WAT 2026

The 13th Workshop on Asian Translation

Proceedings of the Workshop

November 9, 2026

©2026 Association for Computational Linguistics

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 979-8-89176-446-0

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. With the recent progress of AI technology, machine translation is now frequently treated as one of the benchmarks for large language models (LLMs). Nevertheless, machine translation technology is still far from perfect: the translation quality of Asian languages remains lower than that of European languages, and problems such as low-resource languages persist. 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, and WAT will continue to provide a venue for analyzing the current state of machine translation technology and discussing its further development.

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 – WAT2025), WAT2026 will bring together machine translation researchers and users to try, evaluate, share and discuss brand-new ideas about machine translation. For the 13th WAT, we have the 2nd Patent Claims Translation/Evaluation Task and the English→Indic Multimodal Translation Tasks. We have 4 teams who submitted their results.

In addition to the shared tasks, WAT2026 also features research papers on topics related to machine translation, especially for Asian languages. We received 9 research papers submitted including ARR commitment, and the program committee accepted 7 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-IJCNLP2026 organizers for their help with administrative matters.

WAT 2026 Organizers

Organizing Committee

Organizers

Toshiaki Nakazawa, National Institute of Informatics (NII), Japan
Isao Goto, Ehime University, 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, Inc., Japan
Sadao Kurohashi, National Institute of Informatics, Japan

Program Committee

Program Committee Members

Raj Dabre, Google Research Australia, Australia
Shohei Higashiyama, NICT
Hideya Mino, NHK
Takashi Ninomiya, Ehime University
Shantipriya Parida, AMD Silo AI
Katsuhito Sudoh, Nara Women's University
Masao Utiyama, NICT
Takehito Utsuro, University of Tsukuba
Thanmay Jayakumar, Indian Institute of Technology
Deepon Halder, Indian Institute of Technology
Masaaki Nagata, NTT, Inc.
Takashi Tsunakawa, Shizuoka University
Toshiaki Nakazawa, National Institute of Informatics
Isao Goto, Ehime University

Table of Contents

PSMT-eval: A Benchmark Dataset for Machine Translation Evaluation in Punjabi Shahmukhi
Abdul Basit and Abdul Hameed Azeemi 1

LIPI: Transliterating Hindi the Way People Type It
Abhishek Gautam 9

Style Diagnostic Profile for Translation: A Preliminary JA–EN Study
Yuri Amimoto, Ryosuke Mitani, Rem Hida, Ritvik Choudhary and Toshiyuki Sekiya 17

HexaBee-4B: Six-Language Translation with Multilingual Answering and Bidirectional Supervision
Iman Barati, Mohammadmostafa Rostamkhani and Mohammad Badzohreh 46

SerialDistill: Retrieval-Conditioned On-Policy Distillation for Data-Scarce Serial Subtitle Translation
Yiran Zhang and Ziqi Zhang 71

Towards a Rubric-based LLM-as-a-Judge for Patent Claim Translation Evaluation
Yingyi Fu, Takehito Utsuro and Masaaki Nagata 84

Machine-Translated Bengali in the Wild: A Provenance Benchmark and a Detector-Flagged Shift Toward Translation in Common Crawl
Mantaqa Abedin and Jeba Afia Khan 104

UTSK26 at WAT2026 Patent Claims Translation/Evaluation Tasks
Yingyi Fu, Kosei Buma and Takehito Utsuro 119

Ehime-U System for the WAT 2026 Patent Claims Evaluation Task: LLM-Based Translation Error Prediction Using Word Alignments
Wang Xu Yin, Isao Goto and Takashi Ninomiya 134

GABU@WAT 2026: Zero Shot and QLoRA Adaptation of Gemma 4 E4B for Multimodal English to Indic Translation
Shiv Ram Saud, Aashish Mahato, Prajwal Ghimire, Sundeep Dawadi, Ramesh Saud and Sunil Regmi 145

Findings of the 2nd Patent Claims Translation/Evaluation Task at WAT 2026
Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda and Masaaki Nagata 155

Program

Monday, November 9, 2026

09:00 - 09:05 *Welcome by Toshiaki Nakazawa*

09:05 - 10:20 *Research Paper I*

HexaBee-4B: Six-Language Translation with Multilingual Answering and Bidirectional Supervision
Iman Barati, Mohammadmostafa Rostamkhani and Mohammad Badzohreh

LIPI: Transliterating Hindi the Way People Type It
Abhishek Gautam

SerialDistill: Retrieval-Conditioned On-Policy Distillation for Data-Scarce Serial Subtitle Translation
Yiran Zhang and Ziqi Zhang

10:20 - 11:00 *Coffee Break*

11:00 - 12:40 *Research Paper II*

Style Diagnostic Profile for Translation: A Preliminary JA–EN Study
Yuri Amimoto, Ryosuke Mitani, Rem Hida, Ritvik Choudhary and Toshiyuki Sekiya

Machine-Translated Bengali in the Wild: A Provenance Benchmark and a Detector-Flagged Shift Toward Translation in Common Crawl
Mantaqa Abedin and Jeba Afia Khan

PSMT-eval: A Benchmark Dataset for Machine Translation Evaluation in Punjabi Shahmukhi
Abdul Basit and Abdul Hameed Azeemi

Towards a Rubric-based LLM-as-a-Judge for Patent Claim Translation Evaluation
Yingyi Fu, Takehito Utsuro and Masaaki Nagata

12:40 - 14:00 *Lunch Break*

14:00 - 14:50 *Invited Talk by Prof. Alham Fikri Aji*

Monday, November 9, 2026 (continued)

14:50 - 15:05 *Shared Task - English to Indic Multimodal Translation Task: Findings of the English to Indic Multimodal Translation Tasks (Shantipriya Parida)*

15:05 - 15:30 *Shared Task - English to Indic Multimodal Translation Task*

GABU@WAT 2026: Zero Shot and QLoRA Adaptation of Gemma 4 E4B for Multimodal English to Indic Translation
Shiv Ram Saud, Aashish Mahato, Prajwal Ghimire, Sundeep Dawadi, Ramesh Saud and Sunil Regmi

15:30 - 16:00 *Coffee Break*

16:00 - 17:15 *Shared Task - Patent Claims Translation/Evaluation Task*

Findings of the 2nd Patent Claims Translation/Evaluation Task at WAT 2026
Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda and Masaaki Nagata

Ehime-U System for the WAT 2026 Patent Claims Evaluation Task: LLM-Based Translation Error Prediction Using Word Alignments
Wang Xu Yin, Isao Goto and Takashi Ninomiya

UTSK26 at WAT2026 Patent Claims Translation/Evaluation Tasks
Yingyi Fu, Kosei Buma and Takehito Utsuro

17:15 - 17:20 *Closing by Toshiaki Nakazawa*

PSMT-eval: A Benchmark Dataset for Machine Translation Evaluation in Punjabi Shahmukhi

Abdul Basit and Abdul Hameed Azeemi
Lahore University of Management Sciences, Lahore, Pakistan
{a_basit, abdul.azeemi}@lums.edu.pk

Abstract

This paper introduces PSMT-eval, a new dataset for machine translation (MT) evaluation in Punjabi Shahmukhi, a low-resource language spoken by over 80 million people. We constructed an English–Punjabi Shahmukhi parallel corpus through pivot-based human translation, where the Urdu side of existing English–Urdu datasets was manually translated into Punjabi Shahmukhi. Four widely used English–Urdu datasets were selected: tatoeba-test.eng-urd, FLORES101, IN22, and MKB, with total count of 5182 sentences. To assess the dataset, we evaluated translations generated by state-of-the-art large language models (LLMs), including GPT-5, GPT-OSS-120B, Claude 4 Sonnet, LLaMA-3.3 70B Instruct, and Qwen-3 32B. Results show that only Claude 4 Sonnet achieved competitive performance, with a maximum **SacreBLEU score of 17.51** on IN22. GPT-5 followed with a maximum **SacreBLEU score of 10**, while other models performed poorly. Further analysis highlights major challenges in Punjabi Shahmukhi MT. To the best of our knowledge, this is the first benchmark dataset specifically for MT evaluation in Punjabi Shahmukhi. PSMT-eval is a valuable resource for advancing low-resource MT research and is publicly [1] available to support future work.

1 Introduction

There has been tremendous growth in natural language processing (NLP) in terms of tasks, models, and datasets for high-resource languages. However, progress in low-resource languages remains limited, even for fundamental tasks such as machine translation (MT). Among these low-resource languages is Punjabi Shahmukhi, spoken by over 80 million people[2], yet highly underrepresented in the NLP domain.

Punjabi is widely spoken in both Pakistan and India, but it is written in two distinct scripts: Shahmukhi,is a right-to-left script derived from the Perso-Arabic script and primarily used in Pakistan, and Gurmukhi, derived from Indic scripts and used mainly in India (Saini and Lehal, 2011). Existing benchmarks such as FLORES-101 (Goyal et al., 2022) provide valuable resources for MT evaluation in many low-resource languages, including Punjabi, but only in the Gurmukhi script. Extending these resources to Shahmukhi may appear straightforward, yet it presents unique challenges due to script differences, lack of tools, and limited prior resources. As a preliminary experiment, we translated the first 100 sentences from the Tatoeba Translation Challenge (tatoeba-test.eng-urd) dataset into Punjabi Shahmukhi using a pivot-based human translation approach and evaluated it on GPT-4. The system achieved a **SacreBLEU score of only 9**, indicating very poor translation quality. Table 1 illustrates representative cases, where sentences intended to be translated into Punjabi Shahmukhi were instead rendered in Urdu. This outcome highlights the difficulty of the task and motivates the development of a dedicated dataset specifically for machine translation evaluation in Punjabi Shahmukhi. Establishing standardized evaluation datasets accelerates progress in model development by providing consistent and challenging benchmarks. The proposed Punjabi Shahmukhi test set **PSMT-eval** serves as a foundational resource to facilitate systematic evaluation and guide future improvements in model performance.

2 Related Work

Parallel corpora are the foundation of modern machine translation, enabling the training

[1] https://huggingface.co/datasets/Niazi/PSMT_EVAL

[2] https://www.pbs.gov.pk/sites/default/files/population/2017/tables/pakistan/Table11n.pdf

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 1–8
November 9, 2026 ©2026 Association for Computational Linguistics

Source (English)	Reference Translation (Punjabi Shahmukhi)	GPT-4 Translation
How rude of you!.	اتوں بڑا بدتمیز ایں	کتنے بد تمیز ہو تم!
Science can be used for good or evil purposes.	سائنس نوں چنگے یا ماڑے مقصد لئی ورتیا جا سکدا اے۔	سائنس اچھے یا برے مقاصد کے لئے استعمال کیا جا سکتا ہے۔
The animal died from hunger	اوہ جانور بُھک نال مر جاوے گا۔	وہ جانور بھوک سے مر گیا۔
My best friend is a book.	کتاب میری چنگی ساتھی اے۔	کتاب میری بہترین دوست ہے
Urdu is her mother tongue	اردو اوہدی ماں بولی اے۔	اُردو اس کی مادری زبان ہے۔

Table 1: Comparison between reference translations and GPT-4 outputs in Punjabi Shahmukhi.

and evaluation of data-driven models across languages. A number of large-scale resources have been developed for high-resource languages, creating standardized benchmarks for fair system comparison. The Europarl corpus (Koehn, 2005) is one of the earliest and most widely used resources, containing proceedings of the European Parliament. It provides a standardized test set of approximately 40,000 sentence pairs (about one million words) from the last quarter of 2000, with subsets aligned across 11 European languages. Similarly, the United Nations corpus Ziemski et al. (2016) provides a large, multilingual parallel resource for the six official UN languages, comprising test sets of 4,000 unique, one-to-one aligned sentences across all languages. For low-resource languages, Tiedemann (2020) provides training and test sets for machine translation across more than 500 languages. For each language, each test set contains at least 200 sentences. For multilingual and low-resource MT, several benchmark resources have been introduced in recent years. The FLORES evaluation datasets (Guzmán et al., 2019) were specifically designed for low-resource languages such as Nepali and Sinhala. The test set comprises 2,924 sentences for Nepali–English and 2,905 sentences for Sinhala–English. This effort was later expanded into FLORES-101 (Goyal et al., 2022), which covers over 100 languages and provides a unified evaluation framework with the test set of 997 per language pair. However, for Punjabi, FLORES-101 only includes the Gurmukhi script, used primarily in India, while excluding Shahmukhi, the Perso-Arabic script variant widely used in Pakistan. For Indian languages, the IN22 test set was introduced by Gala et al. (2023). It covers 22 Indian languages, with each test set containing 1,503 sentence pairs. While it includes Punjabi, the data is available only in the Gurmukhi script.

In summary, although a wide range of benchmark datasets exist for high-resource and several low-resource languages, Punjabi Shahmukhi remains critically underrepresented. Existing resources either focus on Punjabi Gurmukhi or exclude Shahmukhi altogether. To the best of our knowledge, no standardized dataset has been developed for English–Punjabi Shahmukhi MT evaluation. Our work addresses this gap by introducing PSMT-eval, a parallel corpus designed specifically for evaluating MT systems on Punjabi Shahmukhi, thereby contributing a novel benchmark to the NLP community.

3 Methodology and Experiments

Since Punjabi Shahmukhi is written in the same Perso-Arabic script as Urdu, the two languages share orthography and some vocabulary. However, they remain distinct, with important cultural and linguistic differences (Saini and Lehal, 2011). To construct the dataset, we first identified suitable Urdu resources to serve as the source for translation into Punjabi Shahmukhi. We then engaged a linguistic expert proficient who is a native speaker in both Urdu and Punjabi Shahmukhi and published work in Punjabi to carry out high-quality human translations. Each dataset was translated using a segment-by-segment approach. To validate the translations, we randomly selected ten percent of the sentences from each dataset and asked an independent linguistic expert to verify their accuracy and faithfulness. If needed they provide the alternate translation. The linguist experts were compensated with market-competitive rates.

3.1 Data Creation

To create PSMT-eval, we selected four publicly available test datasets. First, we utilized the tatoeba-test.eng-urd test set (Tiedemann, 2020),

Dataset	Total Sentences	Avg. words per sentence	Avg. length per sentence	Avg. word ratio	Avg. char ratio
Flores101	997	25.89	121.99	0.82	1.04
MKB	1019	20.05	93.25	0.96	1.18
tatoeba-test	1663	6.94	31.84	0.91	0.99
IN22	1503	11.56	53.02	0.88	1.01

Table 2: Statistics of Punjabi Shahmukhi datasets used in this study, including the total number of sentences, average words per sentence, average sentence length in characters, average Word ratio (EN/PNB), and average char ratio (EN/PNB).

Model	**tatoeba-test.eng-urd**	**MKB**	**IN22**	**FLORES101**
GPT-5	7.69	6.64	10.00	8.89
GPT-OSS-120B	7.32	5.72	9.58	7.81
Claude 4 Sonnet	**14.49**	**11.40**	**17.51**	**14.93**
Qwen 3 32B	2.00	3.12	3.01	3.62
LLaMA-3.3 70B Inst.	7.80	5.21	6.54	7.14

Table 3: SacreBLEU scores of five large language models on PSMT-eval

Model	**tatoeba-test.eng-urd**	**MKB**	**IN22**	**FLORES101**
GPT-5	35.42	34.92	39.01	38.26
GPT-OSS-120B	33.88	33.37	38.03	37.7
Claude 4 Sonnet	**41.97**	**40.06**	**45.18**	**44.42**
Qwen 3 32B	19.19	21.30	21.42	24.46
LLaMA-3.3 70B Inst.	28.68	21.27	24.41	23.69

Table 4: Chrf++ scores of five large language models on PSMT-eval

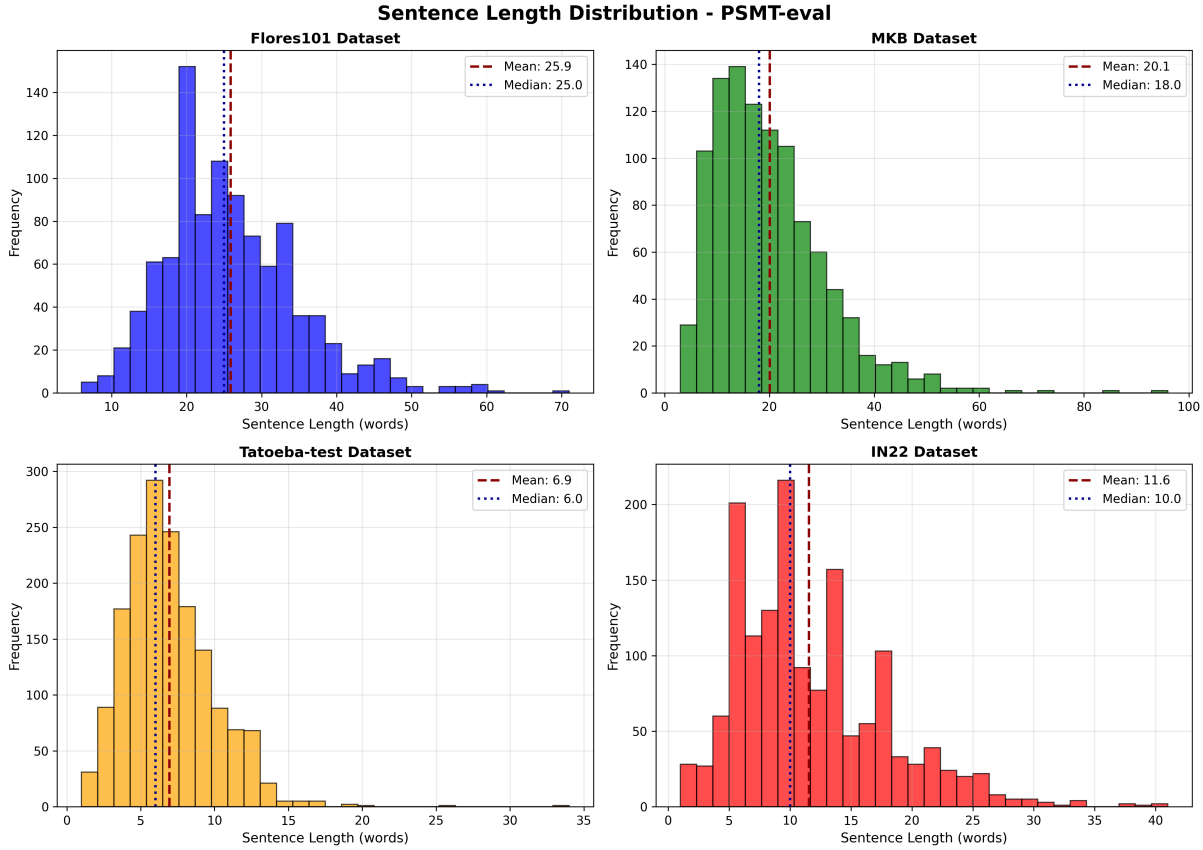


Figure 1: Sentence Length Distribution across Punjabi Language Datasets. This figure illustrates the distribution of sentence lengths (in words) across four Punjabi datasets: Flores101, MKB,tatoeba-test.eng-urd, and IN22. Each subplot shows the mean (red dashed line) and median (blue dotted line) sentence lengths, highlighting variation in dataset characteristics and text complexity.

which is part of the Tatoeba Translation Challenge and includes test sets for over 500 languages. Our study specifically focuses on the Urdu subset. It consists of simple, informal, short sentences. Second, we employed the FLORES-101 dataset (Goyal et al., 2022), a valuable benchmark for low-resource languages covering 101 languages, which is multidomain, from which we used the Urdu portion. Third, we incorporated the Mann Ki Baat test dataset (Siripragada et al., 2020), which contains Urdu translations of speeches delivered by the Indian Prime Minister in multiple Indian languages; its domain is political. Finally, we used the IN22 test set (Gala et al., 2023), which provides a parallel corpus for 22 Indian languages, selecting the Urdu subset for our work. It spans 13 different domains, including Culture, Economy, Education, Entertainment, Geography, Government, Health, Industry, Legal, News, Religion, Sports, and Tourism.

3.2 Inter-Annotator Agreement

Evaluation was conducted by two evaluators holding a Master's degree in Punjabi (Shahmukhi) and fluent in Urdu, who independently rated the Punjabi Shahmukhi translation of each sampled sentence against its Urdu source using a 5-point adequacy scale (1 = Unacceptable to 5 = Perfect).Inter-annotator agreement was measured using quadratic-weighted Cohen's Kappa(Cohen, 1960, 1968) on a 10% sample (518 sentences) drawn independently from each of the four datasets. Combined agreement was moderate (Landis and Koch, 1977) (k = 0.549, 65.1% raw agreement), with substantial agreement on Indic (k = 0.785) and MKB (K = 0.606), moderate agreement on Flores (k = 0.432), and slight agreement on OPUS (k = 0.141) despite 63.3% raw agreement there. The lower scores reflect a systematic one-point scoring offset between evaluators on Flores and a skewed score distribution on OPUS a known cause of deflated kappa despite high observed agreement(Feinstein and Cicchetti, 1990), rather

than random disagreement.

3.3 Data distribution

The figure 1 illustrates clear differences in sentence length distributions across the four Punjabi datasets. The Flores101 dataset exhibits a moderately right-skewed distribution, with most sentences ranging between 15–30 words and a close alignment between the mean (25.9) and median (25), reflecting a balanced structure typical of translation benchmarks. The MKB dataset shows a slightly longer tail, with a mean of 20.1 and a median of 18, indicating moderate variability and the presence of longer sentences. In contrast, the tatoeba-test.eng-urd dataset displays a sharp peak between 5–10 words, with a mean of 6.9 and a median of 6, suggesting shorter and simpler sentence constructions characteristic of conversational or instructional content. The IN22 dataset demonstrates a wider spread, with a mean of 11.6 and a median of 10, reflecting relatively short yet diverse sentence lengths and greater heterogeneity within the corpus.

Table 2 summarizes key statistics for these datasets. Flores101 contains 997 sentences with the highest average of 25.89 words and 121.99 characters per sentence, indicating more complex and information-dense text. tatoeba-test.eng-urd, despite being the largest corpus with 1,663 sentences, exhibits the shortest average sentence length 6.94 words and 31.84 characters, implying simpler or fragmented linguistic structures. MKB and IN22 fall between these extremes, with MKB comprising 1,019 sentences averaging 20.05 words and IN22 including 1,503 sentences with an average of 11.56 words per sentence, representing moderate complexity levels across these corpora.

3.4 Models

To evaluate the effectiveness of PSMT-eval, we selected a diverse set of state-of-the-art large language models (LLMs), encompassing both commercial black-box systems and publicly available open-weight models. Specifically, we assessed the performance of proprietary models GPT-5 (OpenAI) and Claude 4 Sonnet (Anthropic) alongside open-weight models, including GPT-OSS-120B, LLaMA-3.3 70B Instruct (Meta), and Qwen-3 32B (Alibaba). We employed large language models in this study due to the absence of any publicly available machine translation systems capable of translating from English to Punjabi Shahmukhi.

