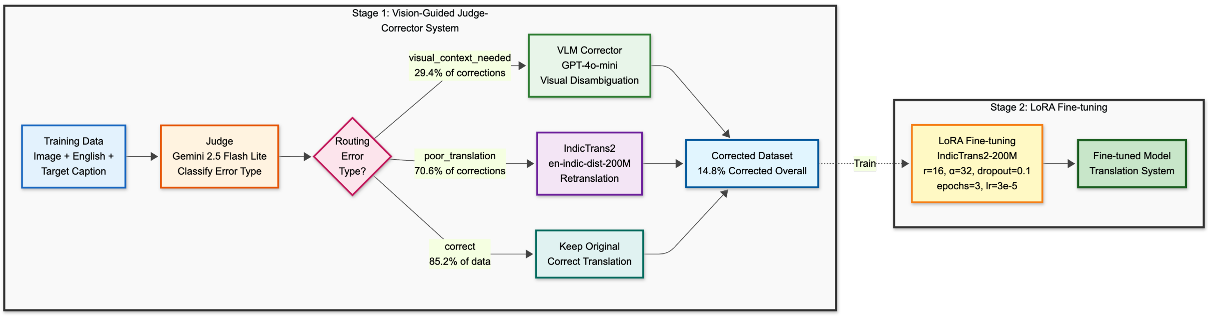

Stage 1: Vision-Guided Judge-Corrector System
Training Data Image + English + Target Caption
Judge Gemini 2.5 Flash Lite Classify Error Type
Routing Error Type?
visual_context_needed 29.4% of corrections
VLM Corrector GPT-4o-mini Visual Disambiguation
poor_translation 70.6% of corrections
IndicTrans2 en-indic-dist-200M Retranslation
correct 85.2% of data
Keep Original Correct Translation
Corrected Dataset 14.8% Corrected Overall
Train
Stage 2: LoRA Fine-tuning
LoRA Fine-tuning IndicTrans2-200M r=16, α=32, dropout=0.1 epochs=3, lr=3e-5
Fine-tuned Model Translation System

Figure 1: Overview of our two-stage approach. Stage 1 uses a vision-guided judge-corrector system to clean the training data, with 14.8% of examples corrected on average across the three languages (Hindi, Bengali, and Odia). Stage 2 applies LoRA fine-tuning to IndicTrans2.

the English caption, and corresponding translations in all four target languages.

Second, we crop the images to their specified bounding box coordinates. The dataset includes images of complete scenes along with coordinates (x, y, width, height) that define rectangular regions corresponding to each caption. We extract these regions to ensure that vision-language models focus on the precise visual content described by the captions, rather than the entire scene.

2.2 Manual Data Quality Assessment

Prior work by Betala and Chokshi (2024) identified systematic quality issues in Visual Genome-based datasets for multimodal machine translation, noting that captions often lack proper visual grounding, contain linguistic errors, and exhibit unnatural phrasing. These observations, made in the context of the WMT2024 English-to-Low Resource Multimodal Translation Task, highlighted a fundamental challenge: noisy training data can severely limit the effectiveness of neural MT systems, even when using state-of-the-art architectures.

Motivated by these findings, we conducted our own manual evaluation to assess whether similar issues were present in the WAT 2025 English-to-Indic datasets. One of the authors, a native Hindi speaker with formal education in Hindi through high school in the Indian education system, systematically reviewed a random sample of 1000 examples from the English-Hindi training set. This manual analysis confirmed the presence of pervasive quality issues and revealed three primary categories of errors:

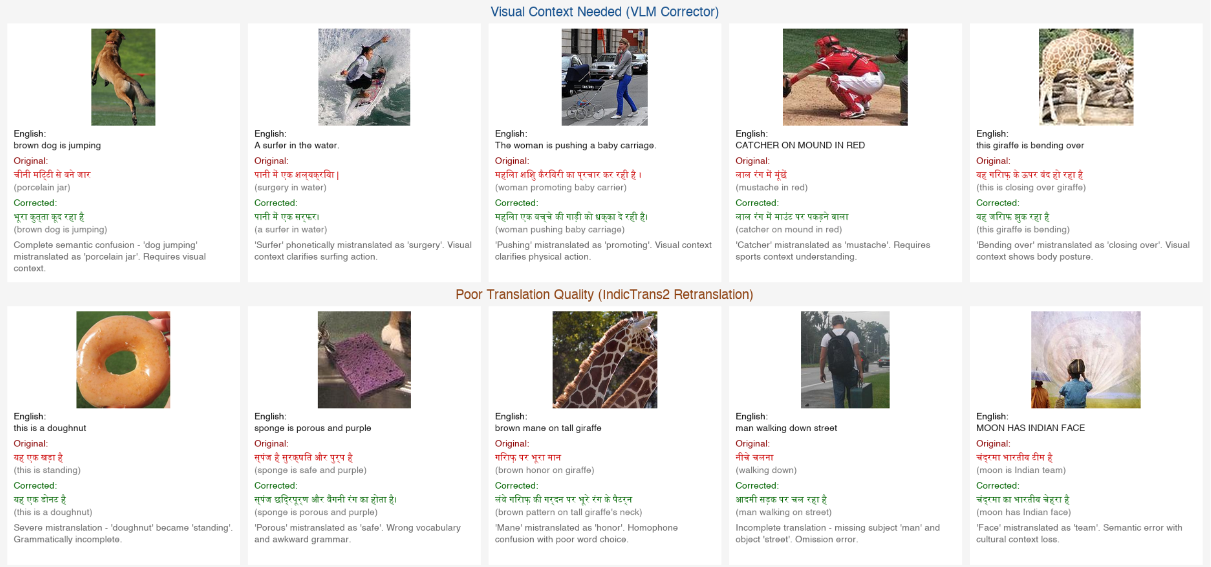
Mistranslations (Semantic Errors): Sampled captions contained clear semantic errors where the Hindi translation did not accurately convey the meaning of the English source. These ranged from minor meaning shifts to complete mistranslations that would mislead a native speaker.