For GPT-5, we configured the reasoning effort parameter to minimal and set verbosity to low, while for the open-weight models we fixed the temperature at 0.1. These settings were chosen to minimize stochastic variation and ensure outputs suitable for machine translation tasks.

To standardize the evaluation, all models were prompted with the instruction: “Please translate the English sentence into Punjabi Shahmukhi script. Only return the translation.” To enhance task clarity, we supplemented this with the contextual cue ”You are an expert English-to-Punjabi translator. Translate the given English text to Punjabi using Shahmukhi script (Perso-Arabic-based script). Provide only the translation without any explanations, notes, or additional text. Maintain the original meaning and context while ensuring natural Punjabi expression.”

In open-source LLMs, GPT-OSS-120B[3] and LLaMA-3.3 70B[4] Instruct were accessed via the Fireworks API , while Qwen-3 32B[5] was deployed using the Groq API. This setup enabled efficient experimentation under consistent conditions, ensuring comparability across both proprietary and open-weight systems.

3.5 Metrics

We evaluate translation performance using the SacreBLEU metric with default settings (Post, 2018), which provides standardized tokenization and scoring for detokenized output, ensuring fair and reproducible comparisons across models. In addition, we report Chrf++ scores (Popović, 2017), a character n-gram–based metric that is particularly effective for languages with rich morphology and complex sentence structures.

3.6 Results

The evaluation results reveal a clear performance gap between proprietary and open-weight models. Among all systems, Claude 4 Sonnet consistently outperforms others, achieving the highest SacreBLEU3 and Chrf++4 scores across all datasets. GPT-5 ranks second overall, producing competitive results particularly on the IN22 and

[3] https://fireworks.ai/models/fireworks/gpt-oss-120b

[4] https://fireworks.ai/models/fireworks/llama-v3p3-70b-instruct

[5] https://console.groq.com/docs/model/qwen/qwen3-32b

Issue	Source	Reference Translation (Punjabi Shahmukhi)	Model Translation	Issue Detail
Language Error	The hostel's guests were mostly citizens of the United Arab Emirates.	ابوسٹل دے زیادہ تر پروہنے متحدہ عرب امارات دے شہری سن	ہاسٹل کے مہمان زیادہ تر متحدہ عرب امارات کے شہری تھے۔	All sentence is translated in Urdu Rather then Punjabi Shahmukhi
Mistranslation	A hostel collapsed in Mecca, the holy city of Islam at about 10 o'clock this **morning** local time.	اسلام دے مقدس شہر مکہ وچ اج **سویرے** مقامی ویلے دے مطابق سویرے 10 وجے ہاسٹل ڈِگ گیا۔	مکہ وچ اک ہوسٹل ڈگ گیا، اسلام دا مقدس شہر، سہ پہر تقریبن 10 بجے مقامی وقت۔	**سویرے** (Morning) is replaced by **سہ پہر** (after noon) in the translation.
Word-Omission	its second claw was larger, giving rise to the name Hesperonychus which means "western claw."	نسبتاً وڈا سی، جس دی وجہ توں ہیسپرونکس دے ناں دی مقبولیت ہوئی، جس دا مطلب اے مغربی پنجہ۔	اُس دی دوجی نکھ وڈی سی، جس کر کے اس دا نام ہیسپیرون	Missing the translation of *which means "western claw."* .
Different language	Pittman suggested that conditions wouldn't improve until sometime next week .	یہ پٹمین نیں اشارہ دتا کہ اگلے ہفتے تک حالات چنگے نئیں ہون گے۔	ਪਿਟਮੈਨ ਨੇ ਸੁਝਾਇਆ ਕਿ ਸਥਿਤੀਆਂ ਸੁਧਾਰ ਨਹੀਂ ਹੋਵੇਗਾ ਅਗਲੀ ਹਫ਼ਤੇ ਦੇ ਕੁਝ ਵੇਲੇ ਤੱਕ।	The whole sentence is translated in Gurmukhi rather than in Punjabi Shahmukhi
Transliteration	Following the **race**, Keselowski remains the Drivers' Championship leader with 2,250 points.	دوڑ دے بعد، 2,250 پوائنٹاں دے نال کیسیالوسکی ڈرائیوراں دی چیمپئن شپ چ برتری رہی۔	ریس کے بعد، کیزلاؤسکی 2,250 پوائنٹس کے ساتھ ڈرائیورز چیمپئن شپ کا لیڈربرقرار رہتا ہے۔	**ریس** used instead of **دوڑ** in translation.

Table 5: Examples of different Punjabi translation issues observed in Neural Machine Translation outputs.

Flores101 datasets though it still lags behind Claude.

In contrast, open-weight models demonstrate mixed and generally weaker performance. GPT-OSS-120B shows promise, achieving relatively strong Chrf++ scores (close to GPT-5 in some cases). However, LLaMA-3.3 70B Instruct and especially Qwen-3 32B perform poorly, with very low SacreBLEU and ChrF++ scores. Their weak results highlight the limitations of current open-weight models when applied to low-resource language translation without domain-specific adaptation.

The performance patterns remain consistent dataset-wise: Claude dominates across tatoeba-test. eng-urd MKB, IN22, and Flores101, followed by GPT-5, while open-weight models trail behind. The stronger ChrF++ scores for proprietary models indicate their robustness in capturing complex linguistic features. This performance indicates that there is a large room for improvement in the open weight models and also in the proprietary models for low-resource languages like Punjabi Shahmukhi.

4 Challenges

Our research has unveiled several challenges associated with Punjabi Shahmukhi translation. These challenges vary across models—some models exhibit a higher frequency of specific errors, while others demonstrate comparatively fewer issues. All issues fall under the domain of accuracy (Lommel et al., 2013). The challenges are summarized below:

- Machine translation between closely related languages tends to be less challenging and generally produces fewer errors compared to translation between distant languages (Popović et al., 2016). Claude Sonnet 4 demonstrates a recurring issue in which it translates the target sentence from English into Urdu instead of Punjabi Shahmukhi. This challenge persists across all evaluated models, likely due to the high degree of orthographic and linguistic similarity between Urdu and Punjabi Shahmukhi. This issue is illustrated in the first row of Table 5.

- When a translation fails to accurately convey the meaning of the source text, it is referred to as a mistranslation (Freitag et al., 2021). This issue was observed across all evaluated models. For instance, the LLaMA-3 70B Instruct model demonstrates this challenge, as illustrated in the second row of Table 5.

- Neural Machine Translation (NMT) systems often omit important words from the source text, which noticeably reduces the adequacy and completeness of the translation (Yang et al., 2019). This challenge is primarily observed in GPT-OSS-120B and, to a lesser extent, in other models. For example, in the third row of Table 5

- Qwen 3 (32B) and LLaMA 3.3 (70B Instruct) frequently produce translations in Punjabi Gurmukhi rather than Punjabi Shahmukhi, despite the contextual instruction clearly specifying that the output should be in the Perso-Arabic (Shahmukhi) script. In contrast, GPT-5 and GPT-OSS-120B exhibit

Error Type	Qwen 3 32B	GPT-OSS-120B	LLaMA-3.3 70B	GPT-5	Claude 4 Sonnet
Language Error	19.0	18.8	20.0	19.2	**33.3**
Mistranslation	14.3	12.5	**20.0**	15.4	16.7
Word Omission	14.3	**31.3**	13.3	15.4	11.1
Different Language	**38.1**	9.4	33.3	11.5	22.2
Transliteration	14.3	28.1	13.3	**38.5**	16.7

Table 6: Error type distribution (%) across models on all dataset

fewer occurrences of this issue. This error is categorized as **Major Severity** because it substantially compromises the clarity, reliability, and practical usability of the translated text for its intended purpose (Alemayehu et al., 2024). This issue is demonstrated in the fourth row of Table 5.

- Transliteration errors may occur due to unclear transliteration choices or inconsistent segmentation between the source and target text (Sennrich et al., 2015). This issue is observed across models, with an example provided of GPT-5 in the last row of Table 5.

- To quantify translation error patterns, a stratified random sample of 270 sentences was drawn proportionally across all four corpora — Flores101 (n=51), MKB (n=54), Tatoeba-test (n=86), and IN22 (n=79) — with each stratum's sample size reflecting its relative proportion of the total 5,182 sentences. Table 6 presents the resulting percentage distribution of each error type across all five models on all four datasets, providing a comprehensive cross-model and cross-corpus view of translation error patterns in Punjabi Shahmukhi.

4.1 Reasons

We observed that the primary challenges arise from models translating sentences either into Urdu or Gurmukhi rather than Punjabi Shahmukhi. Although Urdu shares significant orthographic similarity with Punjabi Shahmukhi, the two languages differ culturally and linguistically. We hypothesize that the limited availability of Punjabi Shahmukhi data contributes to this performance limitation across models.

5 Limitation and Conclusion

In this study, we introduced PSMT-eval, a benchmark dataset designed for evaluating machine translation performance on Punjabi Shahmukhi, and assessed several state-of-the-art large language models (LLMs) using it. Our evaluation revealed that existing LLMs exhibit notable limitations when handling low-resource languages, particularly in accurately distinguishing between closely related scripts such as Urdu, Gurmukhi, and Shahmukhi. These findings underscore the urgent need for more robust translation systems and fine-tuned models that can effectively handle the linguistic and cultural nuances of underrepresented languages.

One key limitation of this work lies in the restricted availability of high-quality Punjabi Shahmukhi data, which may have influenced the models' overall performance. Additionally, the evaluation primarily focused on general-domain text, leaving domain-specific translation (such as news, literature, or social media) an open area for further exploration.

For future research, we plan to expand the PSMT-eval dataset by incorporating diverse domains and larger parallel corpora for Punjabi Shahmukhi. Furthermore, this work can be extended to other low-resource languages of Pakistan, including Sindhi, Balochi, and Pashto, to establish a broader evaluation framework for multilingual translation in the region. Ultimately, we aim to explore the development of specialized large language models optimized for low-resource languages, ensuring inclusivity and equitable language technology advancement across diverse linguistic communities.

References

Hizkiel Mitiku Alemayehu, Hamada M Zahera, and Axel-Cyrille Ngonga Ngomo. 2024. Error

analysis of multilingual language models in machine translation: A case study of english-amharic translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19758–19768.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Alvan R Feinstein and Domenic V Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549.

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.

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

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

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *The Tenth Machine Translation Summit Proceedings of Conference*, pages 79–86. International Association for Machine Translation.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2013. Multidimensional quality metrics: a flexible system for assessing translation quality. In *Proceedings of Translating and the Computer 35*.

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

Maja Popović, Mihael Arcan, and Filip Klubička. 2016. Language related issues for machine translation between closely related south slavic languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 43–52.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Tejinder Singh Saini and Gurpreet Singh Lehal. 2011. Word disambiguation in shahmukhi to gurmukhi transliteration. In *Proceedings of the 9th Workshop on Asian Language Resources*, pages 79–87.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shashank Siripragada, Jerin Philip, Vinay P Namboodiri, and CV Jawahar. 2020. A multilingual parallel corpora collection effort for indian languages. *arXiv preprint arXiv:2007.07691*.

Jörg Tiedemann. 2020. The tatoeba translation challenge–realistic data sets for low resource and multilingual mt. *arXiv preprint arXiv:2010.06354*.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534.

A AI Usage Statement

AI-based tools were used to assist with drafting portions of the text during preparation of this resubmission. All experimental design, data analysis, interpretation of results, and final scientific content were reviewed and are the sole responsibility of the authors.

LIPI: Transliterating Hindi the Way People Type It

Abhishek Gautam

Abstract

Romanized Hindi has no standard spelling: the same word may be typed *nahi*, *nahin*, or *nai*, and no formal romanization rule or general-purpose transliteration system captures this typed convention well. Transliteration quality therefore depends as much on *which* spelling a user would actually type as on phonological plausibility. We present LIPI, a compact, encoder-only Devanagari-to-Roman transliteration model ($\sim$0.5M parameters) that represents each *akshara* compositionally from its ordered Unicode codepoints, expands it into six learned CTC emission slots, and encodes the frames with a two-layer bidirectional GRU under a connectionist-temporal-classification objective. The model is trained on a *usage-ranked* lexicon we build only from the training partitions of Dakshina, Xlit-Crowd, and Aksharantar, ranked by an attestation-gated social-usage prior so that a frequent Roman spelling is preferred only when it is also attested as a transliteration of the source word. LIPI is on par with the much larger multilingual INDICXLIT baseline on general word transliteration (Aksharantar) and far stronger on naturally typed queries (FIRE 2013: 73.2 vs. 34.8 exact match), roughly doubling both baselines' word match on Dakshina and IndoNLP sentences while remaining small enough for on-device use. The word-level gains are statistically significant and hold on words unseen in training, and an emission-slot ablation isolates the learned expansion's contribution. The model, lexicon, and full benchmark results are publicly released.[1]

1 Introduction

Hindi has 528 million first-language speakers in India, 43.6% of the population (Census of India, 2018), and more than 600 million speakers worldwide, behind only English and Mandarin Chinese (Eberhard et al., 2026); the census groups regional varieties such as Bhojpuri under "Hindi", whereas LIPI targets standard Hindi vocabulary as commonly typed. Although formally written in Devanagari, much of its digital text is typed in the Latin script, driven by QWERTY keyboards and everyday Hindi–English code-mixing: 93.2% of posts in a Hindi–English Facebook corpus were in Roman script versus 2.9% in Devanagari (Vyas et al., 2014), Hinglish use on Twitter grew steadily from 2014 to 2022 (Sengupta et al., 2024), and corpora from this setting reach billions of tokens (Nayak and Joshi, 2022; Srivastava and Singh, 2020). Unlike Devanagari, however, this Roman practice has no orthographic standard: the same word is spelled many ways, with *love* typed as *pyaar* or *pyar* and *very* as *bahut*, *bohot*, or *bhut*, as users lengthen or drop vowels, double consonants, and fold in English loans at will. Which spelling appears is governed less by phonology than by informal, community-driven typing conventions, so the target of a good transliteration is not a single canonical form but the form a particular user would actually type.

Existing transliteration tools fit this setting poorly. Formal romanization schemes such as IAST, ISO 15919, and ITRANS are built to represent Devanagari losslessly and reversibly for scholarly or archival purposes (Indic-transliteration contributors, 2024); their diacritic-marked, strictly one-to-one output is rarely what anyone types in a chat window. General multilingual transliteration models are trained toward a single canonical target and inherit the formal conventions of their source lexicons, so their predictions can be phonologically correct yet stylistically *off* for conversational Hindi. In both cases the mismatch is not accuracy in the abstract but agreement with real, typed usage, the requirement that motivates every design decision below.

[1] https://github.com/ABHISHEKgauti25/lipi

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 9–16
November 9, 2026 ©2026 Association for Computational Linguistics

We take a deliberately compact, single-language view of this problem. Our central object is LIPI, a $\sim$0.5M-parameter, akshara-aware model trained with connectionist temporal classification (CTC) (Graves et al., 2006) to map Devanagari words to their naturally typed Roman forms. LIPI rests on two ideas: a *codepoint-compositional* akshara representation, so that unseen conjuncts remain analyzable, and a set of *learned emission slots* that resolve the source-to-target length mismatch inherent to CTC on aksharas. Crucially, its training target is not a formal lexicon but a *usage-ranked* one: candidate spellings are ranked by a social-usage prior gated by paired attestation, so a frequent Roman token is preferred only when it is also attested as a transliteration of the source word. Beyond displaying Devanagari text to users who read Roman Hindi, a practical use is data preparation: since collecting data in both scripts is costly, LIPI can generate Romanized counterparts of Devanagari corpora, or a Romanized view of systems that output Devanagari, while the Devanagari text remains canonical.

Our contributions are: (i) **LIPI**, a compact akshara-aware CTC architecture combining codepoint-compositional cluster embeddings with learned per-akshara emission slots; (ii) a **usage-ranked lexicon recipe** whose attestation-gated scoring injects social-media frequency while limiting English-homograph amplification, built entirely from training-partition data; and (iii) an **open release** of the model, lexicon, and evaluation harness, including full artifact and reproducibility details. To substantiate these, we evaluate LIPI against established transliteration systems on six held-out word- and sentence-level benchmarks, with paired-bootstrap significance tests and a train–test overlap and out-of-vocabulary analysis so that every result is read in the context of how much vocabulary the model has already seen.

2 Related Work

Our work sits at the intersection of code-mixed Hindi resources, Indic transliteration datasets and models, and monotonic neural transduction. We review each in turn.

Romanized and code-mixed Hindi. The dominance of English keyboards has made Roman Hindi a primary medium for online communication, and several corpora document its scale and character. L3Cube-HingCorpus collects 52.93M code-mixed sentences and 1.04B tokens from Twitter and uses them to pretrain HingBERT-family language models, showing that in-domain data materially helps on code-mixed tasks (Nayak and Joshi, 2022). PHINC contributes a parallel Hinglish–English corpus for machine translation and, in building it, documents how noisy and non-standardized user-generated romanization is compared with clean monolingual text (Srivastava and Singh, 2020). We draw on such data only as an aggregate usage signal: no model is trained on raw posts, and Roman unigram counts influence only which *attested* spelling becomes a training target.

Transliteration datasets and models. Dakshina releases native-script text, multi-reference romanization lexicons, and romanized parallel sentences for twelve South Asian languages, and is a standard benchmark for transliteration in both directions (Roark et al., 2020). Xlit-Crowd studies how to collect transliteration pairs cheaply, showing that non-expert crowd workers under explicit, non-redundant quality control produce reliable supervision (Khapra et al., 2014). Aksharantar scales supervised Indic transliteration to 26M pairs across a wide range of languages and trains INDICXLIT, an autoregressive multilingual Transformer and a widely used open baseline for the task (Madhani et al., 2023). Complementing these, rule-based packages that implement formal schemes (ITRANS, IAST, ISO 15919) give deterministic, reversible mappings but target scholarly rather than colloquial spellings (Indic-transliteration contributors, 2024).

Benchmarks closer to real use. Two resources probe transliteration under realistic, informal input. The FIRE 2013 transliterated-search track provides Roman queries typed by users for cross-script retrieval, which are far closer to natural typing than curated lexicons (Roy et al., 2013). The IndoNLP 2025 shared task targets real-time reverse (Roman-to-native) transliteration for romanized Indo-Aryan languages (Sumanathilaka et al., 2025); we repurpose its Hindi pairs and evaluate in the forward (Devanagari-to-Roman) direction.

Monotonic neural transduction. Devanagari-to-Roman transliteration is approximately monotonic, which makes CTC a natural objective: it marginalizes over monotonic alignments without frame-level labels (Graves et al., 2006), and a compact GRU encoder is sufficient for these short sequences (Cho et al., 2014). The characteristic

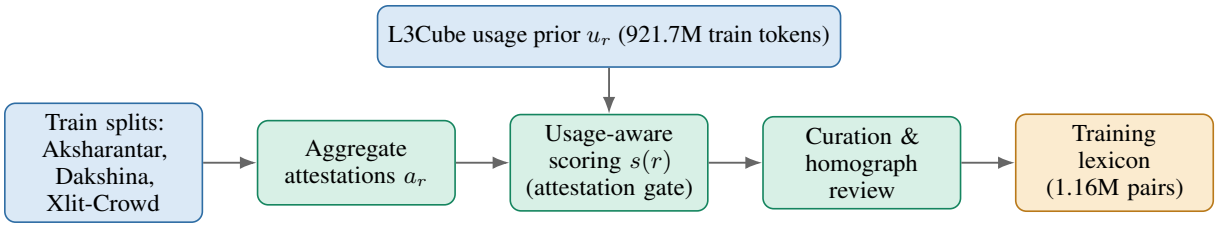


Figure 1: Data-preparation pipeline. Paired attestations a_r and the L3Cube usage prior u_r are combined by the gated score (1) and curated into the training lexicon. All inputs are training partitions.

difficulty is length expansion (one akshara can emit several Roman characters), which LIPI handles with shared, learned emission-slot embeddings rather than by duplicating source vectors. Context-sensitive phenomena such as schwa deletion, much of which orthography-only statistical models already recover (Arora et al., 2020), motivate a bidirectional encoder over a purely local lookup rule.

Because Indic pipelines reuse overlapping public lexicons, we build ours only from training partitions and report each test set's train overlap (Section 6.2).

3 Data Preparation

Because LIPI's target distribution *is* the training lexicon, constructing that lexicon to reflect real usage is part of the method rather than a preprocessing detail. The pipeline (Figure 1) produces the (native, Roman) training pairs.

3.1 Candidate evidence (training partitions)

The lexicon draws on two kinds of paired evidence and one usage prior, all taken from *training* partitions so that every evaluation set stays held out:

- **Aksharantar** Hindi train (native→Roman, one attestation per pair; ∼1.15M native types).
- **Dakshina** Hindi train lexicon (native, Roman, and a human *count* used as attestation weight; 25,000 words).
- **Xlit-Crowd** (Roman↔native, one attestation each; ∼14.9k pairs).

Lexicon keys are normalized to match the runtime tokenizer (Section 4): NFD for native keys, NFKC-casefold for Roman candidates. For each Devanagari word the pipeline aggregates an attestation count a_r over candidate Roman forms r. The usage prior u_r is the frequency of r in the *train* split of L3Cube-HingCorpus (921.7M tokens, validation excluded), computed on lowercased text and retained for the 135,307 candidate spellings that occur. It captures how likely a spelling is to be *typed*, not whether a Roman token expresses the target word; requiring paired attestation before u_r is consulted constrains that ambiguity.

3.2 Usage-aware scoring with an attestation gate

Each candidate is ranked by

$$s(r) = \log(1+a_r) + \lambda \frac{a_r}{a_r + K} \log(1+u_r), \quad (1)$$

with released values $\lambda = 0.3$ and $K = 0$. The factor $a_r/(a_r + K)$ is an *attestation gate*: with $K > 0$ it shrinks the usage boost for thinly-attested candidates, the mechanism that prevents a very frequent English word from outscoring the correct spelling. At $K = 0$ every attested candidate has $a_r > 0$, so the gate equals one and (1) reduces to $s(r) = \log(1+a_r) + 0.3 \log(1+u_r)$. Paired-data attestation therefore dominates, while social usage breaks or reverses close choices. The top-scoring candidate is selected, after which a human override list wins unconditionally.

3.3 Cleaning, curation, and homograph correction

We apply 39,562 train-supported human overrides, keep 632 clean out-of-vocabulary manual insertions, and *reject* 121 malformed (reversed or mixed-script) entries. A small, evaluation-independent override list resolves the residual English-homograph cases that scoring alone cannot: for example, favoring the first-person forms *maine* and *mein* over the English homographs *i* and *me*, and restoring *amavas*, *four*, and *dunga*. The result is the **1,164,037**-entry lexicon on which LIPI is trained.