Visual Ambiguities: These captions contained ambiguous terms that required visual context for proper disambiguation. For example, the English word "dish" could refer to either food or a container—a distinction that native speakers would resolve by examining the accompanying image, but which was often incorrectly translated without such visual grounding.

Unnatural Expressions: Some captions exhibited unnatural phrasing that, while potentially understandable, deviated significantly from how native speakers would naturally express the same concept. These included awkward word choices, non-idiomatic constructions, and grammatically correct but stylistically inappropriate formulations.

It is important to note that these categories are not mutually exclusive; many captions exhibited multiple types of issues simultaneously. For instance, a single caption might contain both a mistranslation and unnatural phrasing. Detailed examples of each error category are provided in Figure 2.

This manual analysis validated the concerns raised by Betala and Chokshi (2024) in the WAT 2025 dataset context, revealing substantial quality issues across the training data. While manual correction by native speakers would be ideal, it is prohibitively expensive and time-consuming for a dataset of nearly 29,000 examples per language. These findings motivated our development of an automated judge-corrector system capable of identifying and correcting major translation errors at scale, which ultimately flagged approximately 17% of captions for correction across the four languages, focusing on cases with clear semantic errors or visual ambiguities.



Visual Context Needed (VLM Corrector)

English:
brown dog is jumping
Original:
चीनी मट्टिी से बने जार
(porcelain jar)
Corrected:
भूरा कुत्ता कूद रहा है
(brown dog is jumping)
Complete semantic confusion - 'dog jumping' mistranslated as 'porcelain jar'. Requires visual context.

English:
A surfer in the water.
Original:
पानी में एक शल्यक्रयिा |
(surgery in water)
Corrected:
पानी में एक सर्फर।
(a surfer in water)
'Surfer' phonetically mistranslated as 'surgery'. Visual context clarifies surfing action.

English:
The woman is pushing a baby carriage.
Original:
महलिा शशिु कैरयिरी का प्रचार कर रही है ।
(woman promoting baby carrier)
Corrected:
महलिा एक बच्चे की गाड़ी को धक्का दे रही है।
(woman pushing baby carriage)
'Pushing' mistranslated as 'promoting'. Visual context clarifies physical action.

English:
CATCHER ON MOUND IN RED
Original:
लाल रंग में मूंछें
(mustache in red)
Corrected:
लाल रंग में माउंट पर पकड़ने वाला
(catcher on mound in red)
'Catcher' mistranslated as 'mustache'. Requires sports context understanding.

English:
this giraffe is bending over
Original:
यह गरिाफ़ के ऊपर बंद हो रहा है
(this is closing over giraffe)
Corrected:
यह जरिाफ झुक रहा है
(this giraffe is bending)
'Bending over' mistranslated as 'closing over'. Visual context shows body posture.

Poor Translation Quality (IndicTrans2 Retranslation)

English:
this is a doughnut
Original:
यह एक खड़ा है
(this is standing)
Corrected:
यह एक डोनट है
(this is a doughnut)
Severe mistranslation - 'doughnut' became 'standing'. Grammatically incomplete.

English:
sponge is porous and purple
Original:
स्पंज है सुरक्षति और पुर्प है
(sponge is safe and purple)
Corrected:
स्पंज छद्रिपूर्ण और बैंगनी रंग का होता है।
(sponge is porous and purple)
'Porous' mistranslated as 'safe'. Wrong vocabulary and awkward grammar.

English:
brown mane on tall giraffe
Original:
गरिाफ़ पर भूरा मान
(brown honor on giraffe)
Corrected:
लंबे गरिाफ़ की गर्दन पर भूरे रंग के पैटर्न
(brown pattern on tall giraffe's neck)
'Mane' mistranslated as 'honor'. Homophone confusion with poor word choice.

English:
man walking down street
Original:
नीचे चलना
(walking down)
Corrected:
आदमी सड़क पर चल रहा है
(man walking on street)
Incomplete translation - missing subject 'man' and object 'street'. Omission error.

English:
MOON HAS INDIAN FACE
Original:
चंद्रमा भारतीय टीम है
(moon is Indian team)
Corrected:
चंद्रमा का भारतीय चेहरा है
(moon has Indian face)
'Face' mistranslated as 'team'. Semantic error with cultural context loss.

Figure 2: Training data correction examples from our judge-corrector system. Top row shows cases requiring visual disambiguation (corrected via GPT-4o-mini VLM), bottom row shows poor translation quality (corrected via IndicTrans2 retranslation). Original translations shown in red, corrections in green, with English glosses.

2.3 Data Cleaning Pipeline

Our automated data cleaning pipeline employs a vision-guided judge-corrector architecture that processes each training example through three stages: judgment, routing, and correction. The system is implemented using DSPy (Khattab et al., 2024), a framework for structured prompting that enables type-safe interaction with large language models.

2.3.1 Judge Module

The judge module (Table 5) evaluates each target language caption using a multimodal language model (Gemini 2.5 Flash Lite) that simultaneously analyzes the cropped image region, the English caption, and the target language translation. For each example, the judge produces four outputs:

- **Status**: Binary classification as "correct" or "incorrect"
- **Reason**: If incorrect, categorized as either "visual_context_needed" (ambiguous terms requiring visual disambiguation) or "poor_translation" (linguistic errors independent of visual context)
- **Confidence**: Numerical score between 0 and 1 indicating judgment certainty
- **Explanation**: Brief justification citing the specific issue identified

The judge is explicitly instructed to focus on major issues while ignoring minor stylistic variations such as punctuation differences, optional particles, or alternative word orderings that preserve semantic equivalence. This design choice reduces false positives that would waste computational resources on unnecessary corrections while also preserving already-correct translations that could be degraded by spurious automated interventions, ensuring the pipeline focuses exclusively on substantive quality problems. Missing or empty captions are automatically classified as "visual_context_needed" without invoking the multimodal model, as they unambiguously require regeneration.

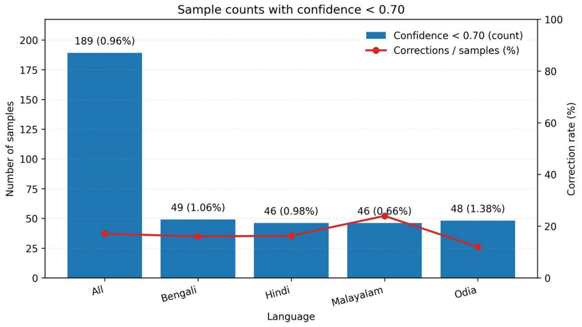


Figure 3: Judge confidence and correction statistics by language. Bars represent the count of training examples where the judge module assigned a confidence score below 0.7, resulting in retention of the original caption. The line plot shows the overall correction rate (percentage of training examples modified) for each language.

To mitigate the impact of uncertain judgments, we implement a confidence threshold: captions flagged as incorrect but with confidence below 0.7 are retained without modification. This conservative approach prevents potentially incorrect corrections in ambiguous cases where the judge's assessment may be unreliable.

2.3.2 Routing and Correction

Based on the judge's classification, captions are routed through one of three paths:

Correct captions ($\sim$83% of examples) are preserved without modification, maintaining the original translation quality where no issues are detected.

Instances labelled as **visual context needed** ($\sim$27% of corrections, $\sim$4.5% of total examples) are processed by GPT-4o-mini, a multimodal language model that regenerates the caption by analyzing both the cropped image and the English source. This approach is specifically designed for cases where ambiguous terms require visual grounding for proper disambiguation. The model is provided with the original (potentially incorrect) caption for reference, but is instructed to prioritize visual evidence when generating the corrected version. The prompt for this module is highlighted in Table 6.

Poor translation cases ($\sim$73% of corrections, $\sim$12.5% of total examples) are retranslated using IndicTrans2 (Gala et al., 2023), a state-of-the-art neural machine translation model specifically trained for English-to-Indic language pairs. This routing strategy leverages IndicTrans2's strong performance on pure translation tasks while reserving the more expensive multimodal LLM for cases where visual context is essential.

2.3.3 Implementation Details

The pipeline processes all four target languages (Hindi, Bengali, Malayalam, Odia) concurrently for each image, with rate limiting to manage API costs and comply with provider constraints.

The pipeline processes all four target languages concurrently for each image, with rate limiting (maximum 4 concurrent API calls) to manage costs and comply with provider constraints. To optimize efficiency, images are loaded once per example and reused across all language evaluations, while automatic checkpointing every 100 examples enables recovery from interruptions.

Based on the parallel corpus statistics as shown in Table 4, the corrector module receives explicit guidance on typical caption lengths for each target language to ensure natural output: Hindi and Odia captions should match English word counts, while Bengali should be approximately 20% shorter and Malayalam 25% shorter. We speculate that these language-specific guidelines might help maintain stylistic consistency with native speaker conventions while preventing unnecessarily verbose or overly terse translations.

2.4 Model Finetuning

Following data cleaning, we fine-tune the IndicTrans2 en-indic 200M distilled model[2] (Gala et al., 2023) using Low-Rank Adaptation (LoRA) (Hu et al., 2022), a parameter-efficient fine-tuning technique that updates only a small subset of model parameters while keeping the base model frozen.