3.4 Validating the attestation gate

To confirm the gate behaves as intended, we re-score the entire candidate pool for $K \in$

$\{0, 3, 5, 10, 20, 50\}$ without retraining and classify the picks that change. At the released $K = 0$, 131 selected spellings are English tokens; inspection sorts them into 67 correct loanwords that should be kept (e.g. *it*, *is*, *do*), 26 *amplification cases* where a better-attested correct spelling is outscored by a frequent English homograph, and 38 *residual* cases where the sources contain no correct Roman candidate at all. Raising K shows that $K \approx 10$ removes all 26 amplification cases while leaving the 67 loanwords untouched, confirming the mechanism. A *global* K is nonetheless a poor default: it flips roughly one well-attested, likely-correct entry as collateral for each case it fixes, and the 38 residual homographs have no correct candidate in the sources, so only a manual override resolves them regardless of K. The released lexicon therefore keeps $K = 0$ and handles the residual cases with the small override list of Section 3.3.

4 LIPI Architecture

The lexicon fixes *what* the model should emit; the architecture must emit it under CTC while keeping unseen conjuncts analyzable and the parameter budget small. Let a Devanagari word be segmented into A orthographic clusters $a_1, \ldots, a_A$ with Roman target $y_1, \ldots, y_M$. LIPI is an encoder-only model that predicts a CTC distribution over target characters; its forward pipeline and tensor shapes are shown in Figure 2.

4.1 Normalization and akshara segmentation

The tokenizer applies Unicode NFD normalization (Unicode Consortium, 2025), removes ZWJ/ZWNJ, strips whitespace, and segments each word into akshara-like clusters. Combining marks are absorbed into the current cluster, and a virama absorbs the following base character so that conjuncts remain intact. The input is bracketed with beginning- and end-of-word units. Each cluster is represented by at most $L = 16$ ordered Unicode codepoints.

4.2 Codepoint-compositional cluster embeddings

A whole-akshara vocabulary would make unseen conjuncts unknown. Instead, a cluster is composed from the embeddings of its constituent codepoints. Given codepoint embeddings $E[x_{i,\ell}] \in \mathbb{R}^{d_c}$ with $d_c = 32$, the ordered embeddings are concatenated and projected:

$$e_i = E[x_{i,1}] \,\|\, \cdots \,\|\, E[x_{i,L}], \tag{2}$$

$$c_i = \mathrm{LN}\big(\mathrm{Dropout}(\mathrm{GELU}(W_c e_i + b_c))\big), \tag{3}$$

where $c_i \in \mathbb{R}^{96}$. Concatenation makes codepoint position implicit in the projection weights while retaining compositionality.

4.3 Learned emission slots

Plain akshara-level CTC often has too few time steps. If a target of length M contains R adjacent repeated-character pairs, any valid CTC path requires at least

$$T_{\min}(y) = M + R \tag{4}$$

frames, because repeated labels must be separated by a blank. LIPI gives each akshara $S = 6$ learned emission opportunities. For slot embeddings $P_s \in \mathbb{R}^{96}$,

$$f_{i,s} = \mathrm{LN}(c_i + P_s), \qquad s = 1, \ldots, S. \tag{5}$$

The $A \times S$ frames are concatenated in source order. Unlike fixed upsampling, the slots are distinct and can acquire consistent within-akshara roles at a cost of only $S \times 96 = 576$ parameters.

4.4 Bidirectional encoder, CTC head, and budget

A two-layer bidirectional GRU with hidden size 128 per direction encodes the expanded frames. Layer normalization (Ba et al., 2016), dropout 0.1, and a linear projection produce logits over a 42-symbol target vocabulary that includes the CTC blank. Training minimizes

$$\mathcal{L}_{\mathrm{CTC}} = -\sum_{b} \log \sum_{\pi \in \mathcal{B}^{-1}(y^{(b)})} \prod_{t=1}^{T_b} p\Big(\pi_t \mid x^{(b)}\Big), \tag{6}$$

where $\mathcal{B}$ collapses repeated symbols and blanks; deployment uses greedy collapse and prefix beam search is optional. The released configuration (32-d codepoint embedding, 96-d model dimension, $S = 6$, two-layer BiGRU with 128 units per direction, dropout 0.1, source/target vocabularies 87/42) has exactly **534,314** trainable parameters, including a 10,794-parameter CTC output layer, with the recurrent encoder alone accounting for 88.0% of the total.

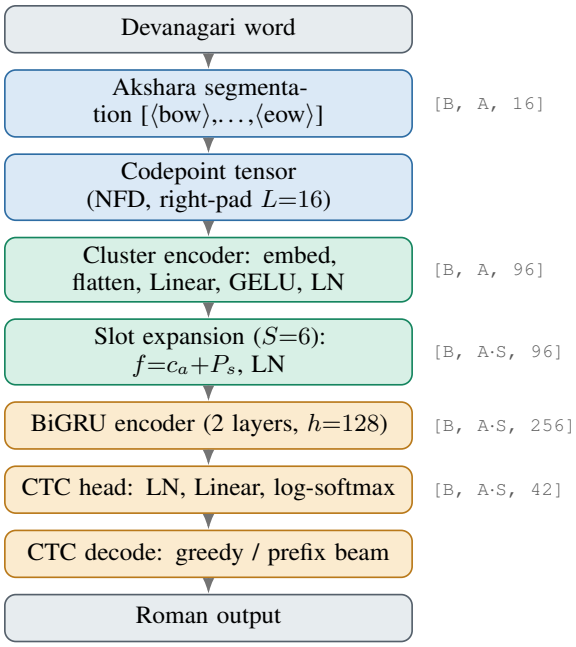


Figure 2: LIPI forward pipeline and tensor shapes. Colors group the stages: input/output (grey), preprocessing (blue), akshara composition (green), sequence model and CTC decoding (amber).

5 Experimental Setup

With the target distribution and architecture fixed, we specify the training recipe, the systems compared, and the held-out benchmarks.

Training data and recipe. The lexicon is converted into (native, Roman) word pairs, normalized (NFD source / NFKC-casefold target), filtered to remove CTC-infeasible pairs (keeping pairs that need at most 6 emission slots per akshara), and split *deterministically* into approximately 98% train, 1% development, and 1% internal test by $\mathrm{md5}(\text{native}) \bmod 100$. This yields 1,140,840 / 11,570 / 11,623 pairs (4 filtered). Training uses batch size 256, Adam (Kingma and Ba, 2015) with initial learning rate 10^{-3}, a cosine schedule, gradient clipping at 5.0, and the CTC loss (blank index 0, `zero_infinity`) for 25 epochs on GPU. The best development exact match is 74.16% (development CER 4.56%). The run used seed 20260814, PyTorch 2.6.0+cu124, and a single NVIDIA GTX 1650; full artifact details (checkpoint hash, framework versions, and the deterministic split) are included in the public release. This internal random split measures fitting to the lexicon, not lexical generalization; external benchmarks therefore carry the evidential weight.

Systems. We evaluate three model-only systems on one harness: **LIPI**, the proposed standalone neural model; **INDICXLIT** (Madhani et al., 2023), the public multilingual baseline and our sequence-to-sequence Transformer comparison (a character-level encoder–decoder with 6+6 layers and ∼11M parameters, trained on 26M Aksharantar pairs across 21 languages), decoded with beam width 5; and **ITRANS**, deterministic rule-based transliteration through the `indic-transliteration` package (Indic-transliteration contributors, 2024). INDICXLIT and ITRANS are external references whose predictions depend on neither our lexicon nor our model.

Evaluation sets. We evaluate on six held-out Hindi subsets: Aksharantar test (10,112 words), FIRE 2013 (2,420 typed tokens), Dakshina words test (2,500 multi-reference words), Dakshina sentences test (5,000), and IndoNLP Sets 1 and 2 (9,984 and 4,991 sentences, evaluated in the forward direction). Aksharantar is the largest direct word benchmark; FIRE is closer to informal typed transliteration; Dakshina is direct and multi-reference at the word level and also provides parallel sentences.

Metrics. Our primary metric, reported as *Score* in all tables, is a strict accuracy whose unit depends on the benchmark. For single-reference word sets (Aksharantar, FIRE) it is *exact match*: the percentage of words whose prediction equals the reference after lowercasing. For the multi-reference Dakshina word set it is *in-attested*: the percentage of words whose prediction is any of the human-attested romanizations. For sentence sets (Dakshina sentences, IndoNLP) it is *word match*: the percentage of reference tokens transliterated exactly, on Latin-lowercased text. As graded measures we also report corpus-level CER (edit distance summed over the corpus, divided by total reference length, using the nearest reference for multi-reference data) and the NEWS character F-score (Li et al., 2009). Score is intentionally strict: it treats plausible alternatives such as *kyun* and *kyu* as different unless both are references. To test significance we report paired-bootstrap 95% confidence intervals and two-sided p-values (B=10,000 resamples) for the LIPI−INDICXLIT difference on each word set.

Dataset	System	Score	CER ↓	NEWS-F
Aksharantar	LIPI	**40.93**	**14.17**	**90.00**
	INDICXLIT	39.67	14.55	89.71
	ITRANS	11.82	28.78	80.21
FIRE 2013	LIPI	**73.18**	**8.16**	**94.71**
	INDICXLIT	34.83	21.17	85.08
	ITRANS	43.84	20.32	88.64
Dakshina words	LIPI	**78.00**	**4.46**	**96.95**
	INDICXLIT	73.04	5.80	96.14
	ITRANS	26.68	22.41	85.95
Dakshina sent.	LIPI	**66.37**	**7.98**	**92.78**
	INDICXLIT	32.97	16.85	88.10
	ITRANS	30.07	16.73	87.56
IndoNLP Set 1	LIPI	**66.86**	**7.90**	**93.08**
	INDICXLIT	32.96	16.88	88.27
	ITRANS	31.68	16.75	87.86
IndoNLP Set 2	LIPI	**55.16**	**14.49**	**90.03**
	INDICXLIT	23.65	24.44	85.03
	ITRANS	26.67	23.57	84.88

Table 1: Test-set results (%). Score is exact match (Aksharantar, FIRE), in-attested (Dakshina words), or token word match (sentence sets); see Section 5. Score / NEWS-F: higher is better; CER: lower is better.

6 Results

Table 1 reports all three systems on the six benchmarks; the remaining tables add analyses the main table does not show: paired-bootstrap significance (Table 2), the train–test overlap profile (Table 3), out-of-vocabulary slices (Table 4), and the emission-slot ablation (Table 5). The text calls out only the conclusions these support.

6.1 Word- and sentence-level accuracy

Two patterns carry the result. On general word transliteration (Aksharantar) LIPI is level with INDICXLIT, edging it by $+1.27$ exact points at 95% CI $[+0.55, +1.99]$ (p=0.001); a statistically real but small gap, which is the intended outcome for a Hindi-only $\sim$0.5M-parameter model set against a much larger multilingual one. On naturally typed and sentence-level Hindi the gap opens sharply: LIPI reaches 73.18 exact on FIRE 2013 against 34.83/43.84 for the baselines, and roughly doubles both baselines' word match on every sentence set while about halving CER. Because sentence tasks require every token to be spelled the way a user would, the usage-aligned target pays off more there than on isolated words. Paired-bootstrap tests place the word-level LIPI−INDICXLIT gap above zero on every set (Table 2).

Set	Δ	95% CI	p
Aksharantar test	$+1.27$	$[+0.55, +1.99]$	0.001
FIRE 2013	$+38.35$	$[+36.03, +40.66]$	$< 10^{-4}$
Dakshina words test	$+4.96$	$[+3.48, +6.44]$	$< 10^{-4}$
FIRE 2013 (OOV-only)	$+15.56$	$[+8.95, +21.79]$	$< 10^{-4}$

Table 2: Paired-bootstrap comparison of LIPI and INDICXLIT on the exact/in-attested word metric (B=10,000 resamples). Positive Δ favors LIPI.

Benchmark split	In LIPI-train
Dakshina words test	8.64
Aksharantar test	15.64
IndoNLP Set 1	55.64
Dakshina sentences test	58.97
IndoNLP Set 2	60.00
FIRE 2013	82.37

Table 3: Fraction of each test set's native word-types present in LIPI's training data (percentages), read alongside Table 1.

6.2 Training overlap and out-of-vocabulary generalization

Benchmarks differ in how much of their vocabulary the model has already seen, so equal accuracy can mean different things across sets (Table 3). Aksharantar (15.6% seen) and Dakshina words (8.6%) probe largely unseen vocabulary, whereas FIRE (82.4%) and the sentence sets reuse many seen types. Measured against INDICXLIT's own training data, the held-out Aksharantar and Dakshina word splits are 0.0% seen, so both systems are tested on unseen words there. The high FIRE overlap raises whether LIPI's FIRE lead is generalization or lookup, so we restrict every system to the out-of-vocabulary slice, native word-types absent from LIPI-train (Table 4). LIPI still leads on every OOV slice, including FIRE by 15.6 exact points (significant, Table 2); INDICXLIT in fact scores *lower* on the seen, colloquial FIRE words than on the OOV ones. The typed-Hindi advantage is therefore genuine generalization to informal Hindi, not merely spelling lookup.

6.3 Emission-slot ablation

Retraining LIPI under matched data, recipe, and seed isolates the emission mechanism (Table 5). Expansion is necessary: a single slot leaves only 20.99% of training pairs CTC-feasible, reaching 100% by S=3. A larger budget helps, learned S=6 beating learned S=3 on every main set and giving the best development exact match (74.16 vs. 73.02). Learned versus fixed slots is roughly a tie on over-

Set	Slice	n	LIPI	INDICXLIT	ITRANS
FIRE 2013	all	2,420	**73.18**	34.83	43.84
	OOV	257	**63.42**	47.86	31.52
Aksharantar test	OOV	8,947	**38.35**	37.47	11.42
Dakshina words test	OOV	2,284	**77.58**	73.34	26.18

Table 4: Out-of-vocabulary generalization. OOV = native word-type absent from LIPI-train; exact match for Aksharantar/FIRE, in-attested for Dakshina. Values are percentages.

Variant	dev	Aksh.	FIRE	Dak.	FIRE-h	Dak-h
Learned S=6	**74.16**	**40.93**	73.18	**78.00**	44.98	**77.10**
Fixed rep. S=6	73.72	40.83	**73.47**	77.12	41.87	75.83
Learned S=3	73.02	40.00	70.99	76.48	**48.80**	75.60

Table 5: Emission-slot ablation (exact / in-attested; "hard" = targets with adjacent repeated characters). Retrained under matched data, recipe, and seed; dev is the internal development split. Values are percentages.

all accuracy, since the BiGRU recovers most of the context, but the learned slots give a consistent gain on the hard slice of targets with adjacent repeated characters, exactly the case they are meant to address (FIRE-hard $+3.1$, Dakshina-hard $+1.3$; the 418-token FIRE-hard slice is noisy). Greedy decoding stays within 0.25 points of beam-5. For deployment, the fp32 checkpoint is 2.1 MB and LIPI transliterates at $\sim$5.3 ms per word ($\sim$2,480 words/s batched) on a single CPU core.

7 Analysis and Discussion

The results support three conclusions, and we note where the evidence is direct or indirect. Compactness need not cost much accuracy on this monotonic, single-language task; domain fit dominates on FIRE and on sentence text, where the usage-aligned target is most valuable and the general-word ordering does not predict the typed-Hindi result; and the overlap profile shows that the low-overlap gains (Aksharantar, Dakshina words) are genuine generalization while the very high FIRE score partly reflects convention transfer on seen types, both useful in deployment but answering different questions. Evidence for the usage-ranked lexicon (contribution ii) is indirect: the gate validation (Section 3.4) shows that the usage prior changes targets without amplifying English homographs, and the largest gains appear on typed and sentence-level text, where target spelling matters most; retraining with λ=0 would isolate its effect directly. Mechanistically, codepoint composition keeps unseen conjuncts analyzable, the learned slots give CTC enough time steps to emit multi-character romanizations of one akshara for only 576 parameters, the usage-gated target supplies the spellings people actually type, and the bidirectional GRU supplies the local context needed for schwa deletion (Arora et al., 2020).

8 Conclusion

LIPI is a compact ($\sim$0.5M-parameter) akshara-aware CTC model for romanizing conversational Hindi. Its two design choices, codepoint-compositional aksharas and learned emission slots, keep unseen conjuncts analyzable and give CTC the time steps it needs to expand one akshara into several Roman characters, while a usage-ranked lexicon built only from training-partition data aligns the model's target with the spellings people actually type. Across six held-out benchmarks LIPI matches a much larger multilingual baseline on general word transliteration and is substantially stronger on naturally typed and sentence-level Hindi, with the advantage holding on out-of-vocabulary words. We release the model, lexicon, and evaluation harness.

Limitations

LIPI has clear boundaries that also mark its natural extensions. Our evaluation leans on exact match, which conflates model quality with reference style because Roman Hindi is many-to-many; Dakshina's in-attested metric and the graded CER and NEWS-F scores only partly address this, so a multi-reference human naturalness study on contemporary messaging and search text remains the right next measurement. Because transliteration is applied token by token, the model cannot resolve homographs, names, or borrowings that require sentential context. In scope, the released model is Hindi Devanagari-to-Roman only: it performs neither translation, grammatical rewriting, nor formal ISO/IAST/ITRANS romanization.

Two questions our ablations leave open would sharpen the architectural claims: a codepoint-versus-whole-akshara comparison with a novel-conjunct slice, to test the compositional representation directly, and a Transformer sequence-to-sequence model trained on our lexicon. Because INDICXLIT differs from LIPI in both architecture and training data, our comparison shows that a compact CTC model can match or exceed a $\sim$20$\times$

larger Transformer, but does not isolate the contribution of CTC itself. The most substantial direction is to carry the same usage-aligned strategy beyond monolingual forward transliteration. Building the multilingual counterpart of this recipe, and applying it to reverse (Roman-to-Devanagari) transliteration, would show whether usage alignment generalizes as broadly as the multilingual systems LIPI is measured against, and would turn a single-language result into a general recipe for typed-text transliteration.

References

Aryaman Arora, Luke Gessler, and Nathan Schneider. 2020. Supervised grapheme-to-phoneme conversion of orthographic schwas in Hindi and Punjabi. In *Proc. ACL*, pages 7791–7795.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv:1607.06450*.

Census of India. 2018. *Census of India 2011, Paper 1 of 2018: Language—India, States and Union Territories* (Table C-16). Office of the Registrar General, India, New Delhi.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. EMNLP*, pages 1724–1734.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2026. *Ethnologue: Languages of the World*. Online edition. SIL International, Dallas, Texas. https://www.ethnologue.com/insights/ethnologue200/.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. 23rd Int. Conf. Machine Learning*, pages 369–376.

Indic-transliteration contributors. 2024. indic_transliteration: Transliteration of Indic scripts. Software package. https://github.com/indic-transliteration/indic_transliteration_py.

Mitesh M. Khapra, Ananthakrishnan Ramanathan, Anoop Kunchukuttan, Karthik Visweswariah, and Pushpak Bhattacharyya. 2014. When transliteration met crowdsourcing: An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control. In *Proc. LREC*, pages 196–202.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Haizhou Li, A. Kumaran, Min Zhang, and Vladimir Pervouchine. 2009. Whitepaper of NEWS 2009 machine transliteration shared task. In *Proc. Named Entities Workshop*, pages 19–26.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul NC, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2023. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of EMNLP*, pages 40–57.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code-mixed Hindi–English dataset and BERT language models. In *Proc. WILDRE-6*, pages 7–12.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Işın Demirşahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: The Dakshina dataset. In *Proc. LREC*, pages 2413–2423.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the FIRE 2013 track on transliterated search. In *Proc. Forum for Information Retrieval Evaluation*, pages 4:1–4:7.

Ayan Sengupta, Soham Das, Md. Shad Akhtar, and Tanmoy Chakraborty. 2024. Social, economic, and demographic factors drive the emergence of Hinglish code-mixing on social media. *Humanities and Social Sciences Communications*, 11. https://doi.org/10.1057/s41599-024-03058-6.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proc. W-NUT*, pages 41–49.

Deshan Sumanathilaka, Isuri Anuradha, Ruvan Weerasinghe, Nicholas Micallef, and Julian Hough. 2025. IndoNLP 2025: Shared task on real-time reverse transliteration for romanized Indo-Aryan languages. *arXiv:2501.05816*.

The Unicode Consortium. 2025. Unicode Standard Annex #15: Unicode normalization forms. UAX #15.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. POS tagging of English-Hindi code-mixed social media content. In *Proc. EMNLP*, pages 974–979.

Style Diagnostic Profile for Translation: A Preliminary JA–EN Study

Yuri Amimoto, Ryosuke Mitani, Rem Hida, Ritvik Choudhary, Toshiyuki Sekiya
Sony Group Corporation, Tokyo, Japan
{yuri.amimoto, ryosuke.mitani, remu.hida,
ritvik.choudhary, toshiyuki.sekiya}@sony.com

Abstract

In Japanese–English machine-translation,especially for creative and localized content, standard MT metrics typically summarize translation quality as scalar scores, but provide little diagnostic information about how style changes in translation. This is limiting because politeness, directness, interpersonal stance, and character voice are encoded differently across Japanese and English, and stylistic change may reflect localization-oriented adaptation rather than error. We propose the Style Diagnostic Profile (SDP) framework: SDP assigns each utterance a seven-dimensional bipolar style vector, and Δ_{SDP}, the difference between target and source SDP vectors, represents translation-induced stylistic change as a multi-axis shift vector. To evaluate the framework, we first validate an LLM-based SDP scorer against bilingual human annotations on professional JA→EN subtitles and show that it approximates human style judgments. Using Style25, a controlled JA↔EN style-perturbation benchmark, we demonstrate that different perturbation categories occupy distinct regions in Δ_{SDP} space. As a case study, we analyze professional subtitles and observe stylistic adaptation rather than simple style copying. These results position the SDP framework as a diagnostic layer complementary to adequacy-oriented MT evaluation.

1 Introduction

For Japanese–English machine-translation, especially for creative and localized content such as subtitles, MT evaluation must consider not only adequacy and fluency but also how style changes across languages. Recent neural metrics such as COMETKiwi (Rei et al., 2022), MetricX-24 (Juraska et al., 2024), and GEMBA-DA (Kocmi and Federmann, 2023) are widely used to assess adequacy and fluency, but provide little diagnostic information about such stylistic changes. Style depends on interlocutor, situation, genre, and cultural convention (Brown and Levinson, 1987; Blum-Kulka, 1987): politeness, intimacy, authority, emotional stance, and directness are encoded differently in Japanese and English , and can shift independently of one another, so that a single scalar cannot capture which aspect changed. This is especially salient in Japanese–English creative translation, including anime, manga, games, and audiovisual media, where translators often adjust register, character voice, and interpersonal stance to satisfy localization policies, length constraints, genre expectations, and reader experience (Venuti, 1995; House, 2015; Díaz Cintas and Remael, 2007). Thus, stylistic change is not necessarily an error, but can be a natural and even necessary part of translation.