2.4.1 Data Preparation

We prepare the cleaned training data for multilingual fine-tuning by converting it into a unified format suitable for IndicTrans2. Each training example consists of four components: (1) the English source text (`english_caption`), (2) the target language translation (either `{language}_corrected` or `{language}_original` depending on the training configuration), (3) the source language code (`eng_Latn`), and (4) the target language code in FLORES-200 format (Costa-Jussà et al., 2022; nll, 2024) (`hin_Deva` for Hindi, `ben_Beng` for Bengali, `mal_Mlym` for Malayalam, `ory_Orya` for Odia).

The training data is processed through the official IndicTransToolkit processor (Gala et al., 2023), which applies language-specific preprocessing including script normalization and tokenization conventions. We create one training example per language per image, resulting in 115,712 total training examples (28,928 images × 4 languages). The development set follows the same preprocessing pipeline, using the original `{language}_text` columns from the official development split.

2.4.2 LoRA Configuration

We apply LoRA to the attention projection layers (`q_proj` and `v_proj`) of the transformer encoder-decoder architecture (Vaswani et al., 2017). The LoRA configuration uses rank $r = 16$ with scaling factor $\alpha = 32$, resulting in approximately 0.8M trainable parameters compared to the base model's 200M parameters (0.4% of total parameters). We set the LoRA dropout probability to 0.1 to prevent

[2] `https://huggingface.co/ai4bharat/indictrans2-en-indic-dist-200M`

overfitting on the relatively small training set. This parameter-efficient approach enables training on consumer hardware while maintaining competitive performance (Hu et al., 2022).

2.4.3 Training Configuration

Training is conducted using the Hugging Face Transformers library (Wolf et al., 2020) with the Seq2SeqTrainer class. We use the following hyperparameters: per-device batch size of 8 with 2-way gradient accumulation, resulting in an effective global batch size of 32 across 2 GPUs; learning rate of 3×10^{-5} with 500 warmup steps using linear scheduling; weight decay of 0.01; and maximum gradient norm of 1.0 for stability. We train for 3 epochs, which balances convergence with computational efficiency. All experiments use float32 precision to ensure numerical stability across different hardware platforms.

The model is trained in a **multilingual** fashion, where a single model learns to translate from English to all four target languages simultaneously (Aharoni et al., 2019). Each training batch contains examples from all languages, enabling the model to share representations across related Indic languages while learning language-specific translation patterns through the FLORES-200 language codes that prefix each input. This multilingual approach has been shown to improve performance for lower-resource languages through cross-lingual transfer (Arivazhagan et al., 2019).

The training employs standard sequence-to-sequence preprocessing: input sequences are tokenized using the IndicTrans2 tokenizer with a maximum length of 256 tokens, and the DataCollatorForSeq2Seq applies padding to create uniform batch sizes while masking padding tokens in label sequences with -100 to exclude them from loss computation. During inference, we use greedy decoding (beam size 1, 'num_beams=1') as a workaround for known beam search compatibility issues in the IndicTrans2 implementation.

2.4.4 Inference

For inference, we load the trained LoRA adapters and merge them with the base IndicTrans2 model using PEFT's `merge_and_unload()` method (Mangrulkar et al., 2022), eliminating the overhead of adapter routing during generation. Translations are generated using the IndicTransToolkit's preprocessing and postprocessing pipelines to ensure consistency with the model's training format. We translate the evaluation and challenge sets in batches of 16 with a maximum generation length of 256 tokens.

2.4.5 Submitted Systems

Due to resource and time constraints, we submit results for three language pairs: English-Hindi, English-Bengali, and English-Odia. We do not submit results for English-Malayalam.

To rigorously evaluate the impact of data cleaning on translation quality, we submit translations from two systems: (1) a LoRA model trained on the original (uncorrected) training data, and (2) a LoRA model trained on our corrected training data. Both models use identical architectures, hyperparameters, and training procedures, with the only difference being the quality of the training captions. This controlled comparison allows us to directly attribute performance differences to the data cleaning pipeline.

2.5 System Classification

We classify our submissions according to the WAT 2025 task guidelines across four dimensions, as specified by the task organizers (Parida et al., 2024).

First, we participate in the **unconstrained track** due to our use of multimodal large language models, specifically GPT-4o-mini[3] (Menick et al., 2024) and Gemini 2.5 Flash Lite[4] (Comanici et al., 2025), as part of our automated data cleaning pipeline. While these models are not used during inference, their use in training data preparation exceeds the pretrained model restrictions of the constrained track.

Second, our approach is classified as **text-only translation**. Although our data cleaning pipeline leverages visual information to identify and correct translation errors in the training set, the final trained model does not use images during inference. At translation time, the model receives only the English source text as input, without access to the corresponding image or bounding box information.

Third, we are **domain-unaware**, using exclusively the officially provided training data (28,928 examples per language) without incorporating the full English Visual Genome corpus or any additional external datasets. Our data cleaning process operates only on the provided parallel captions, improving their quality without introducing new training examples.