Source (JA).
いかん！お前は今死んではいかん！
(Noooo! You must not die now!)
SDP(s): pol= −2, emo= +2, dir= +1

Hyp A over-formalized; less preferred
"It would be inadvisable for you to die at this time."
Ours **SDP(t)**: pol= +1, emo= 0, dir= −1; **SFS**= 0.45
Baseline **COMETKiwi**= 0.85, **GEMBA-DA**= 0.82

Hyp B human-preferred
"Get away! You must not die now!"
Ours **SDP(t)**: pol= −2, emo= +2, dir= +1; **SFS**= 0.89
Baseline **COMETKiwi**= 0.82, **GEMBA-DA**= 0.85

Figure 1: **Motivating Japanese–English example.** Similar baseline scalar scores hide different stylistic shifts in a creative translation settings, which our SDP/SFS make explicit.

For this reason, the key question is not simply whether style is preserved, but how it shifts: whether a translated sentence became more polite, less emotional, more authoritative, or less direct. What is needed is a diagnostic representation of stylistic shift. We therefore propose the *Style Diagnostic Profile* (SDP) framework: SDP assigns each

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 17–45
November 9, 2026 ©2026 Association for Computational Linguistics

utterance a seven-dimensional bipolar style vector over interpretable axes, because stylistic shift is multi-faceted, a single scalar cannot indicate which dimension moved, and ΔSDP, the difference between target and source SDP vectors, identifies which stylistic dimensions changed and in which direction.

Figure 1 illustrates the motivation with a Japanese–English subtitle example. Two English translations of the same Japanese utterance receive similar scalar quality scores, yet human annotators prefer Hyp B. The difference is not primarily one of adequacy, Hyp A over-formalizes an urgent command, reducing emotional force and directness, whereas Hyp B better matches the source's interpersonal and affective profile. The SDP framework makes this contrast explicit, turning a vague sense that "the style is off" into an analyzable pattern.

Through scorer validation, controlled style perturbations, and a case study of professional JA→EN subtitles, we illustrate what an SDP layer can reveal about style in MT evaluation. We position SDP as a diagnostic layer alongside quality-oriented metrics, not as a new adequacy metric. Because it maps stylistic change onto interpretable axes, such a layer may also support human-in-the-loop review workflows (Wu et al., 2022), where making stylistic feedback explicit is valuable(§7).

Our contributions are as follows:

- We propose the SDP framework for Japanese–English style diagnostics: a per-utterance seven-axis bipolar style profile, and ΔSDP, a difference vector that identifies which dimensions changed and in which direction. We validate an LLM-based SDP scorer against bilingual human annotations on professional JA→EN subtitles (§3, §5).

- We introduce Style25, a controlled JA↔EN style-perturbation benchmark, showing that targeted perturbation categories occupy distinct regions in ΔSDP space, i.e., controlled evidence that SDP captures axis-specific stylistic shifts beyond scalar quality scores (§6.1).

- As a case study, we apply ΔSDP to professional JA→EN subtitles in Pro-JE and find evidence of systematic stylistic adaptation rather than simple style copying, with preliminary preference-relevant signal complementary to quality-oriented metrics (§6.2, §6.3).

2 Related Work

Quality-Oriented MT Metrics Widely used MT metrics, including reference-based measures such as BLEU (Papineni et al., 2002) and ChrF++ (Popović, 2017), learned metrics such as COMETKiwi (Rei et al., 2022) and MetricX-24 (Juraska et al., 2024), and LLM-based judges such as GEMBA-DA (Kocmi and Federmann, 2023), summarize translation quality as scalar scores. These metrics answer a different question from ours: whether a translation is good, not how its style changed and in which direction. We therefore treat these metrics as complementary quality-oriented filters rather than baselines for SDP to outperform: when adequacy or fluency is severely degraded, human judgments are dominated by quality problems rather than stylistic differences. §6.1 shows that, once adequacy is controlled, these metrics are largely insensitive to targeted stylistic shifts.

Style Preservation in Translation A separate line of work evaluates style directly. Style-transfer evaluation tests whether a target attribute, such as formality, changes while meaning is preserved (Rao and Tetreault, 2018; Briakou et al., 2021). In MT, RASTA (Havaldar et al., 2025) evaluates cross-cultural style alignment along politeness, intimacy, and formality, and reports that machine translation tends to neutralize stylistic variation, including translation to and from Japanese. Other work studies whether Big-Five personality traits survive Japanese–English translation (Fu et al., 2026), or evaluates style in translation in terms of semantic preservation, cultural specificity, and expression style (Tan et al., 2025). MQM-style frameworks can flag register or tone mismatches (Mariana et al., 2015; Li et al., 2025), but they mark that a stylistic problem exists rather than represent its signed direction. Our goal is therefore not to judge stylistic shifts as errors, but to represent their direction and magnitude at the utterance level.

Translation Studies Translation studies has long treated translation as a contextualized transformation for a new audience, rather than a literal transfer of meaning (House, 2015; Venuti, 1995; Reiss and Vermeer, 2013). This view is especially relevant to Japanese–English creative and subtitle translation, where character voice, register, and localization policy can motivate deliberate stylistic shifts (Díaz Cintas and Remael, 2007). SDP operationalizes this stance by describing how style moves along each

Axis	−2	0	+2
Politeness	Casual	Neutral	Formal
Roughness	Gentle	Neutral	Crude
Emotion	Flat	Neutral	Emotional
Authority	Submissive	Equal	Authoritative
Age	Childlike	Neutral	Mature
Confidence	Hedging	Neutral	Assertive
Directness	Indirect	Neutral	Direct

Table 1: Seven bipolar stylistic axes with pole labels. Functions: Politeness (social distance), Roughness (verbal aggression), Emotion (affective intensity), Authority (power direction), Age (age-graded language), Confidence (epistemic certainty), Directness (communicative directness). Each axis is scored on a five-point scale (-2 to $+2$) for both source and target utterances.

axis before judging whether the shift reflects adaptation, degradation, or preference-relevant variation.

3 Style Diagnostic Profile

3.1 Style Dimensions

To make stylistic movement explicit in Japanese–English translation, SDP represents each utterance as a seven-axis over $[-2, +2]$, with 0 as the neutral (Table 1). The signed value encodes both the direction and the magnitude of a stylistic property. The seven axes were developed through an inductive analysis of a 34,264-line Japanese–English anime subtitle corpus covering 37 works. After removing low-style-content utterances such as formulaic greetings, brief acknowledgments, and duplicates, we organized utterances that reveal character voice and interpersonal stance into recurrent bipolar contrasts. We then selected the final set of axes using four criteria: bipolar contrast, conceptual separability, cross-language applicability, and attestation across diverse types of anime and settings. We treat the axes as designed diagnostic contrasts for human-interpretable reporting. Appendix A.2 shows that empirical factors cut across the designed axes and that the axes exhibit no severe redundancy.

Because these axes are intended for diagnostic interpretation, they are defined as graded dimensions rather than categorical labels. For example, the Age axis does not predict chronological age; following Eckert (1997), it captures age-related linguistic style as a social position on a five-point scale from youthful to mature. Appendix A.5 provides worked examples of stylistically similar but distinct shifts.

3.2 Profile Representation

Given a source utterance s and its translation t, the scorer assigns each a seven-axis vector:

$$\mathrm{SDP}(x) = (\mathrm{SDP}_1(x), \ldots, \mathrm{SDP}_K(x)), \quad (1)$$

where $K = 7$ is the number of style axes and each component $x_k \in \{-2, -1, 0, +1, +2\}$ is the score on axis k. The diagnostic profile is the difference between target and source profiles:

$$\Delta_{\mathrm{SDP}}(s, t) = \mathrm{SDP}(t) - \mathrm{SDP}(s). \quad (2)$$

A zero component means that the source and translation receive the same score on that axis, whereas a non-zero component indicates a perceived style shift. The sign of the component specifies the direction of the shift.

3.3 Auxiliary Scalar Summary: Style Fidelity Score

SDP and Δ_{SDP} are the main representations in this work. However, some controlled perturbation experiments require a single scalar for ranking candidates. For this limited purpose, we define the Style Fidelity Score (SFS) as:

$$\mathrm{SFS}(s, t) = 1 - \frac{1}{K} \sum_{k=1}^{K} \frac{|\Delta_{\mathrm{SDP}_k}(s, t)|}{R}, \quad (3)$$

where $R = 4$ is the maximum possible absolute difference between two scores on the $[-2, +2]$ scale. Thus, SFS $\in [0, 1]$. SFS is a lossy scalar summary of stylistic closeness, not a measure of translation quality or translation appropriateness. Professional subtitle translation may deliberately adapt style to the target language or audience (§6.2), and such adaptation can lower SFS even when the translation is appropriate.

4 Experimental Setup

This section describes the datasets, human annotations, metrics, baselines, and evaluation protocols used in our experiments. We use three Japanese–English subtitle resources for complementary evaluations: Subtitle-100 for scorer validation, Style25 for controlled JA↔EN style perturbations, and Pro-JE for analyzing how SDP dimensions and SFS relate to human preference. All human annotations were conducted by paid in-house bilingual

Japanese–English annotators, not crowdworkers. To clarify the data contribution: the underlying utterances come from existing resources (professional anime subtitles for Subtitle-100 and Pro-JE; OpenSubtitles (Lison and Tiedemann, 2016) for Style25, not redistributed), whereas the style-perturbed variants (in Subtitle-100 and Style25), the Pro-JE LLM candidates, and all seven-axis SDP and BWS annotations are newly created through our in-house bilingual workflow.

We compare three metrics families. The SDP family consists of SDP, a seven-axis diagnostic profile, and SFS, a scalar style-fidelity score computed from ΔSDP. SA-3 is a controlled three-axis ablation based on the three style axes used in RASTA (Havaldar et al., 2025), using the same scorer and distance-to-fidelity transformation as SFS.[1] Quality-oriented baselines include COMETKiwi, MetricX-24, and GEMBA-DA[2].

4.1 Subtitle-100: Scorer Validation

Subtitle-100 contains 100 professional JA→EN anime subtitle pairs. Three bilingual annotators, masked to the condition, independently rate each pair on all seven axes using the five-point bipolar scale (36.5 annotator-hours total). Of the 100 pairs, 25 retain the original professional subtitle (*Official* condition) while the remaining 75 use stylistically modified versions (*Modified* condition), created by rewriting the official subtitle along one or more axes (Appendix A.1); the larger Modified set provides more perturbed items for the SFS separation test, while the Official set anchors scorer validation against unmodified professional style (Appendix D).

4.2 Style25: Controlled Style Perturbations

Style25 contains 200 source utterances: 100 items in each direction for JA⇔EN subtitle items sampled from OpenSubtitles, used to obtain broad subtitle dialogue beyond anime for automatic sensitivity tests; the original subtitle text is not redistributed. For each seed translation, we use Gemma3-27B-IT (Team et al., 2025) to rewrite it into 25 target styles, grouped into 9 categories (Table 9), at two intensity levels via a one-shot style-conditioned prompt (Appendix B.2). To keep candidates comparable in adequacy, each rewritten variant must pass a COMETKiwi quality gate (≥ 0.80); variants below the threshold are regenerated when possible or otherwise discarded, so that surviving candidates differ from the unmodified translation (the anchor) mainly in style rather than meaning.

For each item, we rank the anchor together with its style-perturbed variants, and define Anchor-Recovery Preservation@1 (Pres@1) as the fraction of items for which a metric ranks the anchor first. Since the anchor is a context-aware translation tuned to the work's overall tone, it is expected to be closer to the source style than any single-style variant, so a style-sensitive metric should favor it on average. We therefore read Pres@1 as a metric's aggregate sensitivity to controlled style shifts, not as a measure of quality, preference, or ideal style copying; it need not reach 1.0, since a perturbed variant may occasionally align with the source style by chance.

The primary comparison is SFS vs. SA-3, which share the same scorer and distance-to-fidelity transformation and thus isolate the effect of the axis set. MetricX-24 and GEMBA-DA serve as quality-oriented baselines; COMETKiwi is excluded as a selector, as it is already the generation-time quality gate.

4.3 Pro-JE: Human Preference in Professional Subtitles

Pro-JE contains 32 professional JA→EN anime subtitle items, separate from Subtitle-100. Each item has four candidates: the official human subtitle and outputs from three LLMs (Qwen3-14B-Instruct (Yang et al., 2025), Gemma3-12B-IT, and Llama3.1-8B-Instruct (Grattafiori et al., 2024); Appendix B.3). Four bilingual annotators judged them using Best–Worst Scaling (BWS; Louviere et al., 2015). We use Pro-JE as a case study of how particular SDP dimensions and SFS relate to human preference, not to argue that style preservation is generally preferred.

We compare metric-induced rankings against human BWS preferences, reporting Kendall's τ and Top-1 accuracy (with fractional credit for ties; Appendix F). For scalar metrics, higher scores indicate either better predicted quality (COMETKiwi, MetricX-24, GEMBA-DA) or greater source-style closeness used diagnostically (SFS, SA-3); SDP itself is a seven-axis diagnostic profile, not a scalar ranking score.

[1] SA-3 is not a reproduction of RASTA, but an axis-set ablation under our scoring framework.

[2] We use a reference-free DA-style prompt that asks an LLM for a single 0–100 quality score given the source and candidate

5 Validating the LLM-based SDP Scorer

Because all downstream SDP analyses depend on reliable axis assignment, and LLM-as-a-judge reliability is not guaranteed (Zheng et al., 2023), we validate candidate scorers against human annotations before applying SDP in our diagnostics.

We use Subtitle-100, which contains human annotations for 100 professional JA→EN anime subtitle pairs on all seven SDP axes. Each source and target utterance was scored separately on each axis ($100 \times 2 \times 7 = 1{,}400$ labels), and the scorer was queried once per utterance–axis pair using the same five-point axis rubric. For the human reference, three bilingual annotators independently scored each item; we averaged their scores and rounded to the nearest integer on the five-point scale to obtain the final label.

Human agreement is moderate overall, but varies by axis: authority is relatively reliable, whereas age and directness are weaker. The mean Krippendorff's α across the seven axes is 0.611 for Japanese and 0.492 for English (overall mean 0.551), comparable to prior work on pragmatic and social meaning annotation (Pei and Jurgens, 2020). Detailed inter-annotator agreement and axis-level validation results are reported in Appendix C.

Table 2 compares three open-weight candidate scorers from different model families and scales, together with two non-adaptive baselines against human labels. We report five agreement metrics: exact-match accuracy, within-one (±1) accuracy on the five-point scale, mean absolute error (MAE), root mean squared error (RMSE), and the mean per-axis Spearman correlation (ρ). Qwen3-14B-Instruct (Yang et al., 2025) achieves the best overall agreement with human annotations: exact 55.6%, within-one 93.9%, MAE 0.51, RMSE 0.81, and mean per-axis Spearman ρ=0.47. The all-zero baseline always predicts the neutral label, while the majority-vote baseline predicts the most frequent human label for each axis. Because both baselines make constant predictions, their Spearman correlations are undefined.

We therefore use Qwen3-14B-Instruct as the SDP scorer in the downstream analyses. This validation supports using its outputs as automatic style diagnostics, while results involving weaker axes such as age and directness should be interpreted cautiously. The scorer also shows a systematic zero bias relative to humans: it outputs the neutral label 0 more often than human annotators, with the largest gap on age and politeness (Appendix C.7). This bias should be kept in mind when interpreting near-zero ΔSDP values as an absence of style shift.

Model	Exact↑	±1↑	MAE↓	RMSE↓	ρ ↑
Qwen3-14B-Instruct	**55.6**	**93.9**	**0.51**	**0.81**	**0.47**
GPT-oss-120B	46.1	88.4	0.69	1.03	0.37
Gemma3-9B	48.9	87.5	0.67	1.04	0.39
All-zero baseline	45.5	88.6	0.66	0.94	n/a
Majority-vote baseline	49.8	88.9	0.63	0.95	n/a

Table 2: LLM SDP scorer validation on Subtitle-100. Exact and ±1 accuracy are percentages; ρ is the mean per-axis Spearman correlation. Constant baselines have undefined ρ due to zero prediction variance.

6 Results and Analysis

Using the validated SDP scorer (§5), we report three complementary analyses. §6.1 uses the controlled Style25 benchmark to test *whether ΔSDP is diagnostically sensitive to deliberate style perturbations*, i.e., whether perturbation categories occupy distinct regions in ΔSDP space. §6.2 applies ΔSDP to professional Pro-JE subtitles to examine *how style is adapted in practice*. §6.3 relates these shifts to human preference to ask *which dimensions carry preference-relevant signal*. We treat Style25 as primary controlled evidence and Pro-JE as a case study.

6.1 Controlled JA↔EN Style Sensitivity

We use Style25 to test whether perturbation categories produce category-specific shifts in ΔSDP space, and whether these shifts differ between JA→EN and EN→JA. The full seven-axis ΔSDP is the diagnostic space; for visualization, axes that move together under a common stylistic direction can be aggregated into one interpretable dimension. Figure 2 uses one such two-dimensional view:

$$\textit{interpersonal force} = \text{mean}(\Delta\text{dir}, \Delta\text{aut}, -\Delta\text{pol}),$$

$$\textit{expressive intensity} = \text{mean}(\Delta\text{emo}, \Delta\text{rou}, \Delta\text{con}).$$

These two axes are an exploratory visualization motivated by the interpersonal and affective characteristics that are especially salient in anime dialogue. Age is omitted here as it contributes little discriminative signal for these categories, but remains part of the full representation. This projection is only a task-specific visualization: the category separation is not an artifact of it, as a data-driven PCA on the full seven-axis ΔSDP reproduces the same structure,

with the first two components explaining most of the variance (Appendix H).

The 25 individual styles are grouped into nine categories. The community-register category has the largest expressive-intensity shift, while pragmatic strategy stays near the origin. The anchor is not at the origin, indicating that naturally occurring translations can already involve cross-lingual stylistic shifts rather than being style-perfect.

Anchor recovery as an auxiliary diagnostic. As an auxiliary diagnostic, we test whether each metric ranks the anchor above its style-perturbed variants (Pres@1, Rank@3, Rank@5; §4.2); high recovery indicates sensitivity to controlled perturbations, not higher quality or human preference.

Table 3 shows that anchor recovery depends on metric type and direction. Both style metrics exceed chance (5.7% for JA→EN, 3.7% for EN→JA) by a wide margin. SFS outperforms SA-3 in JA→EN, whereas SA-3 is slightly higher in EN→JA, suggesting that the most useful style cues differ by direction. GEMBA-DA appears strong under default ranking, but largely due to ties from heavily discretized scores (Appendix D), while MetricX-24 rarely selects the anchor. These results characterize metric behavior under controlled perturbations, not human preference for the anchor.

Category-specific sensitivity Anchor recovery also varies by perturbation category (Table 18 in Appendix E.4). SFS is stronger than SA-3 for categories involving dimensions beyond formality and politeness, such as dialect, locale, character voice, and speech style. SA-3 is competitive for categories more directly tied to formality or politeness, such as register and stance. This pattern is consistent with the seven-axis design: broader axes help when perturbations affect multiple aspects of character voice, while a smaller formality-oriented set is sufficient for narrower shifts.

6.2 Professional Style Adaptation in Pro-JE

This case study asks how professional translators adapt style in practice. For the 32 professional JA→EN anime subtitles in Pro-JE, we score source and target on all seven axes with the validated scorer, compute the per-axis mean ΔSDP (target minus source), and read each axis as preserved (near-zero shift) or adjusted (Table 4).

Emotion and directness remain nearly unchanged (average shifts -0.02 and $+0.03$), suggesting that translations preserve the affective and structural core of the utterance. In contrast, interpersonal register is adjusted: politeness ($+0.24$), authority ($+0.19$), and roughness ($+0.17$) all move toward a more neutral target profile.

These changes are consistent with the nature of anime dialogue, where reduced-politeness or very gentle soft source-side expressions are rendered with a more neutral tone. Overall professional subtitle translation in this sample preserves emotion and directness while normalizing some interpersonal register cues for the target language. A broader comparison with controlled Style25 perturbation profiles is provided in Appendix E.3.

6.3 Relation to Human Preference

As a secondary check on Pro-JE, we relate style shift to human preference on a set that contains professional JA→EN subtitles and outputs from three LLMs, together with human BWS judgments. We use this setting not to claim that source-style preservation is always preferred, but to examine which style dimensions carry preference-relevant signal.

On average, the Human translation is most preferred (group-mean BWS $+0.358$; Table 26 in Appendix I). It also shows the smallest emotion shift ($|\Delta|$=0.22), while Llama, which has the largest directness shift ($|\Delta|$=1.07), ranks last (-0.345). At the same time, the contrast is not absolute: the LLM outputs are often fluent and adequate, and some items do not yield a clear human-translation advantage. This makes Pro-JE a useful but limited setting for testing whether style contributes information beyond standard quality signals.

A representative Pro-JE item illustrates how ΔSDP can explain preference patterns missed by scalar quality scores. For the source utterance すげえ 野望のサイズがデカい (“the scale of that ambition is huge”), COMETKiwi ranks the Qwen3-14B-Instruct candidate above the official subtitle (0.83 vs. 0.79), whereas human BWS strongly prefers the official subtitle ($+1.00$ vs. 0.00). The ΔSDP profile shows that both translations reduce roughness and directness, but the Qwen3-14B-Instruct candidate additionally raises politeness and confidence ($+1$ each), making the line more standard and assertive than the source character voice. The full example is shown in Table 25.

SFS shows a modest positive association with human preference using nominal significance thresholds ($\tau = +0.147$, uncorrected $p < .05$). This association appears complementary to quality met-

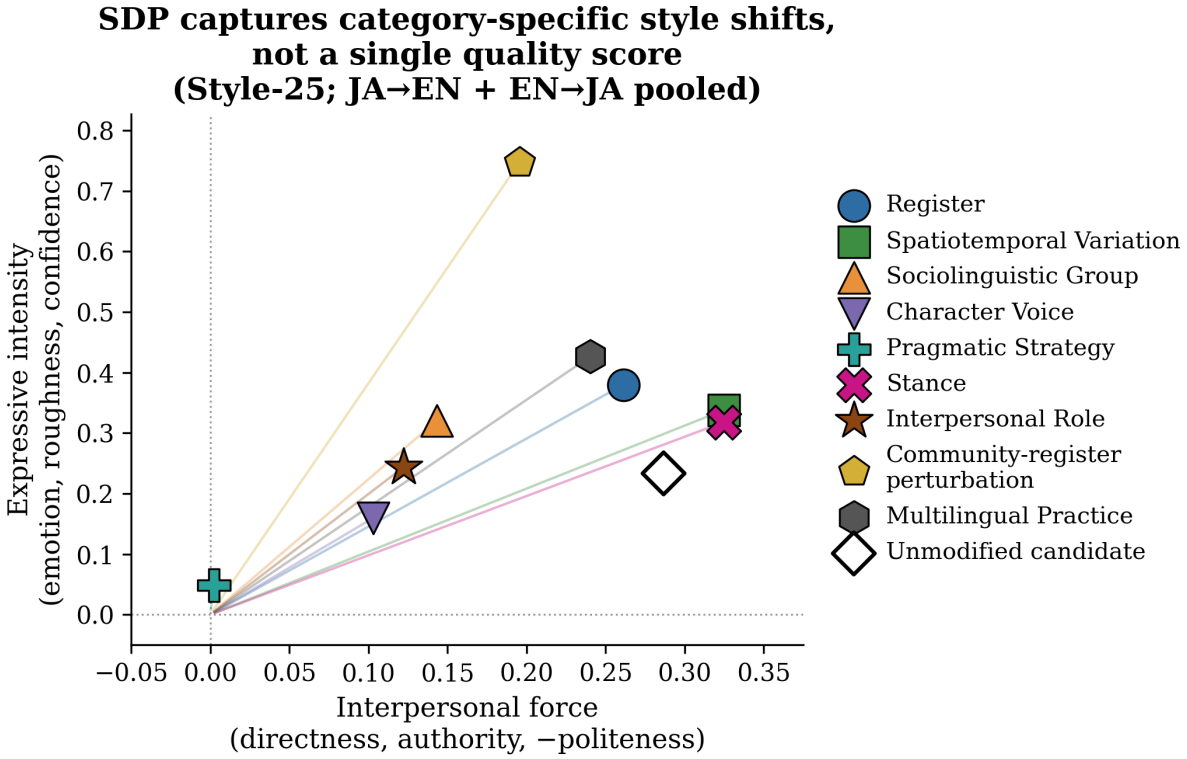


Figure 2: **Interpersonal–expressive style plane (Style25, JA↔EN pooled).** Each point is a Style25 category placed by its mean ΔSDP after projecting the seven-axis representation onto two visualization axes; the white diamond is the unmodified OpenSubtitles anchor. Categories occupy distinct regions, indicating that ΔSDP captures category-specific stylistic shifts rather than collapsing them into a single quality score.

rics: on Pro-JE, COMETKiwi itself is uncorrelated with BWS ($\tau \approx -0.008$), while the SFS association strengthens after quality gating ($\tau = +0.225$, uncorrected $p < .05$). GEMBA-DA itself attains a comparable or slightly higher τ on this small sample ($\tau = +0.160$, uncorrected $p < .05$; Appendix F.1); none of these associations survive Holm–Bonferroni correction for multiple comparisons (Appendix F.1, F.2), so we treat them as case-study observations rather than confirmatory findings. SFS's value lies in explaining preference through an interpretable, quality-independent axis rather than in exceeding quality-metric correlation. We interpret this as case-study evidence that source-style closeness can carry preference-relevant signal in some cases, not that style preservation is generally optimal. Full per-metric correlations and quality-gate analyses are reported in Appendix F.

7 Discussion

SDP should be interpreted as a diagnostic complement to quality-oriented MT metrics, not a replacement: metrics such as COMETKiwi, MetricX-24, and GEMBA-DA provide scalar quality judgments, whereas ΔSDP describes how style moves across axes.

We are deliberately careful to separate what our evidence supports at different strengths. Style25 is a controlled, automatic test at scale, and it shows that perturbation categories produce category-specific shifts in ΔSDP space rather than collapsing into a single quality-like dimension. The Pro-JE preference result is intentionally narrower: on 32 items, style fidelity carries preference-relevant signal complementary to scalar quality scores, but the associations are uncorrected and do not survive multiple-comparison correction (§6.3). We therefore treat Style25 as our primary controlled evidence for the SDP framework's diagnostic sensitivity and Pro-JE as a case study, not as evidence that source-style preservation is generally optimal.

A further potential use case is human-in-the-loop translation review. Entertainment localization often involves multi-person workflows, where translators, editors, and quality-control reviewers may differ in their access to source-language nuance. SDP could help make otherwise vague stylistic feedback more explicit by mapping perceived changes onto

Metric	JA→EN			EN→JA			Type
	Pres@1	Rank@3	Rank@5	Pres@1	Rank@3	Rank@5	
GEMBA-DA	74.0	77.0	81.0	36.0	40.0	41.0	quality
SFS (7-axis)	40.0	57.0	69.0	20.0	35.0	42.0	style
SA-3 (RASTA)	31.0	52.0	62.0	24.0	40.0	47.0	style
MetricX-24	4.0	19.0	31.0	0.0	3.0	4.0	quality

Table 3: Anchor-recovery behaviour (Pres@1, Rank@3, Rank@5, %) on Style25. Pres@1/Rank@3/Rank@5 measure how often each metric ranks the unmodified OpenSubtitles anchor first, within the top 3, or within the top 5, respectively, among its deliberately style-perturbed variants. These are diagnostics of sensitivity to controlled perturbations, not a quality ranking, a human-preference result, or a claim that style copying is desirable. Chance for Pres@1 is approximately 5.7% for JA→EN and 3.7% for EN→JA, based on the average number of candidates after filtering (approximately $3\times$ and $5\times$ these values for Rank@3 and Rank@5, respectively). COMETKiwi is used as a generation-time quality gate and is therefore not included as a selector.

Axis	Src (JA)	Tgt (EN)	Shift
Politeness	-0.21	$+0.03$	$+0.24$
Authority	-0.11	$+0.08$	$+0.19$
Roughness	-0.45	-0.28	$+0.17$
Age	$+0.36$	$+0.24$	-0.12
Confidence	$+0.39$	$+0.47$	$+0.08$
Directness	$+0.32$	$+0.35$	$+0.03$
Emotion	$+0.33$	$+0.31$	-0.02

Table 4: Professional subtitle adaptation by axis in Pro-JE (Human, $N{=}32$). Emotion and directness show near-zero average shifts, while politeness, authority, and roughness show larger interpersonal-register adjustments.

interpretable axes, such as politeness, directness, and emotional intensity. This suggests a possible role for SDP as an orientation aid for post-editing and review, although validating such use requires dedicated user studies.

8 Conclusion

This paper asked whether stylistic change in translation can be made visible and analyzable, rather than collapsed into a single preservation or quality score. We introduced the SDP framework, a seven-axis per-utterance style profile together with ΔSDP, its signed source–translation style-shift vector, and Style25, a controlled JA↔EN style-perturbation benchmark. Together, these tools allow stylistic shifts to be diagnosed in terms of which dimensions changed and in which direction.

Our analysis suggests that ΔSDP captures category-specific stylistic shifts that scalar quality metrics cannot express. In our Pro-JE sample, professional JA→EN subtitles show evidence of systematic style adaptation rather than simple style copying. We also find uncorrected, case-study-level evidence that SFS carries preference-relevant signal after quality filtering, suggesting that style-sensitive signals can complement adequacy-oriented metrics. Throughout, our aim is descriptive rather than prescriptive: SDP is designed to show *how* style moved along each axis, not to judge whether that movement is inherently good or bad.

These findings suggest that signed style profiles can support more interpretable evaluation of Japanese–English creative and localized translation, where stylistic shifts may be acceptable or even desirable. Future work should improve LLM-based SDP scorer calibration, extend validation to other Asian-language pairs, and develop context-aware SDP scoring. The present scorer is intentionally lightweight, relying only on source–target utterance pairs, but stylistic acceptability in subtitles may depend on broader discourse context, speaker relationships, visual cues, and source-side vocal performance. We also plan to explore ΔSDP as structured feedback for post-editing or localization review.

Limitations

This study focuses on Japanese–English creative translation, especially subtitle-style data. The seven SDP axes were developed from Japanese–English anime subtitles, so other domains, genres, or language pairs may require different axes or subsets.

SDP is context-limited: scores are assigned from source–target utterance pairs without surrounding discourse, speaker identity, addressee information, scene context, or subtitle length constraints. Interpersonal axes such as politeness, authority, and directness therefore reflect perceived utterance-level stance, not verified claims about speaker relationships or the full communicative situation.

The LLM-based SDP scorer is not a substitute

for human judgment. It can attenuate shift magnitude, especially on axes with lower human–LLM agreement such as age and directness. Relatedly, our scorer selection is itself limited in scope: we compare only openly available models, motivated by reproducibility and computational-resource constraints, and leave an exhaustive search over scorers, including proprietary or larger models, to future work.

Finally, the human preference analysis is case study and Style25/Pres@1 should be interpreted as diagnostic sensitivity tests rather than translation-quality measures.

Ethics Statement

All human annotations were produced by professional bilingual Japanese–English data annotators through our in-house workflow rather than crowdworkers; they were informed of the research use and compensated as part of their regular paid work (§4). Style25's synthetic community-register condition (§4.2) is a controlled stress test, not a depiction of any real community, identity group, or protected category.

References

Shoshana Blum-Kulka. 1987. Indirectness and politeness in requests: Same or different? *Journal of Pragmatics*, 11(2):131–146.

Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021. Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer. In *Proceedings of EMNLP*, pages 1321–1336.

Penelope Brown and Stephen C. Levinson. 1987. *Politeness: Some Universals in Language Usage*. Cambridge University Press.

Jorge Díaz Cintas and Aline Remael. 2007. *Audiovisual Translation: Subtitling*. St Jerome.

Penelope Eckert. 1997. Age as a sociolinguistic variable. In Florian Coulmas, editor, *The Handbook of Sociolinguistics*, pages 151–167. Blackwell.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

Xinger Fu, Masaaki Nagata, and Chenhui Chu. 2026. A study on speaker individuality preservation in English–Japanese translation. In *Proceedings of the 32nd Annual Meeting of the Association for Natural Language Processing*, pages 4162–4167. In Japanese.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shreya Havaldar, Adam Stein, Eric Wong, and Lyle Ungar. 2025. Towards style alignment in cross-cultural translation. In *Proceedings of ACL*, pages 32213–32230.

Juliane House. 2015. *Translation Quality Assessment: Past and Present*. Routledge.

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 and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203.

Yunmeng Li, Jun Suzuki, Makoto Morishita, Kaori Abe, and Kentaro Inui. 2025. MQM-chat: Multidimensional quality metrics for chat translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3283–3299, Abu Dhabi, UAE. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of LREC*, pages 923–929.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best–Worst Scaling*. Cambridge University Press.

Valerie Mariana, Troy Cox, and Alan Melby. 2015. The multidimensional quality metrics (MQM) framework: A new framework for translation quality assessment. *The Journal of Specialised Translation*, 23:137–161.

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.

Jiaxin Pei and David Jurgens. 2020. Quantifying intimacy in language. *Preprint*, arXiv:2011.03020.

Maja Popović. 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.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of NAACL-HLT*, pages 129–140.

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.

Katharina Reiss and Hans J. Vermeer. 2013. *Towards a General Theory of Translational Action: Skopos Theory Explained*. Routledge.

Xin Tan, Bowei Zou, and AiTi Aw. 2025. A benchmark for translations across styles and language variants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2389–2402, Suzhou, China. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Lawrence Venuti. 1995. *The Translator's Invisibility: A History of Translation*. Routledge.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 46595–46623.

The appendices are organized as follows. Appendix A provides full axis definitions, the degeneracy analysis motivating the seven-axis design, and worked diagnostic case examples; Appendix B and Appendix C detail the datasets and LLM scorer validation; Appendix D and Appendix E.1 report additional statistical analyses on Style25; Appendix F reports the full Pro-JE preference and quality-gate analysis; Appendix G presents direction-specific style planes; and Appendix I presents complete case studies.

A Axis Definition

A.1 Full Axis Definitions with Anchor Examples

Each axis is scored on a five-point bipolar scale (-2 to $+2$). Table 5 provides the scale-point descriptions used during annotation and LLM scoring. Table 6 shows bilingual anchor utterances for one representative axis (politeness); anchors for all axes follow the same structure and are included in the LLM prompt (Appendix C.5).

A.2 Degeneracy Analysis: Why 7 Axes

To assess whether the 7 axes are redundant, inter-axis correlations and Variance Inflation Factors (VIF) were examined from human annotation data ($N = 100$ items, 3-annotator means). VIF values are reported in Table 7; all are below 4.0, indicating acceptable collinearity.

Inter-axis correlation structure. Three interpretable clusters emerge: (1) **Dominance cluster**: roughness–authority ($\rho = .75/.70$) and authority–confidence ($\rho = .54/.50$). (2) **Politeness–roughness**: inversely correlated ($\rho = -.56/-.48$) (Brown and Levinson, 1987). (3) **Peripheral axes**: age and directness show low to moderate correlation. Correlated axes are not merged: a character can be *authoritative yet gentle*, which a merged axis could not represent.

A.3 Empirical Validation of Three-Layer Structure

To validate the conceptual three-layer organization (Table 8), we conducted exploratory factor analysis, hierarchical clustering, and reliability analysis on human-annotated Subtitle-100 data (N=100 utterances, 3 annotators).

Correlation matrix. Spearman correlations reveal heterogeneous within-layer relation-

Axis	Function	-2	-1	0	$+1$	$+2$
Politeness	Social distance	Very Casual: zero social distance; friends/family	Casual: close but with consideration; relaxed settings	Neutral: standard social distance; public	Formal: clear hierarchy; business	Very Formal: maximum respect; ceremonies
Roughness	Verbal aggression	Very Gentle: comforting; tender word choice	Gentle: calm tone; respectful	Neutral: objective; neither aggressive nor soft	Somewhat Rough: warning/irritation; mild reprimands	Very Rough: gangster/pirate speech; vulgar language
Emotion	Affective intensity	Very Restrained: robot-like; textbook reading	Somewhat Restrained: calm professional; news reporting	Neutral: standard; moderate expression	Somewhat Emotional: recognizable emotion (joy, anger)	Very Emotional: screaming; extreme excitement
Authority	Power direction	Very Submissive: absolute obedience; pleading	Somewhat Submissive: subordinate; polite requests	Equal: colleagues; mutual exchange	Somewhat Authoritative: teacher; giving instructions	Very Authoritative: warden/dictator; commands
Age	Age-graded language	Very Childlike: baby talk; language errors	Youthful/Teen: teen slang; SNS-style	Neutral: standard adult; no age markers	Mature: professional adult speech	Very Mature/Archaic: historical drama; elderly speech
Confidence	Epistemic certainty	Very Uncertain: timid; avoids assertions	Somewhat Uncertain: cautious; some hedging	Neutral: balanced; fact-based	Somewhat Confident: expert opinion; clear	Very Confident: absolute claims; no doubt
Directness	Communicative directness	Very Indirect: heavy hedging; implies rather than states; avoids the point	Somewhat Indirect: softened suggestions; qualified requests	Neutral: states the point plainly, without extra softening or bluntness	Somewhat Direct: clear statements; direct requests	Very Direct: blunt imperatives; explicit commands; no hedging

Table 5: Seven bipolar stylistic axes with full scale-point descriptions.

ships (pooled across JA source and EN target):**Interpersonal**: Roughness–Authority strongly correlate ($r = 0.73$), but Politeness negatively correlates with both Roughness ($r = -0.50$) and Authority ($r = -0.48$).**Affective-Epistemic**: Emotion–Confidence show weak positive correlation ($r = 0.22$). **Expressive-Structural**: Age–Directness show weak *negative* correlation ($r = -0.35$).

Exploratory factor analysis (3 factors, oblimin rotation). The empirical factor structure differs from the three-layer organization, consistent with the orthogonal axis design: **Factor 1** ("Interpersonal Forcefulness"): high loadings on Roughness ($+0.92$), Authority ($+0.94$), and Politeness (-0.73). **Factor 2** ("Maturity & Assertiveness"): high loadings on Age ($+0.77$) and Confidence ($+0.72$), crossing layer boundaries. **Factor 3** ("Linguistic Simplicity"): dominated by Directness ($+0.96$) with secondary Emotion loading ($+0.44$). Variance explained: 75.4% (Japanese), 74.9% (English).

Reliability analysis (Cronbach's α). Cronbach's α reflects the designed axis orthogonality: **Interpersonal**: $\alpha = -0.39$ (Japanese), -0.35 (English); negative values result from inverse correlations (Politeness vs. Roughness/Authority). **Affective-Epistemic**: $\alpha = 0.34$ (Japanese), 0.22 (English); low but positive, reflecting weak correlation between independent dimensions. **Expressive-Structural**: $\alpha = -0.92$ (Japanese), -1.34 (English); Age and Directness measure independent aspects.

Interpretation. The low and negative α values validate the intended orthogonal design. The three-layer organization serves as a conceptual scaffold for interpretation, not a claim of empirical factor cohesion.

Score	Japanese	English
-2	やー、パス	Nah, can't
-1	ごめん、ちょい難しいかも	Sorry, I can't make it.
0	ごめんなさいね、難しそうです	I have a slightly different view.
$+1$	大変申し訳ございませんが、お引き受け致しかねます	I sincerely apologize, but I'm unable to accommodate that.
$+2$	誠に恐縮ではございますが諸般の事情によりお受けいたしかねます	I deeply regret that circumstances prevent me from accepting.

Table 6: Bilingual anchor examples for politeness (representative). Anchors for all seven axes follow the same structure.

Axis	VIF (JA)	VIF (EN)
politeness	3.01	3.62
roughness	2.78	3.25
emotion	1.70	1.38
authority	3.69	3.26
age	1.50	2.32
confidence	1.51	1.40
directness	1.30	1.87

Table 7: Variance Inflation Factors (full $N = 100$ human-annotated items). All below 4.0 (conventional threshold: 5.0), indicating low collinearity among the seven axes.

Layer	Axes	Function
Interpersonal	Pol, Aut, Rou	Social relationship constraints
Affective Epistemic	Emo, Con	Speaker emotional and epistemic state
Expressive Structural	Age, Dir	Sociolinguistically marked variation

Table 8: Three-layer organization of stylistic axes. Pol = Politeness, Aut = Authority, Rou = Roughness, Emo = Emotion, Con = Confidence, Dir = Directness.

A.4 Example of Profile Representation

For illustration, consider the Japanese source おおいやだ毛虫じゃあるまいし and its professional translation "I ain't a caterpillar!". Suppose the scorer assigns the following non-zero SDP(s) and SDP(t) components:

$$s: \{\text{pol:}-1,\ \text{rou:}+1,\ \text{emo:}+1,\ \text{aut:}-1,\ \text{con:}+1,\ \text{dir:}+1\},$$
$$t: \{\text{pol:}-2,\ \text{rou:}+1,\ \text{age:}-1\}.$$

Axes omitted from the display are treated as zero. The resulting diagnostic profile is:

$$\Delta_{\text{SDP}}(s,t) = \{\text{pol:}-1,\ \text{emo:}-1,\ \text{aut:}+1,\ \text{age:}-1,\ \text{con:}-1,\ \text{dir:}-1\}.$$

Here, zero components are omitted. Non-zero components indicate axes on which the translation differs stylistically from the source. Thus, ΔSDP identifies where stylistic nuance changes, while the sign of each component indicates the direction of that change. A scalar score cannot express this axis-level information.

A.5 Diagnostic Case Examples

This section provides extended diagnostic case examples illustrating how per-axis human annotation captures specific stylistic deviations. Each case reports three rows on the seven SDP axes: the **Source** (Japanese source utterance, scored directly), the **Official** translation, and a stylistically **Modified** variant.

Case 1: Emotion and Roughness Flattening.
Source (JA): くそ！何やってんだよ！
Official (EN): "Damn it! What the hell are you doing?!"
Modified (EN): "I'm afraid your actions are inappropriate."

	Pol.	Rough.	Emo.	Auth.	Age	Conf.	Dir.
Source	-2	$+1$	$+2$	$+1$	-1	$+2$	$+1$
Official	-1	$+2$	$+2$	$+1$	0	$+1$	$+1$
Modified	$+1$	-1	-1	0	0	0	-1

Diagnostic interpretation: The modification shifts 6 of 7 axes, with the largest deviations on roughness ($|\Delta|$=3), emotion ($|\Delta|$=3), and politeness ($|\Delta|$=2). Directness also drops ($|\Delta|$=2) as the blunt imperative becomes a hedged, formal evaluation, while authority and confidence shift only mildly ($|\Delta|$=1 each). This identifies the stylistic distortion as *emotion flattening with register elevation*, distinct from a single-axis politeness shift or simple lexical substitution.

Case 2: Authority and Confidence Hedging.
Source (JA): 金輪際メガネなんてかけないからな

Official (EN): "I'm never wearing glasses again!"
Modified (EN): "I probably won't wear glasses anymore... I think."

	Pol.	Rough.	Emo.	Auth.	Age	Conf.	Dir.
Source	−1	+1	+1	+2	0	+2	+2
Official	−1	0	+1	+1	0	+2	+1
Modified	0	−1	0	−1	0	−1	−1

Diagnostic interpretation: Primary deviations occur on confidence ($|\Delta|$=3), authority ($|\Delta|$=2), and directness ($|\Delta|$=2). The hedging markers ("probably," "I think") shift epistemic certainty and power direction while increasing processing load, illustrating multi-dimensional stylistic degradation.

Case 3: Politeness and Directness Elevation.
Source (JA): たまには道場の方に寄っていきませんか？
Official (EN): "Maybe you could stop by the dojo?"
Modified (EN): "I would be honored if you might consider visiting our training facility at your convenience."

	Pol.	Rough.	Emo.	Auth.	Age	Conf.	Dir.
Source	0	−1	0	−1	0	−1	+2
Official	0	−1	0	0	0	0	+1
Modified	+2	−1	0	−1	+1	−1	−2

Diagnostic interpretation: The modification shifts politeness ($|\Delta|$=2) and directness ($|\Delta|$=3) while preserving emotion and roughness. This demonstrates axis independence: register and complexity can shift while affective dimensions remain stable.

B Dataset Details

B.1 Subtitle-100

Subtitle-100 is a scorer-validation set consisting of 100 professional JA→EN anime subtitle pairs. All items are taken from works originally produced in Japanese, with Japanese source utterances paired with their official English subtitle translations. Each item is annotated on the seven SDP axes for both the source and target utterances. We use this dataset to validate the LLM-based style scorer and to test whether the scorer's ΔSDP tracks known stylistic deviations.