[3] `https://platform.openai.com/docs/models/gpt-4o-mini`

[4] `https://deepmind.google/models/gemini/flash-lite/`

Finally, our system is **multilingual**, employing a single IndicTrans2 model that simultaneously translates from English to all four target languages (Hindi, Bengali, Malayalam, and Odia). Rather than training separate pairwise models for each language pair, our multilingual approach enables cross-lingual transfer and parameter sharing across the related Indic languages (Arivazhagan et al., 2019), while using FLORES-200 language codes to distinguish target languages during generation.

3 Results

We present the results of our two-stage approach: first analyzing the impact of our automated data cleaning pipeline on training data quality, then evaluating how these improvements translate to translation performance on the official evaluation and challenge test sets.

3.1 Data Cleaning Statistics

Our automated judge-corrector pipeline processed all 28,928 training examples across four target languages, identifying and correcting quality issues in 19,806 captions (17.1% of total examples). Table 1 summarizes the correction statistics by language.

The correction rates vary significantly across languages, with Malayalam requiring the most corrections (24.0%) and Odia requiring the fewest (12.1%). This variation likely reflects differences in the original annotation processes for each language dataset (Parida et al., 2019; Sen et al., 2022). Across all languages, the majority of corrections (73.3%, 14,513 out of 19,806) address poor translation quality that can be resolved without visual context, while 26.7% (5,290 corrections) involve visually ambiguous terms requiring multimodal understanding. The low number of missing captions (3 total) indicates that the original datasets were largely complete, with quality issues primarily manifesting as incorrect or unnatural translations rather than absent captions.

3.2 Translation Performance

Table 2 presents the main results comparing models trained on original versus corrected data across three language pairs on both the evaluation set (1,595 examples) and challenge set (1,400 examples). We report BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010) scores, two standard automatic metrics for evaluating machine translation quality.

3.2.1 Impact of Data Cleaning

The results demonstrate that data cleaning yields consistent improvements for English-Bengali across both test sets, with BLEU score gains of +1.30 on the evaluation set (42.00 → 43.30) and +0.70 on the challenge set (44.90 → 45.60). These improvements are substantial given that Bengali had a moderate correction rate (16.1%) in the training data. The challenge set improvements are particularly noteworthy, as this set specifically targets ambiguous cases where visual context is crucial for disambiguation—precisely the type of errors our vision-guided corrector is designed to address.

For English-Odia, we observe a +0.60 BLEU improvement on the evaluation set (41.00 → 41.60), though performance on the challenge set shows a marginal decline of -0.10 points. This mixed result is notable given that Odia had the lowest correction rate (12.1%) among all submitted languages, suggesting that the original Odia training data was already of relatively high quality. The slight performance decrease on the challenge set may indicate that automated correction can occasionally degrade high-quality original translations when the error rate is already low.

English-Hindi shows the smallest improvements, with identical BLEU scores (42.10) on the evaluation set and only +0.10 improvement on the challenge set (53.90 → 54.00). However, we observe consistent RIBES improvements for Hindi on the evaluation set (+0.0021), indicating better word ordering despite similar BLEU scores. The minimal BLEU improvements for Hindi may reflect that this widely-studied language already had relatively clean training data (16.3% correction rate), limiting the potential gains from automated cleaning.

3.2.2 Error Type Analysis

Examining the relationship between correction types and performance gains reveals instructive patterns. Bengali, which showed the largest improvements, had a balanced distribution of error types (31% visual context, 69% poor translation). This suggests that both the vision-guided corrections (handled by GPT-4o-mini) and the text-based retranslations (handled by IndicTrans2) contributed meaningfully to improved model quality. The fact that substantial improvements were achieved despite correcting only 16.1% of the training data underscores the importance of targeting high-impact errors rather than achieving perfect coverage.

Language	Correct	Corrected	Visual	Translation
Hindi	24,201 (83.7%)	4,727 (16.3%)	1,314	3,412
Bengali	24,280 (83.9%)	4,648 (16.1%)	1,436	3,211
Malayalam	21,983 (76.0%)	6,945 (24.0%)	1,507	5,438
Odia	25,442 (87.9%)	3,486 (12.1%)	1,033	2,452
Total	**95,906 (83.0%)**	**19,806 (17.1%)**	**5,290**	**14,513**

Table 1: Data cleaning statistics across four languages. "Visual" indicates corrections requiring visual context (handled by GPT-4o-mini), while "Translation" indicates poor translations (handled by IndicTrans2). Percentages show proportion of total 28,928 examples per language.

Language Pair	Model	Test Set	BLEU	RIBES	Δ BLEU
English-Hindi	Original	Evaluation	42.10	0.815	—
	Corrected	Evaluation	**42.10**	**0.817**	+0.00
	Original	Challenge	53.90	**0.867**	—
	Corrected	Challenge	**54.00**	0.865	+0.10
English-Bengali	Original	Evaluation	42.00	0.759	—
	Corrected	Evaluation	**43.30**	**0.770**	+1.30
	Original	Challenge	44.90	**0.813**	—
	Corrected	Challenge	**45.60**	0.809	+0.70
English-Odia	Original	Evaluation	41.00	**0.847**	—
	Corrected	Evaluation	**41.60**	0.846	+0.60
	Original	Challenge	**40.10**	**0.873**	—
	Corrected	Challenge	40.00	0.870	-0.10

Table 2: Translation performance comparing LoRA finetuning on original versus corrected training data. Bold indicates best performance for each language pair and test set. Δ BLEU shows the improvement (+) or degradation (-) from data correction.