B.2 Style25

Style25 is a controlled style-perturbation benchmark built in both JA→EN and EN→JA directions. For each direction, we sample 100 seed items from OpenSubtitles (Lison and Tiedemann, 2016). The seed translations are community-produced OpenSubtitles translations.

For each seed translation, Gemma3-27B-IT rewrites the translation into 25 target styles at two intensity levels (Table 9). Together with the unmodified OpenSubtitles translation, this yields up to 51 possible candidates per item before filtering. We apply a COMETKiwi quality gate (≥ 0.80) during generation: candidates below the threshold are regenerated when possible, but those that still fail are discarded. Therefore, fewer than 51 candidates may remain after filtering, with survival rates differing by direction.

We use Style25 to test whether automatic scores respond systematically to controlled style shifts and whether they can identify the unmodified OpenSubtitles translation as the no-perturbation anchor among deliberately style-perturbed variants. This evaluation is diagnostic and should not be interpreted as a measure of translation quality or human preference.

B.2.1 Style25 Prompt Specifications

Each of the 25 styles is defined for both translation directions at two intensity levels. Each level consists of an instruction and a set of examples (3 per level). At prompt-construction time, one example is selected at random from the valid examples to build a 1-shot prompt. The templates are shown below, followed by the full instructions and one representative example per category/intensity.

JA→EN template

```
Translate the following Japanese subtitle
lineinto English.
<DESCRIPTION_ON_STYLE>
Do not add or remove information from the
original. Output ONLY the English translation.

Example:
Japanese: <JAPANESE_EXAMPLE_SENTENCE>
English: <ENGLISH_EXAMPLE_SENTENCE>

Now translate:

Japanese: <SOURCE>
English:
```

EN→JA template

```
以下の英語のセリフを日本語に翻訳してく
ださい。
<DESCRIPTION_ON_STYLE>
原文の情報を追加・削除しないこと。
日本語訳のみを出力してください。

例：
English: <ENGLISH_EXAMPLE_SENTENCE>
日本語:<JAPANESE_EXAMPLE_SENTENCE>
```

#	Category	Style	Instruction (abbrev.)	Example (JA→EN, mild)
Register	01	Textbook	Clean, standard English; educational clarity; no slang.	もう関わりたくない。放っておいてくれ。 → I no longer wish to be involved. Please leave me alone.
	02	Academic	Academic register; nominalization, passive voice.	→ The subject expresses a desire to discontinue involvement and requests non-interference.
	03	Literary	Polished, novelistic prose; aesthetic word choice.	→ A weariness had settled over him—the kind that begs for silence and solitude.
	04	Tabloid	Punchy, sensational headline style.	→ ENOUGH! "Just back off and leave me alone!"
	05	Casual	Conversational tone; contractions, relaxed grammar.	→ Ugh, I'm so over this. Just lemme be, yeah?
	06	SNS	Social-media post; short, informal.	→ done with this lol just leave me alone
	07	Report	Official report; impersonal, third-person.	→ The individual has indicated a preference to discontinue engagement.
Spatiotemporal Variation	08	Regional-1	Regional variety A (Southern / Kansai).	→ I don't wanna deal with this no more. Just leave me be.
	09	Regional-2	Regional variety B (British / Hiroshima).	→ I'd rather not be involved any further, if you don't mind.
	10	Era	1940s period style (film-noir / wartime).	→ I'm through with this racket. Just leave me alone, will ya?
Sociolinguistic Group	11	Business	Corporate register; business jargon.	→ Per our earlier discussion, I'd like to disengage going forward.
	12	Elderly	Gentle, unhurried elderly speech.	→ I've had enough of all this fuss, really. Just let me rest.
	13	Gen-Z	Contemporary youth slang.	→ I literally cannot with this anymore, just leave me alone.
Character Voice	14	Deadpan	Flat, unemotional; short declaratives.	→ I do not want involvement. Leave me alone.
	15	Bitter	Sarcastic, cynical, world-weary.	→ Funny how I keep having to say this—I'm done. Leave me alone.
	16	Brooding	Quiet, introspective; restrained emotion.	→ …I just…I can't do this anymore. Please, just leave me be.
Operational Register	17	Community-register	Expressive, community-associated register.	→ Oh, I am just done with this whole thing. Leave me be, would you?
Multilingual Practice	18	Code-switching	Identity-driven code-switching across languages.	→ I'm done with this, c'est la vie. Just leave me alone.
Pragmatic Strategy	19	Blunt	Direct; no hedging or pleasantries.	→ Done. Leave me alone.
	20	Roundabout	Indirect, roundabout phrasing.	→ I've been thinking, and maybe it might be better if we went our separate ways.
	21	Hedging	Frequent uncertainty markers.	→ I think maybe I probably don't want to be involved anymore…
Stance	22	Assertive	Confident, declarative; no hedging.	→ I'm done. This is over. Leave me alone.
Interpersonal Role	23	Commanding	Authoritative tone; imperative forms.	→ That's enough. Stand down and leave me alone.
	24	Submissive	Timid, deferential toward a superior.	→ Um, I'm sorry, but…I think I'd like to be left alone…
	25	Mentoring	Warm, guiding mentor tone.	→ Listen, I've been where you are. But I need my space now.

Table 9: Style25 taxonomy and style definitions: 25 styles across 9 categories, using the same category names as the main-text figure. Instructions are abbreviated; each style is defined for both translation directions at two intensity levels (mild/strong). All examples share the same source sentence 「もう関わりたくない。放っておいてくれ。」 to illustrate stylistic contrast. The community-register perturbation is a synthetic operational stress test: it does not represent any particular real-world or protected community, and is unrelated to the human-produced OpenSubtitles translations used as unmodified references. Each style is assigned a numeric ID (01–25), matching the numbered points in the PCA figure (Figure 10 and 11).

次の文を翻訳してください：

`English: <SOURCE>`
日本語:

B.2.2 Style25 Style Definitions

The 25 styles are organized into 9 categories. Table 9 lists the full taxonomy together with an abbreviated instruction and one representative example (JA→EN, mild level) for each style. The `<DESCRIPTION_ON_STYLE>` slot in the templates above is filled with the corresponding instruction, and the example sentences are drawn from the examples associated with each style/intensity. For illustration, all examples in Table 9 share the same source sentence 「もう関わりたくない。放っておいてくれ。」 so that stylistic contrasts can be compared directly. Concretely, `<JAPANESE_EXAMPLE_SENTENCE>` and `<ENGLISH_EXAMPLE_SENTENCE>` in the JA→EN template correspond to the source and target sides of the example shown in the rightmost column of Table 9.

B.3 Pro-JE

Pro-JE is a human-preference set of 32 professional JA→EN anime subtitle items, separate from Subtitle-100. Each item contains four candidates: the official human subtitle and outputs from three LLMs (Qwen3-14B-Instruct (Yang et al., 2025), Gemma3-12B-IT (Team et al., 2025), and Llama3.1-8B-Instruct (Grattafiori et al., 2024)).[3] Human preferences are collected using Best–Worst Scaling (BWS). We use Pro-JE for a case study (§6.3) of how SDP dimensions and SFS relate to human preference, not to argue that source-style preservation is generally the preferred outcome.

Candidate generation for Pro-JE. The three LLM candidates for each Pro-JE item were generated by prompting each model to produce a plain, natural English translation of the Japanese source, with no style guidance, so that any stylistic shift reflects the model's default translation behavior rather than an imposed target style. The prompt instructed the model to output only the English translation of the given anime subtitle line. Generation used the chat template of each model with sampling (temperature 0.3, up to 256 new tokens) in bfloat16, and the first non empty output line was taken as the translation after removing surrounding quotes and any reasoning trace markers. This contrasts with Style25 (Appendix B.2), where generation is explicitly style conditioned; in Pro-JE the LLM outputs are unconditioned baselines against which the official human subtitle is compared.

Table 10 summarizes the three datasets used in the experiments that follow.

C Scorer Validation Details

C.1 Annotation Guidelines

All human annotations were produced by in-house bilingual Japanese–English annotators following a shared written guideline. This section summarizes the instructions given to annotators; the per-axis scale descriptions and bilingual anchor examples used during annotation are those in Appendix A (Table 5 and Table 6).

[3] For one Pro-JE item (`pair_089`), Gemma3-12B-IT and Llama3.1-8B-Instruct produced an identical English translation; the duplicate was removed during BWS survey construction, leaving only 3 distinct candidates (Human, Qwen3-14B-Instruct, and the shared Gemma3-12B-IT/Llama3.1-8B-Instruct translation) for this item instead of 4. The resulting 127 (not 128) candidates are used in Appendix F's correlation and quality-gate analyses.

Task and unit of judgment. For each item, annotators independently read the source (Japanese) and target (English) utterances and scored each utterance on all seven SDP axes using the five-point bipolar scale (-2 to $+2$). Japanese and English were scored independently, so that translation-induced style shifts could later be recovered as ΔSDP. Annotators were instructed that there is no correct or incorrect answer and to record their intuitive impression as a number.

Masked, text-only judgment. Speaker identity and surrounding context were hidden. Even when an annotator could infer which character produced a line, they were instructed to ignore character background and judge only the linguistic expression of the given text. For axes whose everyday interpretation depends on the interlocutor or situation, annotators were explicitly told to set aside such context and to score the flat, text-level impression. For example, politeness and authority were judged without assuming a particular speaker–listener relationship, emotion without assuming intimacy from the topic, age from lexical and stylistic markers rather than topic, confidence from epistemic and stylistic markers rather than factual correctness, and directness from the interpretive cost of the utterance rather than from an elaborated imagined context.

Per-axis decision procedure. Each axis was accompanied by a short, stepwise rubric that first removed confounding context, then checked concrete linguistic markers, and finally integrated them into a single score. Representative markers included honorific and register forms for politeness, aggressive or softening expressions for roughness, interjections and emphatic forms for emotion, directive versus deferential framing for authority, age-graded vocabulary and style for age, hedging versus assertive markers for confidence, and lexical difficulty and structural or rhetorical complexity for directness. Annotators were told to assign ± 2 when a property was clearly present and ± 1 when it was mild or when they were uncertain, and to use 0 for neutral text.

Independence and comments. Japanese and English scores were not required to match: divergence between them is the signal of interest. Annotators could optionally leave a free-text comment noting cues they relied on or cases where translation changed the impression, which we use to inform later development. Each annotator worked on an

	Subtitle-100	Style25	Pro-JE
Purpose	scorer validation	automatic style-sensitivity test	supportive human preference analysis
Direction	JA→EN	JA↔EN	JA→EN BWS items
Texts / candidates scored	200 utterances: source and target for each pair	up to 200 × 51 candidates before filtering; fewer after filtering	up to 128 candidates: 32 items × 4 translations (127 scored; one item has only 3, see Appendix B.3)
Judgment units	700 pair-axis judgments: 100 pairs × 7 axes	200 Pres@1 decisions: one per seed item	32 item-level BWS preferences
Human annotation	seven-axis ratings by 3 annotators	none; automatic scoring only	BWS by 4 annotators
Main use	validate the LLM-based SDP scorer	test whether metrics identify the no-perturbation anchor among deliberately style-perturbed candidates	examine whether SDP dimensions relate to human preference

Table 10: Overview of the three datasets. We distinguish items/questions, candidate texts, and judgment units because the datasets differ in structure: Subtitle-100 contains paired source–target utterances, Style25 contains many controlled perturbation candidates per seed item, and Pro-JE contains four translation candidates per item.

independent port, and judgments were saved per item.

C.2 Human Annotation Reliability

For scorer validation, three annotators independently rated Subtitle-100 on all seven SDP axes using a five-point bipolar scale. Annotators were masked to the condition of each item. Because each translation pair requires judgments for both the source and target utterances across seven axes, the annotation cost is substantial: annotating the 100 Subtitle-100 items took 36.5 annotator-hours in total.

Table 11 reports inter-annotator agreement using Krippendorff's α for each axis and language. Agreement varies by axis. Authority is the most reliable axis, whereas age and directness are among the weakest. The mean Krippendorff's α across the seven axes is 0.611 for Japanese and 0.492 for English, with an overall mean of 0.551 across both languages.

This level of agreement is comparable to prior work on pragmatic and social meaning annotation. For example, Pei and Jurgens (2020) reported Krippendorff's $\alpha = 0.548$ over 212 annotated tuples and interpreted it as moderate agreement. Our validation should be interpreted relative to this human agreement level, especially for axes with low inter-annotator agreement.

Axis	α (JA)	α (EN)
Authority	.730	.591
Age	.629	.347
Politeness	.621	.577
Emotion	.616	.530
Confidence	.599	.587
Roughness	.591	.410
Directness	.491	.400

Table 11: Inter-annotator agreement by SDP axis and language. Japanese shows higher agreement than English on all axes. The largest drops for English are observed on age, roughness, and directness.

C.3 Axis-Level LLM Scorer Validation

We further examine the selected Qwen3-14B-Instruct scorer at the axis level. Table 12 reports Spearman's ρ and Kendall's τ for each axis, split by source language (JA) and target language (EN), following the same per-language breakdown used for inter-annotator agreement in Table 11. Agreement is strongest for emotion, confidence, and authority, which are all significant in both languages, whereas age shows the weakest agreement with English-side correlation near zero and directness shows comparatively lower values. This pattern diverges from the human inter-annotator pattern in Table 11, where authority shows the highest agreement and directness the lowest: the LLM scorer tracks human judgments well on emotion and confidence despite those axes involving more subjective distinctions. This suggests that LLM scorer errors are not simply inherited from human-annotation difficulty, but partly reflect axis-specific scorer biases (see the zero-bias analysis in Appendix C.7).

Weak scorer agreement, however, does not necessarily mean that an axis is uninformative. We retain all seven axes because they are part of the predefined SDP schema. The Pro-JE axis-subset

Axis	MAE↓[†]	Spearman ρ		Kendall τ	
		JA	EN	JA	EN
Politeness	.540	.47***	.54***	.43***	.51***
Roughness	.340	.55***	.42***	.52***	.40***
Emotion	.440	.62***	.58***	.57***	.55***
Authority	.395	.47***	.62***	.44***	.59***
Age	.545	.26**	.00	.24**	.00
Confidence	.555	.58***	.59***	.53***	.53***
Directness	.755	.39***	.46***	.34***	.42***

Table 12: Axis-level validation of the selected Qwen3-14B-Instruct scorer against human mean annotations, split by source language (JA) and target language (EN). *$p < .05$, **$p < .01$, ***$p < .001$ (uncorrected for multiple comparisons). [†]MAE is pooled across JA and EN.

Model	±1 Agr.	MAE	Mean ρ
Qwen3-14B-Instruct	**93.9%**	**0.51**	**0.47**
GPT-oss-120B	88.4%	0.69	0.37
Gemma3-9B	87.5%	0.67	0.39

Table 13: LLM scorer comparison. Figures are identical to Table 2 (same 100-item Subtitle-100 set, pair_001–100) and are repeated here for convenience alongside the axis-level breakdown for the selected scorer in Table 12. Mean ρ is the mean per-axis Spearman correlation with human ratings. Qwen3-14B-Instruct achieves the highest ±1 agreement, the lowest MAE, and the highest mean ρ, and is therefore selected as the scorer used throughout the paper.

ablation in Appendix C.9 shows that no small axis subset dominates across objectives (Top-1 selection vs. Kendall's τ), and that scorer agreement alone does not reliably predict which axes carry preference-relevant signal.

C.4 LLM Scorer Comparison

We compared three LLM scorers against human annotations (Table 13). Qwen3-14B-Instruct achieved the highest ±1 agreement, the lowest MAE, and the highest mean per-axis Spearman correlation, and was therefore selected as the scorer used throughout the paper; the axis-level breakdown for this scorer is reported separately in Table 12.

C.5 LLM Scorer Prompts

This section gives the exact prompts for the three LLM-based scorers used in the paper: the SDP scorer (seven axes), our reference-free GEMBA-DA adaptation, and the SA-3 style baseline (RASTA axes).

SDP scorer. Each axis is scored by a separate LLM call (14 calls per item: 7 axes × 2 languages).

Prompt template:

```
Evaluate the {axis_name} of this
{language} sentence on a scale from
-2 to +2.

Sentence: {text}

Scale:
-2: {scale_-2_description}
-1: {scale_-1_description}
0: {scale_0_description}
1: {scale_+1_description}
2: {scale_+2_description}

IMPORTANT: Respond with ONLY a single
number (-2, -1, 0, 1, or 2). Do
not use any tags, explanations, or
thinking process.

Score:
```

Generation parameters:

- Model: Qwen3-14B-Instruct
- Max new tokens: 50
- Temperature: 0.1
- Output format: Label only (no explanation)

GEMBA-DA (reference-free adaptation). As noted in §4, our GEMBA-DA scores follow the Direct Assessment formulation of Kocmi and Federmann (2023) but omit the human-reference field, matching our other reference-free metric (COMETKiwi). Each candidate is scored by a single LLM call.

```
Score the following translation
from {src_lang} to {tgt_lang} on
a continuous scale from 0 to 100,
where:

- 0 means completely wrong,
incomprehensible, or full of errors
- 100 means perfect translation, no
errors, completely natural

Source ({src_lang}): {source}
Translation ({tgt_lang}):
{translation}

Consider accuracy, fluency,
terminology, and style. Provide
only a numeric score.

Score:
```

The returned 0–100 value is normalized to $[0, 1]$. Generation uses Qwen3-14B-Instruct with greedy decoding (max 16 new tokens); the heavy discretization of these greedy scores is analyzed in Appendix D.

Strategy	Zero%	ρ	MAE	Att.%	Rev.%
P0 Baseline	45.7	**.509**	**.600**	32.8	**9.4**
P1 Anti-zero	46.0	.493	.600	32.8	10.9
P2 Few-shot	55.3	.455	.660	42.4	11.0
P3 CoT	39.0	.501	.670	26.6	13.9
P4 Two-stage	41.7	.504	.646	28.1	12.0
P5 Combined	58.7	.431	.638	46.7	10.2

Table 14: Prompt calibration results. No strategy improves all metrics over the baseline. Strategies that reduce zero-rate increase polarity reversals.

SA-3 (RASTA style baseline). SA-3 reuses the three RASTA axes (Havaldar et al., 2025): politeness, intimacy, and formality, each scored on a five-point $[0, 1]$ scale $(0, 0.25, 0.5, 0.75, 1.0)$ by a separate LLM call, mirroring the SDP scorer structure so that the two style measures differ only in their axis inventory.

```
Evaluate the {axis_name} of this
{language} sentence.

{axis_description}

Sentence: {text}

Scale (0 to 1):
0.0: {scale_0.0_description}
0.25: {scale_0.25_description}
0.5: {scale_0.5_description}
0.75: {scale_0.75_description}
1.0: {scale_1.0_description}

IMPORTANT: Respond with ONLY a
single number (0.0, 0.25, 0.5,
0.75, or 1.0). Do not use any tags,
explanations, or thinking process.

Score:
```

Generation uses Qwen3-14B-Instruct (max 50 new tokens, temperature 0.1). SA-3 fidelity is computed from these axis scores using the same MAD-based distance as SFS (Appendix C.8), restricted to the three RASTA axes.

C.6 Prompt Calibration Experiment

Five alternative prompt strategies were tested to determine whether LLM axis attenuation can be mitigated through prompt engineering ($N = 100$ items $\times$ 7 axes $\times$ 2 languages). Results are shown in Table 14.

No strategy improves all metrics over the baseline. Strategies that reduce zero-rate increase polarity reversals. Few-shot examples *increase* zero-rate (55–59%) as anchors function as comparison standards. Baseline achieves highest overall ρ (.509) and lowest reversal rate (9.4%).

Axis	$N_{\neq 0}$	Correct	Atten.	Reversal
politeness	118	53.4%	36.4%	10.2%
roughness	84	40.5%	47.6%	11.9%
emotion	118	72.0%	24.6%	3.4%
authority	74	64.9%	27.0%	8.1%
age	79	27.8%	65.8%	6.3%
confidence	139	61.9%	28.8%	9.4%
directness	146	68.5%	17.1%	14.4%
Total	758	57.8%	32.8%	9.4%

Table 15: Per-axis LLM error decomposition (full 100 items, JA+EN combined). Age has the highest attenuation (65.8%); emotion has the lowest reversal (3.4%). The LLM's primary failure mode is attenuation, not polarity reversal.

Function	d (H)	d (L)	r (H↔L)
MAD_{sim}	**1.01**	**0.84**	**.478**
Norm. Manhattan	1.01	0.84	.478
Norm. Euclidean	1.09	0.71	.435
RMS_{sim}	1.09	0.71	.435
Pearson r	0.78	0.07	.365
Cosine	0.63	0.17	.336
Dot product	-0.23	-0.31	.506
Max deviation	0.85	0.59	.387
Agreement (± 1)	0.72	0.47	.301
Agreement (± 0)	0.41	0.22	.186

Table 16: Distance function comparison. d (H) = Cohen's d for human scoring; d (L) = Cohen's d for LLM scoring; r (H↔L) = correlation between human and LLM composite scores. MAD_{sim} achieves the best joint performance.

C.7 Per-Axis LLM Error Decomposition

LLM scoring errors are decomposed by axis ($N = 100$ items $\times$ 7 axes $\times$ 2 languages) in Table 15. On non-zero human ratings, the LLM preserves polarity in 58% of cases, attenuates to 0 in 33%, and reverses polarity in 9%.

The LLM shows overall zero bias: it outputs 0 on 57% of all ratings versus 46% for humans, an 11-point gap most severe on age (91% vs. 55%) and politeness (54% vs. 35%).

C.8 SFS Distance Function Comparison

Ten distance functions for computing SFS were compared, evaluating each on condition separation (Cohen's d) and Human–LLM agreement (Pearson r). Results are reported in Table 16.

MAD_{sim} achieves the best joint performance. Cosine similarity is invariant to uniform scaling, so it cannot detect "style flattening" where both vectors are proportionally attenuated. Using MAD instead of cosine improved d from 0.63 to 1.01 and r from .336 to .478.

MAD and normalized Manhattan are algebraically identical: $\text{MAD}_{\text{sim}} = 1 - \frac{1}{K}\sum_{i=1}^{K}\frac{|s_i - t_i|}{R}$.