The challenge set results provide evidence for the value of vision-guided corrections, particularly for Bengali which showed consistent gains across both test sets. However, Odia's slight decline on the challenge set highlights an important limitation: automated correction, even with multimodal guidance, cannot perfectly replicate human judgment and may occasionally introduce errors when applied to already high-quality translations. This suggests that automated cleaning provides the greatest benefit for datasets with moderate to high error rates, while datasets with very low error rates (such as Odia at 12.1%) may see diminishing or mixed returns.

3.2.3 Comparison to Full Finetuning Approaches

Our LoRA-based approach represents a parameter-efficient alternative to full finetuning, enabling rapid experimentation and comparison between original and corrected training data. While our results demonstrate clear benefits from data cleaning using LoRA, we note that full finetuning of IndicTrans2 could potentially yield even stronger performance. The IITP-AI-NLP-ML team, which achieved top rankings on multiple leaderboards in this shared task, employed full finetuning of IndicTrans2 across all language pairs. This suggests that the improvements we observe from data cleaning with LoRA likely represent a lower bound on the potential gains, and that combining our data cleaning approach with full finetuning could yield further performance improvements.

3.3 Limitations and Future Work

Test set quality. An important limitation of our evaluation is that we applied data cleaning only to the training set. Since the evaluation and challenge test sets were curated using the same annotation process as the training data, they likely contain similar quality issues—mistranslations, visual ambiguities, and unnatural expressions. The presence of errors in the reference translations could artificially suppress our reported BLEU and RIBES scores, as these metrics penalize deviations from the references even when our model's output may be more accurate or natural than the reference itself. If the test set references were corrected using our pipeline or through manual annotation by native speakers, the true performance of our corrected-data model would likely be higher, and the gap between original and corrected models would be more pronounced. This represents an important direction for future work: applying our data cleaning methodology to create higher-quality evaluation benchmarks for low-resource multimodal translation.

Language coverage. Due to resource and time constraints, we submitted results for only three of the four target languages (Hindi, Bengali, and Odia), omitting Malayalam. Given that Malayalam exhibited the highest correction rate (24.0%) in our data cleaning analysis, it represents a particularly interesting case for future investigation. The substantial number of corrections in Malayalam suggests that this language pair could benefit significantly from our approach, and we encourage future work to validate this hypothesis.

Model capacity. Our experiments focused exclusively on parameter-efficient LoRA finetuning rather than full model finetuning. While this enabled rapid experimentation and fair comparison between original and corrected data, it likely underestimates the full potential of our data cleaning approach. Combining our corrected training data with full finetuning could yield additional performance gains, as evidenced by the strong results achieved by teams employing full finetuning strategies.

3.4 Key Takeaways

Our experimental results validate three main findings:

(1) Data quality significantly impacts translation performance: Across three language pairs, training on corrected data yields consistent improvements or competitive performance compared to original data, with Bengali showing substantial gains (+1.30 BLEU on evaluation, +0.70 on challenge). This demonstrates that automated data cleaning can meaningfully improve translation quality for low-resource languages, even when correcting a relatively small proportion (16-17%) of the training data.

(2) Correction effectiveness varies by initial data quality: Languages with moderate error rates (Bengali: 16.1%) and balanced error distributions benefit most from automated correction, while languages with very low error rates (Odia: 12.1%) show more modest or mixed improvements. This suggests that automated cleaning provides the greatest value for datasets with known quality issues, and that careful analysis of error rates should guide the decision to apply automated correction.

(3) Vision-guided correction addresses a real need: The improvements on the challenge set—specifically designed to test ambiguous cases requiring visual context—validate the core hypothesis that multimodal language models can effectively resolve translation ambiguities that text-only approaches miss. However, the mixed results on some language pairs indicate that automated multimodal correction works best when applied to datasets with moderate error rates rather than already high-quality data. The success on Bengali (+0.70 BLEU on challenge set) demonstrates that when error rates justify intervention, vision-guided correction provides measurable value.

4 Conclusion

We presented a vision-guided judge-corrector system that addresses training data quality in low-resource multimodal translation. Our automated pipeline processed 28,928 examples across four languages, correcting 17.1% of captions through intelligent routing between multimodal LLMs (for visual ambiguities) and IndicTrans2 (for translation errors). LoRA finetuning on corrected data yields measurable BLEU improvements: +1.30 for Bengali (eval), +0.70 (challenge), +0.60 for Odia (eval), and +0.10 for Hindi (challenge).

Our approach demonstrates that automated data cleaning can meaningfully improve low-resource MT, particularly for datasets with moderate error rates. However, important limitations remain: test set quality issues likely suppress reported scores, automated correction cannot perfectly replicate human judgment (as seen in Odia's mixed results), and our LoRA-only experiments likely underestimate the full potential when combined with full finetuning.