C.9 Axis-Subset Robustness

To test whether the seven-axis SFS is redundant, we exhaustively evaluate all $2^7 - 1 = 127$ non-empty axis subsets on Pro-JE against two objectives: Top-1 selection accuracy and Kendall's τ with human BWS preference. The two objectives favor different axes. The subset that maximizes Top-1 accuracy at k=5 (politeness, roughness, emotion, authority, age) reaches 56.2% but only τ=+0.041, whereas the best subset for τ alone at k=3 (emotion, confidence, directness) reaches τ=+0.195 (above the seven-axis SFS's τ=+0.147), yet is not the Top-1-best subset (k=7: 46.9% Top-1). No subset dominates both objectives simultaneously, and *authority* never appears among the best subsets for either objective, while *emotion* appears in nearly all of them. This shows that human–LLM scorer agreement (Appendix C.3) is not a reliable guide to which axes matter for preference: directness has comparatively strong scorer agreement yet contributes little to Top-1 accuracy, while age has weaker agreement but appears in the best Top-1 subset. We therefore retain the full seven-axis SFS as a fixed, task-agnostic inventory: pruning to any single small subset would optimize one objective at the expense of another, and axis informativeness is objective-dependent rather than reducible to a single ranking.

D Statistical Details

Inter-annotator agreement heatmap. Figure 3 shows Krippendorff's α per axis. This page intentionally includes additional figures for reference.

Condition separation and LLM error profile. Figure 4a shows the SFS distribution by condition (Official vs. Modified) for human and LLM scoring on the full N=100 Subtitle-100 set; both scoring sources clearly separate the two conditions ($d = 0.66$ for $\text{SFS}_{\text{Human}}$, $d = 0.30$ for SFS_{LLM}). Figure 4b shows the per-axis LLM error decomposition (left) and per-axis Human–LLM Spearman ρ (right, N=100 items, JA+EN pooled): age has the highest attenuation rate (65.8%, Table 15) and the weakest axis-level Human–LLM correlation (Table 12), whereas emotion has the lowest reversal rate (3.4%) and among the strongest correlations.

Distance function comparison. Figure 5 visualizes the tradeoff between condition separation and Human–LLM agreement for 10 distance functions.

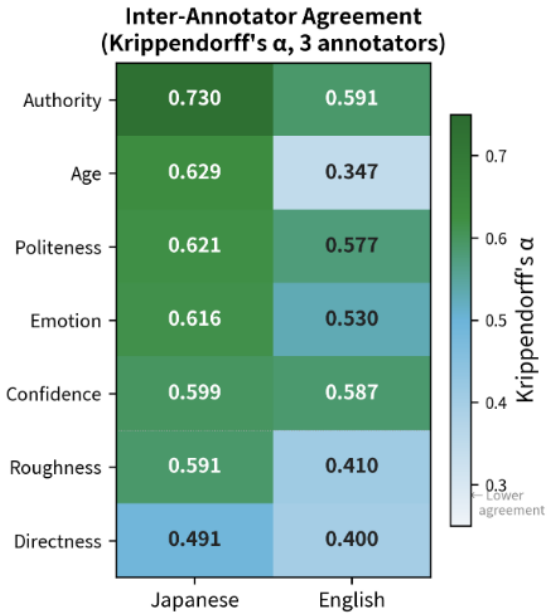
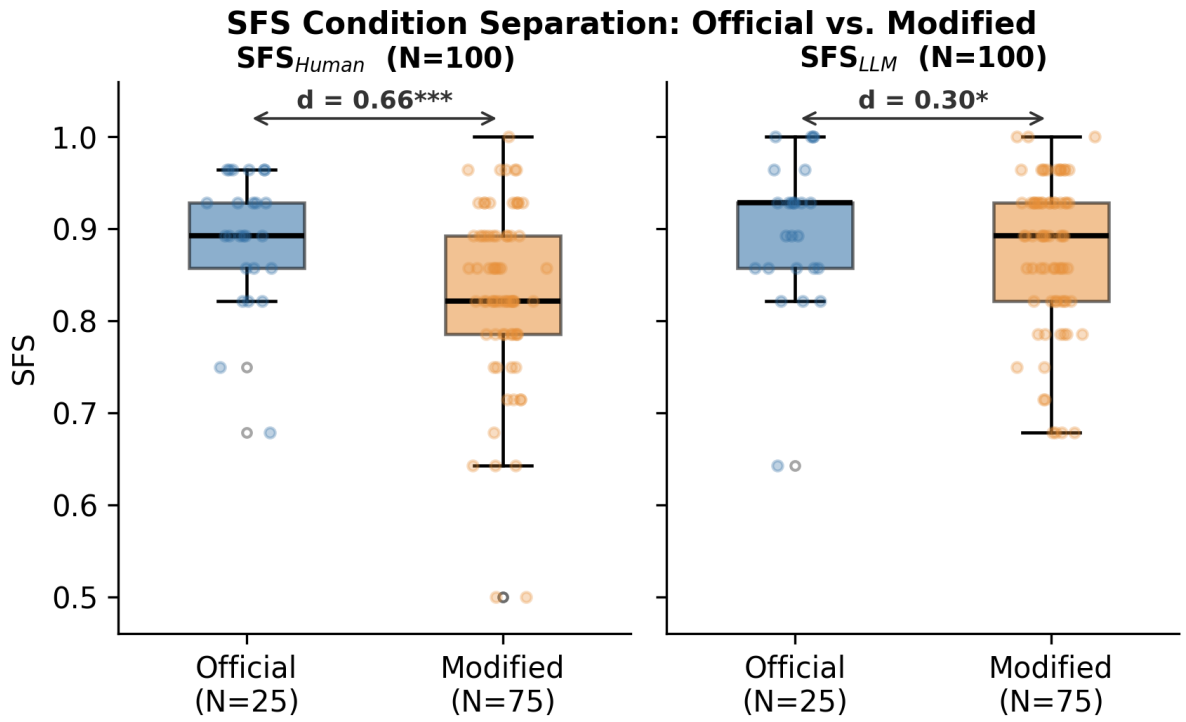


Figure 3: Inter-annotator agreement (Krippendorff's α) per axis.

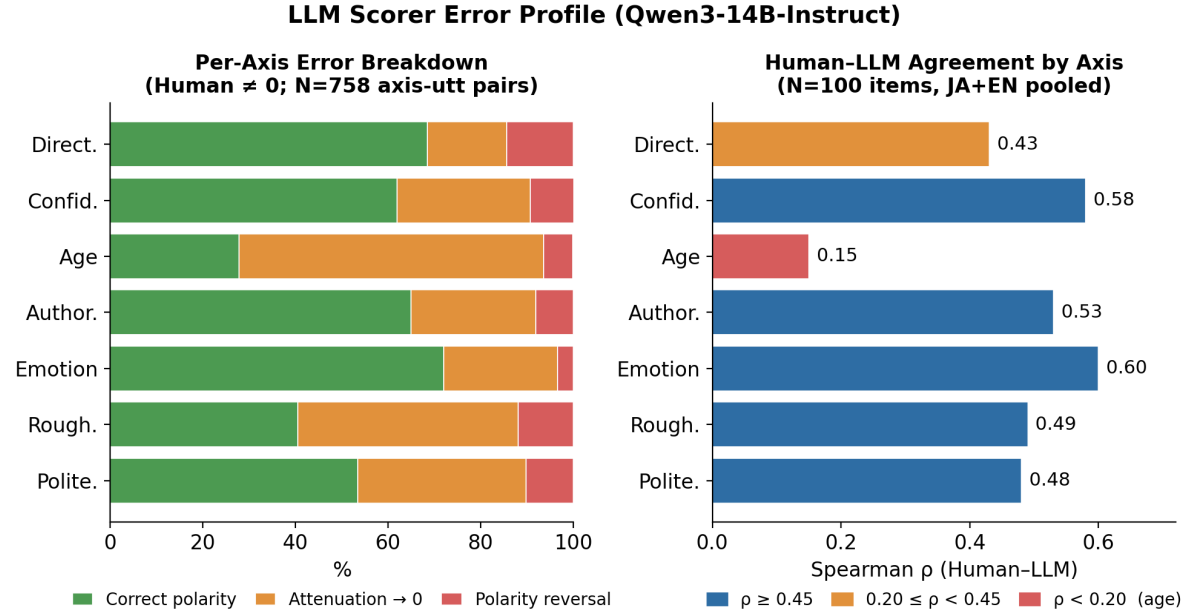
Corpus invariance test. Bootstrap subsampling test (B = 1,000, drawing N = 15 items from 100 JA→EN pairs) illustrates utterance-level design: SFS shows SD = 0.013 (sampling noise only), whereas corpus-level Pearson r fluctuates with SD = 0.12, ranging from 0.08 to 0.75.

GEMBA-DA tie-handling caveat. The GEMBA-DA scores used here come from our reference-free adaptation (§1): an LLM judge is prompted for a single 0–100 score under greedy decoding. In practice the outputs collapse onto a handful of discrete values (6–10 unique values per direction; e.g. 66.8% of JA→EN scores equal exactly 0.95), producing very high anchor–candidate tie rates (54.4% in JA→EN, 33.1% in EN→JA). Such score compression under greedy or low-temperature decoding is a known property of LLM-based DA metrics rather than an artifact of our prompt (Freitag et al., 2024).

Because the anchor is always the first entry in each item's candidate ordering, Pres@1 in Table 3 silently breaks top ties in its favor. Table 17 recomputes Pres@1 under three tie-handling rules: under strict no-tie ranking GEMBA-DA falls to or below chance in both directions, and under fair split-credit it stays below chance in EN→JA while remaining modestly above chance in JA→EN. Thus tie handling explains most, but not all, of GEMBA-



(a) SFS distribution by condition (Official vs. Modified) for human scoring (left) and LLM scoring (right), N=100 items. Both scoring sources clearly separate the two conditions (SFS$_{\text{Human}}$: $d = 0.66$; SFS$_{\text{LLM}}$: $d = 0.30$), confirming that SFS reliably distinguishes perturbed from unperturbed translations regardless of the scoring source.



(b) LLM error profile by axis (left: per-axis error decomposition when Human ≠ 0; right: per-axis Human–LLM Spearman ρ, N=100 items, JA+EN pooled). Age has the highest attenuation rate (65.8%) and the weakest Human–LLM correlation (ρ=0.15); emotion has the lowest reversal rate (3.4%) and among the strongest correlations (ρ=0.60). The dominant error mode is attenuation toward the neutral label rather than polarity reversal.

DA's apparent anchor recovery.

GEMBA-DA's own scores do not support an adequacy-reduction explanation: an item-level comparison of the anchor against its perturbed candidates' mean shows no significant difference in either direction (paired t-test $p = 0.71/0.77$) and no direction asymmetry ($p = 0.79$, Mann–Whitney U). This behavior is expected to generalize: be-

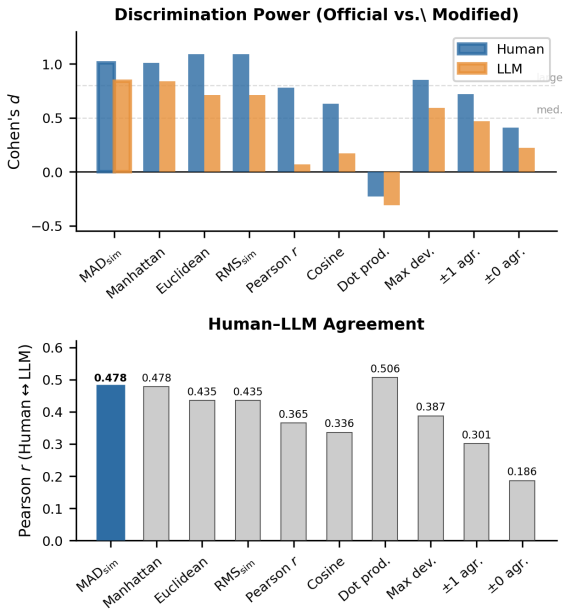


Figure 5: Distance function comparison. MAD_{sim} achieves the best joint performance: the highest discrimination power between Official and Modified conditions (Cohen's d, top) and strong Human–LLM agreement (Pearson r, bottom).

GEMBA-DA tie handling	JA→EN	EN→JA	Type
Standard (Table 3)	74.0	36.0	quality
Strict no-tie	2.0	0.0	quality
Split-credit ties	8.6	2.3	quality

Table 17: GEMBA-DA Pres@1 (%) on Style25 under three tie-handling rules. *Standard* reproduces the ranking used for Table 3, where ties with the anchor are silently broken in its favor; *Strict no-tie* credits the anchor only when its score is the sole maximum; *Split-credit* divides credit evenly among tied top-scoring candidates. Chance levels are 5.7% (JA→EN) and 3.7% (EN→JA). Both corrections fall at or below chance in EN→JA; under split-credit the JA→EN figure remains modestly above chance, so tie handling explains most, but not all, of GEMBA-DA's apparent anchor recovery.

cause the perturbed Style25 candidates preserve meaning and grammaticality (they pass the same COMETKiwi≥ 0.80 gate as the anchor), adequacy-oriented scoring, including reference-based DA and GEMBA-MQM, is by design largely insensitive to style-only changes and would plausibly show comparable compression under greedy decoding.

E Additional Style25 Diagnostics

E.1 Style Shift Magnitude by Category and Intensity

Figure 6 measures the magnitude of the induced shifts. Strong perturbations produce larger mean absolute ΔSDP than mild ones, indicating that ΔSDP is sensitive to the intensity as well as the direction of controlled style changes. The unmodified anchor has the smallest average shift here, a diagnostic property of Style25, not a claim that minimal shift is preferable in professional translation.

E.2 Style Fidelity by Categories

Figure 7 shows mean SFS by category and direction. Style-shift magnitude differs substantially by perturbation type. Lower SFS means a larger shift from the source-style profile, not lower translation quality or human preference.

E.3 Style Adaptation Profile

Figure 8 compares signed ΔSDP profiles across unmodified translations and controlled Style25 perturbations for both translation directions, complementing the professional adaptation analysis in §6.2. The main text finding that professional JA→EN subtitles raise politeness ($+0.24$) and neutralize roughness ($+0.17$) should be read in the context of anime dialogue rather than as evidence of overformalization. Japanese is often described as a language with rich politeness marking, yet many anime utterances involve close social relations, conflict, or strong character voice, and therefore use reduced politeness or highly intimate forms. When such utterances are rendered into idiomatic English subtitles, they frequently map onto a more standard interpersonal register, which appears as a small positive politeness shift on our scale simply because the target expression is closer to a neutral social distance. A parallel effect holds for roughness: very gentle or soft source-side expressions (note the negative source-side roughness, -0.45 in Table 4) tend to be rendered in English with a less marked, more neutral tone, producing a positive roughness shift toward 0 rather than toward crudeness.

These two shifts are therefore not contradictory: both reflect movement toward a standard English interpersonal baseline rather than a uniform increase in formality or aggression. This is consistent with Figure 8, where the unmodified Pro-JE (Human) and Style25 Original rows show smaller and more structured shifts than the controlled LLM-

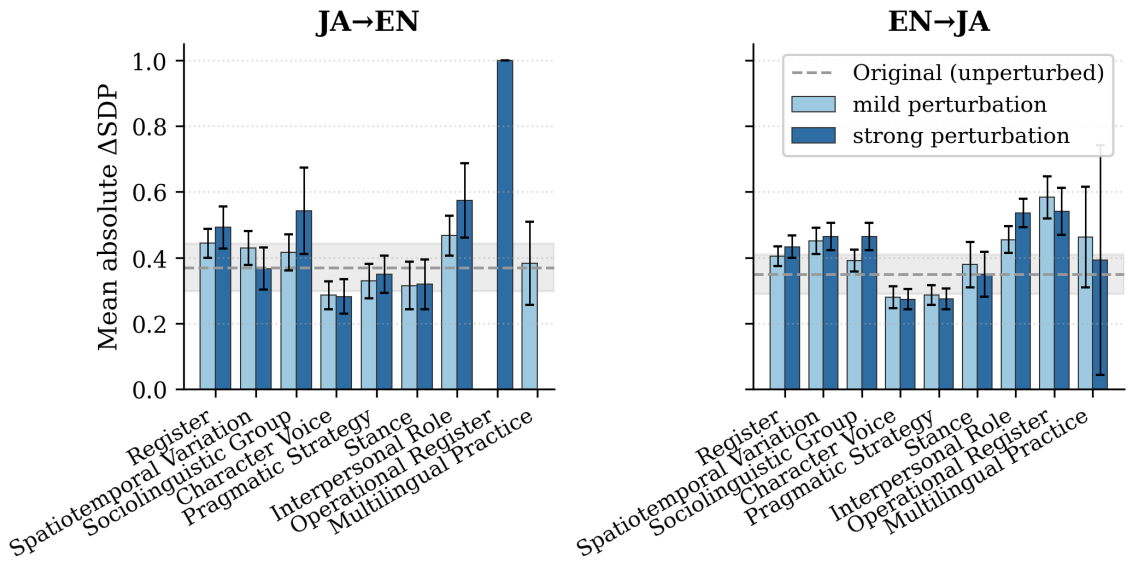


Figure 6: **Style shift magnitude by category and intensity (Style25).** Mean absolute ΔSDP ($\frac{1}{7}\sum_d |\Delta\text{SDP}_d|$) per category, split into mild/strong variants, with the unmodified OpenSubtitles anchor. Strong perturbations generally induce larger shifts than mild ones. Single-style categories, such as community-register and code-switching, are based on fewer samples and should be read with caution.

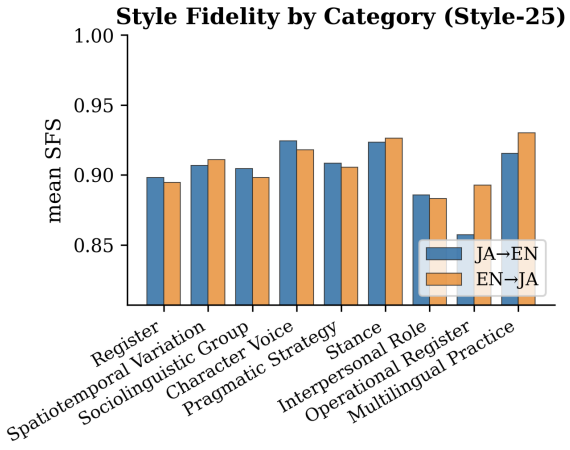


Figure 7: **Style fidelity by category (Style25).** Mean SFS by category and direction. Variation across categories indicates that controlled perturbations induce different magnitudes of style shift. Lower SFS indicates a larger shift from the source-style profile, but does not imply lower translation quality or lower human preference.

perturbation rows. Taken together, the professional sample preserves emotion and directness while normalizing selected interpersonal register cues for the target language, supporting the view that these changes are localization-oriented adaptation rather than style-copying failure.

Category	Dir.	SFS	SA-3	Δ
dialect	JA→EN	78.1	69.9	**+8.2**
locale	JA→EN	60.5	51.2	**+9.3**
character-voice	JA→EN	56.8	50.0	**+6.8**
stance	JA→EN	64.0	73.0	−9.0
register	EN→JA	36.4	51.5	−15.2
speech-style	EN→JA	55.8	42.1	**+13.7**

Table 18: SFS vs. SA-3 anchor recovery by category on Style25 (Pres@1, %). Positive Δ indicates higher Pres@1 for SFS. These six labels are 6 of the 9 categories in the Style25 taxonomy of Table 9, each itself a coarser grouping of the 25 individual Style25 styles; only the categories with the largest $|\Delta|$ are shown here. Correspondence to Table 9: *dialect* = Sociolinguistic Group (Business, Elderly, Gen-Z); *locale* = Spatiotemporal Variation (Regional-1, Regional-2, Era); *character-voice* = Character Voice (Deadpan, Bitter, Brooding); *speech-style* = Pragmatic Strategy (Blunt, Roundabout, Hedging); *stance* = Stance (Assertive); *register* = Register (Textbook, Academic, Literary, Tabloid, Casual, SNS, Report). Interpersonal Role, Operational Register, and Multilingual Practice are not shown here because they did not rank among the largest $|\Delta|$ categories.

E.4 SFS vs. SA-3 by Category

Table 18 breaks down anchor-recovery performance for SFS and the SA-3 baseline by category, restricted to the categories with the largest $|\Delta|$.

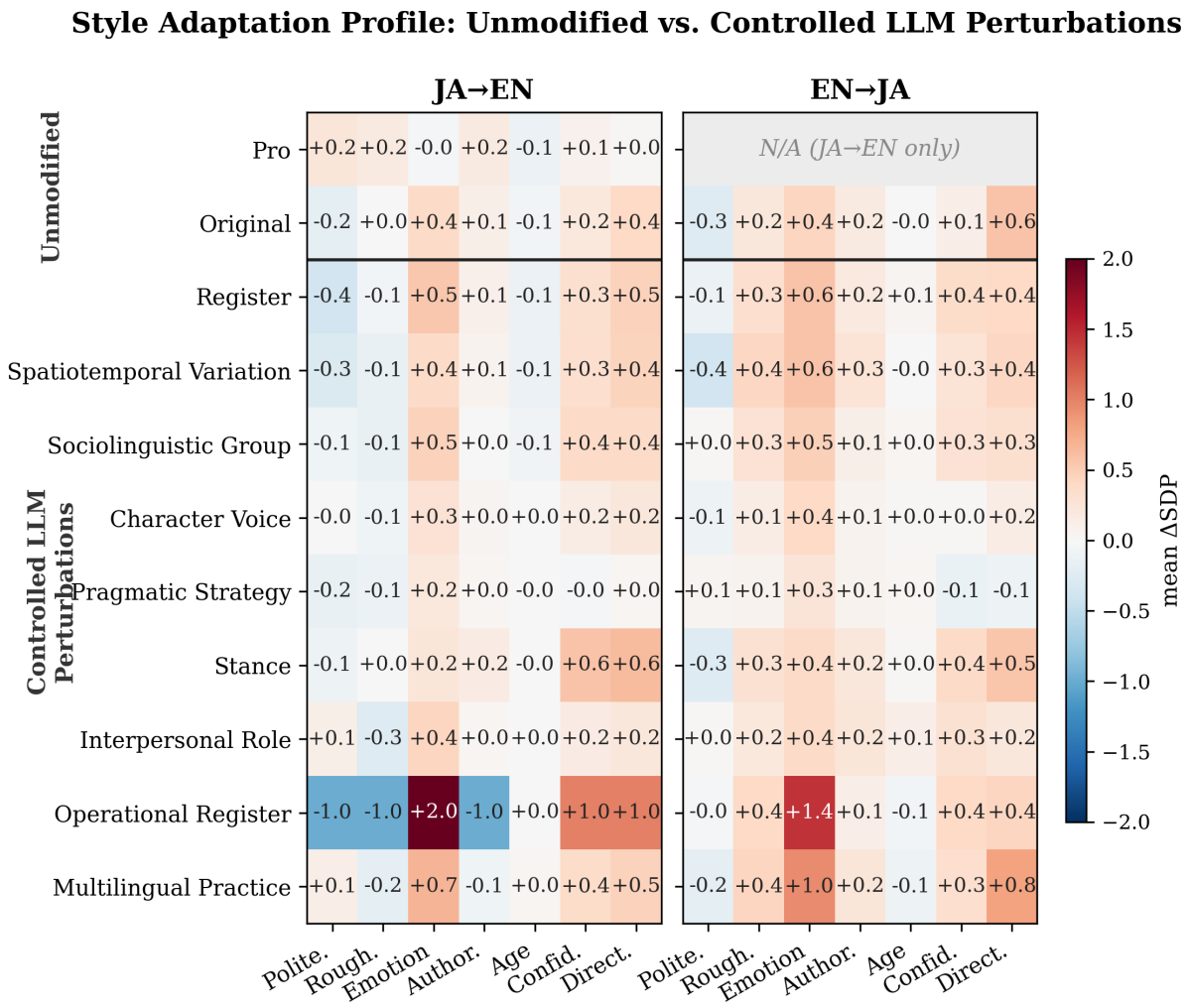


Figure 8: **Style adaptation profile.** Mean signed ΔSDP (translation minus source) for JA→EN (left) and EN→JA (right) on a shared colour scale. Rows are grouped into *Unmodified* translations (Pro-JE professional Human, available for JA→EN only, and Style25 Original) and *Controlled LLM Perturbations* (Style25 categories). The unmodified rows show smaller and more structured shifts than the controlled perturbation rows, while LLM perturbations induce larger changes especially in emotion, confidence, and directness. This supports the view that style changes should be analyzed as signed adaptation profiles rather than treated simply as copying or failure to copy. The community-register condition is an operational stress test and does not represent a real community.