Future work should focus on three key directions: (1) human evaluation by native speakers to validate improvements beyond automatic metrics, (2) applying our pipeline to create higher-quality test (eval and challenge) sets for more accurate evaluation, and (3) combining corrected data with full model finetuning to validate whether quality improvements compound with increased capacity.

We hope that our publicly released dataset, code, and models provide a foundation for future research in automated quality assurance for multimodal datasets, potentially enabling more robust and equitable AI systems across diverse languages.

References

2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Seif Ahmed, Mohamed T Younes, Abdelrahman Moustafa, Abdelrahman Allam, and Hamza Moustafa. 2025. Msa at imageclef 2025 multimodal reasoning: Multilingual multimodal reasoning with ensemble vision language models. *arXiv preprint arXiv:2507.11114*.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, and 1 others. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Lisa Beinborn, Teresa Botschen, and Iryna Gurevych. 2018. Multimodal grounding for language processing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2325–2339, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Siddharth Betala and Ishan Chokshi. 2024. Brotherhood at WMT 2024: Leveraging LLM-generated contextual conversations for cross-lingual image captioning. In *Proceedings of the Ninth Conference on Machine Translation*, pages 852–861, Miami, Florida, USA. Association for Computational Linguistics.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924, Vancouver, Canada. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. Dspy: Compiling declarative language model calls into self-improving pipelines.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and 1 others. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Jacob Menick, Kevin Lu, Shengjia Zhao, E Wallace, H Ren, H Hu, N Stathas, and F Petroski Such. 2024. Gpt-4o mini: advancing cost-efficient intelligence. *Open AI: San Francisco, CA, USA*.

Aarathi Rajagopalan Nair, Deepa Gupta, and B Premjith. 2024. Investigating translation for indic languages with bloomz-3b through prompting and lora fine-tuning. *Scientific Reports*, 14(1):24202.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shantipriya Parida, Ondřej Bojar, Idris Abdulmumin, Shamsuddeen Hassan Muhammad, and Ibrahim Sa'id Ahmad. 2024. Findings of wmt2024 english-to-low resource multimodal translation task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 677–683.

Shantipriya Parida, Ondřej Bojar, and Satya Ranjan Dash. 2019. Hindi visual genome: A dataset for multimodal english to hindi machine translation. *Computación y Sistemas*, 23(4):1499–1505.

Arghyadeep Sen, Shantipriya Parida, Ketan Kotwal, Subhadarshi Panda, Ondřej Bojar, and Satya Ranjan Dash. 2022. Bengali visual genome: A multimodal dataset for machine translation and image captioning. In *Intelligent Data Engineering and Analytics: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)*, pages 63–70. Springer.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Lucia Specia, Stella Frank, Khalil Sima'An, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *First Conference on Machine Translation*, pages 543–553. Association for Computational Linguistics (ACL).

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Siu Ming Wong, Ho Leung, and Ka Yan Wong. 2024. Efficiency in language understanding and generation: An evaluation of four open-source large language models.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. 2025. Llava-critic: Learning to evaluate multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13618–13628.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

A Dataset and Task Description

We utilize the official datasets provided by the WAT 2025 organizers for the English-to-Indic Multimodal Translation Task. The datasets are derived from the Visual Genome corpus (Krishna et al., 2017) and consist of image-caption pairs across four target languages: Hindi (Parida et al., 2019), Bengali (Sen et al., 2022), Malayalam[5], and Odia[6]. Each example comprises an image, bounding box coordinates specifying a rectangular region of interest, an English caption describing that region, and corresponding translations in the target languages.

A.1 Task Definition

The task requires generating captions in the target language given three inputs: (1) an image, (2) a rectangular region within that image specified by bounding box coordinates, and (3) an English caption describing the visual content of that region. Participants may choose to utilize any combination of these inputs, leading to three possible translation modalities: text-only translation (using only the English caption), image-only captioning (using only the visual information), or multimodal translation (leveraging both text and image).

A.2 Dataset Statistics

Task	Source
English→Hindi	Hindi Visual Genome 1.1 (Parida et al., 2019)
English→Bengali	Bengali Visual Genome 1.0 (Sen et al., 2022)
English→Malayalam	Malayalam Visual Genome 1.0[7]
English→Odia	Odia Visual Genome 1.0[8]

Table 3: Tasks and their corresponding dataset sources.

The training set contains 28,928 examples per language, while three evaluation sets are provided for assessment: (1) a development set (Dev) with 998 examples for model validation, (2) an evaluation set (Eval) with 1,595 examples for primary assessment, and (3) a challenge set (Challenge) with 1,400 examples specifically designed to test ambiguous cases where visual context is crucial for disambiguation (Parida et al., 2024). Our official submissions were evaluated on both the Eval and Challenge sets.

Table 3 shows the data sources of datasets for each task. Table 4 shows the parallel corpus statistics across the various languages.

B Prompts

[5] https://lindat.mff.cuni.cz/repository/items/7ed34663-0bd4-4163-8ae9-89b2a8323269

[6] https://lindat.mff.cuni.cz/repository/items/58e6a33d-4f0f-413b-a3f3-c921e0489022

[7] https://ufal.mff.cuni.cz/malayalam-visual-genome

[8] https://ufal.mff.cuni.cz/odia-visual-genome

Set	Sentences	English	Hindi	Bengali	Malayalam	Odia
Train	28,930	143,164	145,448	113,978	107,126	141,652
Dev	998	4,922	4,978	3,936	3,619	4,912
Test	1,595	7,853	7,852	6,408	5,689	7,734
Challenge	1,400	8,186	8,639	6,657	6,044	8,100
Total	32,923	164,125	166,917	130,979	122,478	162,398

Table 4: Parallel corpus statistics (word counts) for each dataset split across different language pairs.

Judge Module: Caption Quality Evaluation Prompt

System Role: You are an expert multilingual translator evaluating Indian language captions.
Primary Task: Determine if the target language caption correctly represents what's shown in the image and accurately conveys the English caption meaning.

Focus on MAJOR issues — ignore minor stylistic differences:

Category 1: VISUAL CONTEXT NEEDED — Translation depends on visual information

- Ambiguous words with multiple meanings (e.g., "dish" = food vs. container)
- Gender-specific terms requiring visual verification
- Spatial/directional terms (left/right/above/beside)
- Physical attributes (color, size, material, quantity)
- Object types/categories visible in image

Category 2: POOR TRANSLATION — Incorrect, incomplete, or unnatural

- Mistranslation or wrong meaning (semantic error)
- Missing key information from English
- Severe grammatical errors making it hard to understand
- Completely unnatural phrasing (not just stylistic preference)
- Wrong script or excessive script mixing

IGNORE these minor issues — mark as CORRECT:

- Minor punctuation differences (|, ., etc.)
- Optional articles or particles (a/the/one equivalents)
- Stylistic word order variations (both correct)
- Minor postposition variations if meaning is clear

Special handling: Empty captions → mark "incorrect" with "visual_context_needed"

Required Outputs:

- `status`: "correct" or "incorrect"
- `reason`: "visual_context_needed", "poor_translation", or "none"
- `confidence`: Score 0-1
- `explanation`: Brief justification citing the specific issue (1-2 sentences)

Table 5: Judge module prompt template. The judge evaluates caption quality using visual context and classifies captions into three categories: correct, requiring visual context for disambiguation, or poor translation quality. Explicit instructions guide the model to focus on major issues while ignoring minor stylistic variations.

Corrector Module: Natural Caption Generation Prompt

System Role: Expert translator creating natural Indian language captions.
Generation Process:

1. **Analyze the IMAGE** to understand visual context
2. **Use visual details** to resolve ambiguities (e.g., ”dish” = food vs. container)
3. **Create natural captions** that native speakers would use
4. **Match English meaning** while respecting target language style

Target Language Length Guidelines (be concise, not verbose):

- **Hindi**: Similar word count to English
- **Bengali**: ∼20% fewer words than English
- **Malayalam**: ∼25% fewer words than English
- **Odia**: Similar word count to English

Important Note: Original caption may be wrong/missing — **trust the IMAGE first**

Required Outputs:

- `corrected_caption`: Natural, accurate caption in target language
- `explanation`: What you corrected and why (1-2 sentences)

Table 6: Corrector module prompt template. The corrector generates natural captions using visual evidence with language-specific length guidelines to ensure native-like output. The model is instructed to prioritize image information when the original caption may be incorrect or missing.

Author Index

Akhtar, Md Shad, 54
Azami, Haruto, 30

Betala, Siddharth, 124
Betala, Vipul, 124
Bojar, Ondřej, 103

Dabre, Raj, 78
Dhaka, Akash Kumar, 109
Dhal, Debasish, 109
Dias, H. Rukshan, 66

Edamatsu, Taishi, 16
Ekbal, Asif, 115
Endo, Rei, 98

Gain, Baban, 115
Goto, Isao, 1, 16

Halder, Deepon, 78

Ieda, Takashi, 1
Ito, Hitoshi, 93

Jayakumar, Thanmay, 78

Kajita, Futo, 30
Kasada, Kazuhiro, 1
Kawai, Yoshihiko, 93, 98
Kinugawa, Kazutaka, 93
Kumar, Deepak, 115

Mino, Hideya, 93, 98

Nagata, Masaaki, 1
Nakazawa, Toshiaki, 1
Ninomiya, Takashi, 16
Nishimura, Nobuyori, 30

Okuyama, Shoichi, 1

Parida, Shantipriya, 103, 109

R, Revathy V, 109
Raj, Kushan, 124
Rajput, Siddharth, 54

Saswade, Rohan, 124
Sekhar, Sambit, 109
Shirai, Naoto, 93
Shrivastava, Manish, 35
Singh, Kshetrimayum Boynao, 115
Sood, Sanmay, 54
Sudoh, Katsuhito, 1
Sumanathilaka, Deshan, 66

Tsunakawa, Takashi, 1

Utsuro, Takehito, 30

Yadav, Saumitra, 35

Zhang, Yin, 30