F Full Per-Metric Correlations and Quality-Gate Analysis on Pro-JE

This appendix reports the full per-metric correlations and quality-gate analysis on Pro-JE referenced in §6.3. All results use Pro-JE only (32 items; up to 128 candidates) and are reported as case-study observations (§4.3). Significance markers follow $^{\dagger}p < .10$, $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$ (two-sided).

Top-1 accuracy with fractional credit. For Top-1 accuracy, metric ties receive fractional credit: if the metric's top set $\mathcal{M}_{\text{top}}$ contains n tied candidates, the credit is

$$\frac{|\mathcal{M}_{\text{top}} \cap \mathcal{B}_{\text{top}}|}{n}, \tag{4}$$

where $\mathcal{B}_{\text{top}}$ is the set of BWS-best candidates, and BWS ties are treated as an equally best set. This is the Top-1 definition used both in the main text (§4.3) and throughout this appendix.

F.1 Per-Metric Correlation with Human BWS

Table 19 reports, for each candidate metric, Kendall's τ against human BWS scores and Top-1 accuracy (the fraction of items for which the metric's top candidate matches the BWS-best candidate). Chance-level Top-1 accuracy is 0.250 (one of four candidates per item).

Metric	Type	τ	Top-1	Note
COMETKiwi	Adequacy	-0.008	0.438	Uncorrelated with BWS on Pro-JE
MetricX-24	Quality	$+0.038$	0.469	Weak overall
GEMBA-DA	Quality	$+0.160^{*}$	0.380	Strongest τ among all metrics; ties on 84% of items lower its Top-1 (see note below)
SFS (7-axis)	Style	$+0.147^{*}$	0.404	Style signal, largely independent of quality
SA-3 (RASTA axes)	Style	$+0.008$	0.307	Weak on Pro-JE

Table 19: Per-metric correlation with human BWS preference on Pro-JE. Chance-level Top-1 accuracy is 0.250. Metric ties receive fractional credit (credit $= |\mathcal{M}_{\text{top}} \cap \mathcal{B}_{\text{top}}|/n$, where n is the number of tied top candidates); BWS ties are treated as an equally-best set (§4.3).

GEMBA-DA attains the strongest rank correlation with human preference on Pro-JE ($\tau = +0.160^{*}$) but a comparatively low Top-1 accuracy (0.380): on 84% of Pro-JE items, GEMBA-DA assigns all four candidates the same score (0.95), so its top-ranked set is frequently a four-way tie. Under fractional-credit scoring this depresses Top-1 accuracy even though the metric's ordinal correlation with BWS is the strongest observed. This illustrates that τ and Top-1 accuracy capture different aspects of alignment with human preference and should be interpreted jointly rather than in isolation. COMETKiwi, an adequacy-oriented metric, is uncorrelated with BWS on Pro-JE ($\tau \approx -0.008$), consistent with the claim in §6.3 that style fidelity (SFS) provides preference-relevant signal that is complementary to, rather than redundant with, quality-oriented metrics. Applying Holm–Bonferroni correction across these five tests ($m = 5$) raises the two nominally significant p-values (GEMBA-DA $p = .030$; SFS $p = .032$) above the corrected threshold ($p_{\text{Holm}} = .148$ for both), so neither survives family-wise correction; we report this openly rather than treat the uncorrected markers as conclusive.

F.2 Quality-Gate Sensitivity

Table 20 shows how the SFS–BWS and SA-3–BWS associations change when candidates are restricted to those passing increasingly strict quality thresholds (COMETKiwi, MetricX-24, and/or GEMBA-DA), starting from all 127 scored Pro-JE candidates.

Quality gating consistently strengthens, and never reverses, the SFS–BWS association: SFS τ ranges from $+0.116$ to $+0.225^{*}$ across gates, compared to $+0.147^{*}$ unfiltered, and SFS Top-1 accuracy rises from 0.404 (unfiltered) to as high as 0.656 under the combined COMET+GEMBA gate. In contrast, SA-3 remains uncorrelated or weakly negatively correlated with BWS under every gate (τ between -0.041 and $+0.008$). This pattern supports interpreting semantic adequacy and style fidelity as jointly necessary, rather than substitutable, signals for human preference on Pro-JE: once low-adequacy candidates are removed, the residual variance in human preference is better explained by style fidelity (SFS) than by the narrower RASTA-derived axis set (SA-3). Treating all twelve SFS/SA-3 tests across the six gate conditions as one family, Holm–Bonferroni correction ($m = 12$) leaves none significant at $\alpha = .05$ (smallest raw $p = .032$ against a corrected threshold of .0042); the gate-wise pattern above is therefore best read as a consistent descriptive trend rather than a family-wise-significant effect.

F.3 Axis-Level Association with Human Preference

Table 21 and Table 22 report Kendall's τ between each individual style axis and human BWS scores on Pro-JE, for the SDP seven-axis set and the SA-3 baseline, respectively.

On Pro-JE, no single SA-3 axis reaches significance, and among the seven SDP axes only emotion is significant ($\tau = +0.153^{*}$), with confidence showing a marginal positive trend ($\tau = +0.139^{\dagger}$). This axis-level breakdown is consistent with §6.2's finding that professional translators leave emotion and directness nearly unchanged on average: the present result further suggests that, when translations do vary in emotional expression, human raters on Pro-JE prefer the more emotionally expressive candidate.

F.4 Confound Check Between Axis Values and Quality Gate

Because §F.2 restricts candidates by quality-metric thresholds, an axis–preference association could

Quality gate	Pass N	SFS τ	SFS Top-1	SA-3 τ	SA-3 Top-1
(none)	127	$+0.147^{*}$	0.404	$+0.008$	0.307
COMET ≥ 0.8	62	$+0.161$	0.575	-0.010	0.383
MetricX ≥ 0.895	63	$+0.116$	0.533	$+0.006$	0.479
GEMBA ≥ 0.950	95	$+0.143^{\dagger}$	0.512	-0.019	0.366
COMET + MetricX	50	$+0.129$	0.559	-0.041	0.422
COMET + GEMBA	51	$+\mathbf{0.225}^{*}$	**0.656**	-0.028	0.354

Table 20: Effect of quality gating on the SFS–BWS and SA-3–BWS associations on Pro-JE. Pass N is the number of Pro-JE candidates retained after applying the quality threshold(s). Bold marks configurations whose SFS association exceeds the unfiltered (none) baseline on both τ and Top-1.

SDP axis	τ	Note
Politeness	-0.051	n.s.
Roughness	$+0.002$	n.s.
Emotion	$+0.153^{*}$	Strongest single-axis association on Pro-JE; more emotional candidates are preferred
Authority	$+0.041$	n.s.
Age	-0.120	Negative trend, n.s.
Confidence	$+0.139^{\dagger}$	Weak positive trend (marginal)
Directness	$+0.108$	n.s.

Table 21: Axis-level Kendall's τ between SDP axes and human BWS on Pro-JE.

SA-3 axis	τ	Note
Politeness	$+0.071$	n.s.
Intimacy	$+0.004$	n.s.
Formality	$+0.032$	n.s.

Table 22: Axis-level Kendall's τ between SA-3 axes and human BWS on Pro-JE.

Family	Axis	η^2
SA-3	Formality	0.000
SA-3	Intimacy	0.011
SA-3	Politeness	0.009
SDP	Politeness	0.035
SDP	Emotion	0.025
SDP	Directness	0.044
SDP	Age	0.016
SDP	Roughness	0.003
SDP	Authority	0.009

Table 23: Confound check: η^2 between each style axis and quality-gate pass/fail status on Pro-JE. All values are small (≤ 0.044), indicating no substantial confound between axis values and the quality gate in the Pro-JE candidate set.

in principle be inflated if axis values happen to correlate with the pass/fail quality-gate status itself. Table 23 reports η^2 between each axis and the (COMET + GEMBA) quality-gate pass/fail indicator on Pro-JE.

All η^2 values on Pro-JE are small (≤ 0.044), indicating that, unlike the Style25-derived controlled perturbation sets, the Pro-JE candidate pool (official subtitle plus three LLM outputs) does not exhibit a systematic relationship between axis values and quality-gate status. Pro-JE is therefore an appropriate baseline for interpreting the axis-level and quality-gate associations reported above without confound-driven inflation.

F.5 Pooled versus Within-Group Decomposition

As a further robustness check, Table 24 decomposes the pooled Kendall's τ between each axis and BWS into a within-group component (computed after conditioning on the COMET+GEMBA quality-gate group) and a between-group inflation component (pooled τ minus within-group τ), for the axes with the largest pooled associations on Pro-JE. Within τ and Residual τ are two distinct within-group estimators rather than the same quantity under different names: Within τ is the size-weighted average of Kendall's τ computed separately inside each quality-gate group, whereas Residual τ is a single pooled τ computed after group-mean-centering both the axis value and BWS within each group. The two can diverge because Kendall's τ is not linear, so averaging per-group correlations is not equivalent to correlating pooled, group-centered residuals.

On Pro-JE, most of the pooled association for SA-3 formality and SDP emotion survives within-group conditioning (residual τ of $+0.053$ and $+0.065$, respectively), whereas SDP directness shows a larger inflation component ($+0.105$) relative to its within-group residual ($+0.020$), suggesting that its pooled association is driven more by between-group differences than by within-group rank agreement. SDP

Axis	Pooled τ	Within τ	Inflation	Residual τ
SA-3 Formality	$+0.085$	$+0.073$	$+0.012$	$+0.053$
SDP Emotion	$+0.123$	$+0.068$	$+0.055$	$+0.065$
SDP Directness	$+0.125^{\dagger}$	$+0.020$	$+0.105$	$+0.026$
SDP Confidence	$+0.105$	$+0.136$	-0.031	$+0.110^{\dagger}$

Table 24: Pooled vs. within-group (COMET+GEMBA quality-gate group) Kendall's τ on Pro-JE. Inflation is the pooled-minus-within difference, i.e., the portion of the pooled association attributable to between-group mean differences rather than within-group rank agreement.

confidence shows negative inflation, i.e., its within-group association ($+0.136$) is if anything stronger than the pooled association. Together with the confound check in §F.4, these decompositions indicate that Pro-JE's candidate design does not introduce the kind of confound structure observed in the Style25-derived comparison sets, supporting its use as the primary basis for the case-study human-preference claims in §6.3.

G Direction-Specific Style Planes for Style25

To complement the pooled Style25 visualization in Figure 2, we plot the same interpersonal–expressive style plane separately for JA→EN and EN→JA. The two composite axes are defined as in §6.1: *interpersonal force* $=$ mean(Δdir, Δaut, $-\Delta$pol) and *expressive intensity* $=$ mean(Δemo, Δrou, Δcon).

As shown in Figure 9, Style25 categories occupy different regions in both directions, supporting the pooled analysis. The unmodified anchor is not exactly at the origin in either direction, indicating that naturally occurring translations may already introduce cross-lingual shifts. These plots serve only as diagnostic evidence that ΔSDP captures systematic style-shift patterns, not as quality or preference judgments.

H Data-Driven Projection of Style25 Category Shifts

The two-dimensional view in Figure 2 and Figure 9 uses a hand-chosen projection (§6.1). To check that the observed category separation is not an artifact of this specific projection, we also apply principal component analysis (PCA) to the full seven-axis ΔSDP vectors of all Style25 variants, pooling both translation directions.

Figure 10 shows the result. The first two components explain 75.5% and 13.2% of the variance (88.7% cumulative), indicating that most category-level variation in the seven-axis space is captured by a two-dimensional structure. The point is that the same categories that are separated under the hand-chosen projection remain separated under this data-driven projection, indicating that the separation in Figure 2 is not only for the particular axis combination we chose for visualization. The full seven-axis ΔSDP representation can be examined through other axis pairs or combinations depending on the stylistic contrast of interest, and the PCA plane used here is only one such view that happens to capture most of the variance (PC1 + PC2 $=$ 88.7%).

The figure also plots the individual base-style means (light points), making within-category and cross-style variation directly visible. To make this within-category structure easier to read, Figure 11 re-plots the same base-style means on an identical PCA scale, split into nine per-category panels. As this shows, categories differ in internal spread: multi-style categories such as Register (n=7), Pragmatic Strategy (n=3) and Interpersonal Role (n=3) contain styles that spread widely, sometimes in opposing directions, whereas single-style categories (Stance, Community-register perturbation, and Multilingual Practice, each n=1) are represented by a single point and have no within-category variation by construction. We take this to suggest that our categories are a convenient grouping, and that styles placed in the same category may respond along different axes.

Two aspects of the seven-axis representation deserve comment; the categories themselves are used only as a convenient grouping for display. First, the separation in Figure 2 is not an artifact of our hand-chosen projection. Under a data-driven PCA projection, the categories still separate from the unmodified anchor and from one another, lying away from the origin rather than collapsing toward it, so the category-level means do not result from large internal cancellation. Second, a low-dimensional view, whether hand-chosen or data-driven, is a reading aid rather than the representation our diagnostics operate on, and the high variance captured by PC1 does not imply that fewer axes suffice. PCA maximizes explained variance, not diagnostic interpretability: a principal component mixes axes and cannot by itself indicate *which* stylistic property changed, whereas SDP attributes a shift to named, independently meaningful axes (as in the per-axis Pro-JE analysis of §6.2). The within-category spread in Figure 11 further shows that

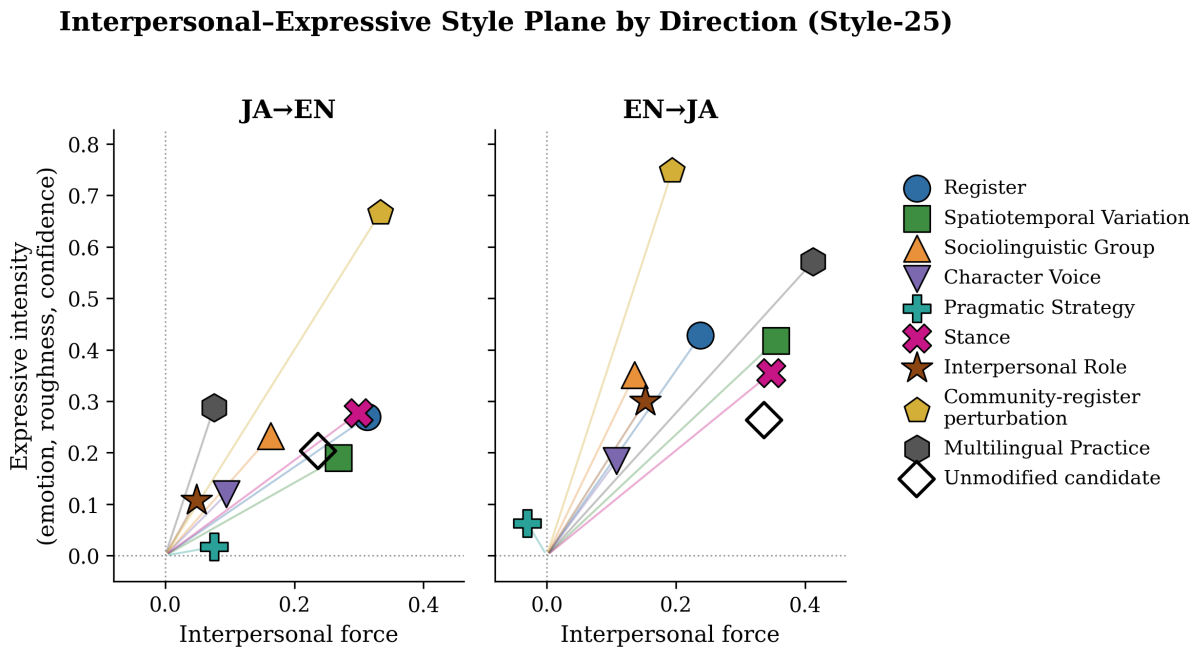


Figure 9: **Interpersonal–expressive style plane by direction (Style25).** Each point is a Style25 category placed by its mean signed ΔSDP separately for JA→EN and EN→JA. The white diamond is the unmodified OpenSubtitles anchor. As in the pooled view (Figure 2), categories occupy distinct regions in both directions, confirming that the category-specific pattern is not an artifact of pooling across translation directions.

styles sharing similar PC1 and PC2 coordinates can differ substantially on the remaining axes, so individual styles are not fully recovered by any two dimensions. We therefore report the full seven-axis SDP profile and treat Figure 2 and Figure 10 as visual summaries only.

I Case Studies

Table 25 illustrates how ΔSDP diagnoses human preference at the item level. Here COMETKiwi scores Qwen3-14B-Instruct highest, but human BWS strongly prefers the official subtitle. The official subtitle is not a style-identical copy; it slightly reduces roughness and directness, yet it preserves the source's emotional force and informal character voice through its hesitation and expressive phrasing. The LLM outputs instead neutralize the utterance by reducing roughness, increasing confidence, and sometimes reducing directness or emotion. ΔSDP thus identifies which stylistic dimensions shift, explaining preference patterns that scalar quality scores miss.

Beyond this single item, Table 26 reports candidate-level aggregates over all $N = 32$ Pro-JE items. The Human (Official) translation attains the highest group-mean BWS ($+0.358$) and is chosen best most often (50.8%), while Llama3.1-8B is the least preferred (-0.345) and most often chosen worst (44.8%). The same axes that distinguish the candidates in the item-level example also separate them in aggregate: the Human translation shows the smallest mean emotion shift ($|\Delta\text{emo}| = 0.22$), whereas Llama shows the largest mean directness shift ($|\Delta\text{dir}| = 1.07$). This pattern is consistent with the interpretation in §6.3: preferred translations tend to preserve the source's emotional force, while less preferred ones drift most on directness. We report these as descriptive case-study observations, not confirmatory findings.

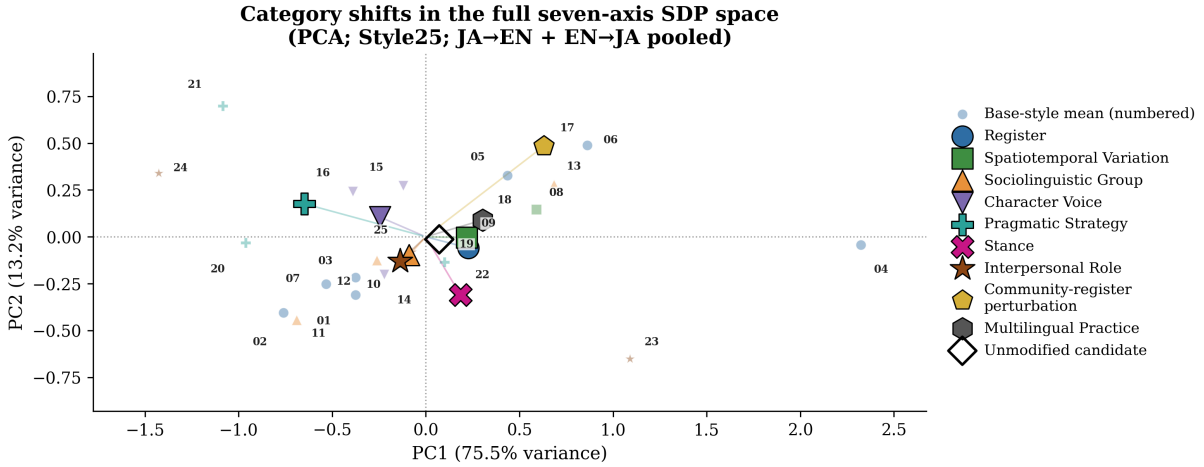


Figure 10: **Category shifts in the full seven-axis SDP space (PCA; Style25, JA↔EN pooled).** PCA on the seven-axis ΔSDP vectors. Large markers are the nine category means; light points are individual base-style means, showing within-category variation. Numbers (01–25) correspond to the style IDs in Table 9. The white diamond is the unmodified OpenSubtitles anchor. The first two components explain 75.5% and 13.2% of the variance.

Candidate	Translation	BWS	SFS	COMET	Major ΔSDP
Human	W-wow... Talk about a supersized ambition...	1.00	0.93	0.79	Rou −1, Dir −1
Gemma3-12B	That's an incredibly ambitious goal.	−0.75	0.89	0.68	Pol +1, Rou −1, Conf +1
Qwen3-14B-Instruct	Wow, the scale of the ambition is huge.	0.00	0.86	0.83	Pol +1, Rou −1, Conf +1, Dir −1
Llama3.1-8B	That's a pretty huge ambition.	−0.25	0.82	0.76	Pol +1, Rou −1, Emo −1, Conf +1, Dir −1

Table 25: **Case study.** ΔSDP diagnoses item-level differences between human preference and scalar quality scores. COMETKiwi ranks Qwen3-14B-Instruct highest, whereas human BWS prefers the official subtitle. The preferred translation is not a perfect style copy, but better preserves the source's emotional and informal character voice. Source (JA):すげえ 野望のサイズがデカい(pair_070).

Candidate	BWS Score	Best (%)	Worst (%)	mean |Δemo|	mean |Δdir|
Human (Official)	**+0.358**	50.8	15.0	**0.22**	0.63
Qwen3-14B	+0.050	22.5	17.5	0.41	0.72
Gemma3-12B	−0.075	16.7	24.2	0.41	0.78
Llama3.1-8B	−0.345	10.3	44.8	0.42	**1.07**

Table 26: Candidate-level human preference (BWS) and mean absolute style shift on Pro-JE ($N = 32$). The most-preferred Human translation shows the smallest emotion shift, while the least-preferred Llama shows the largest directness shift.

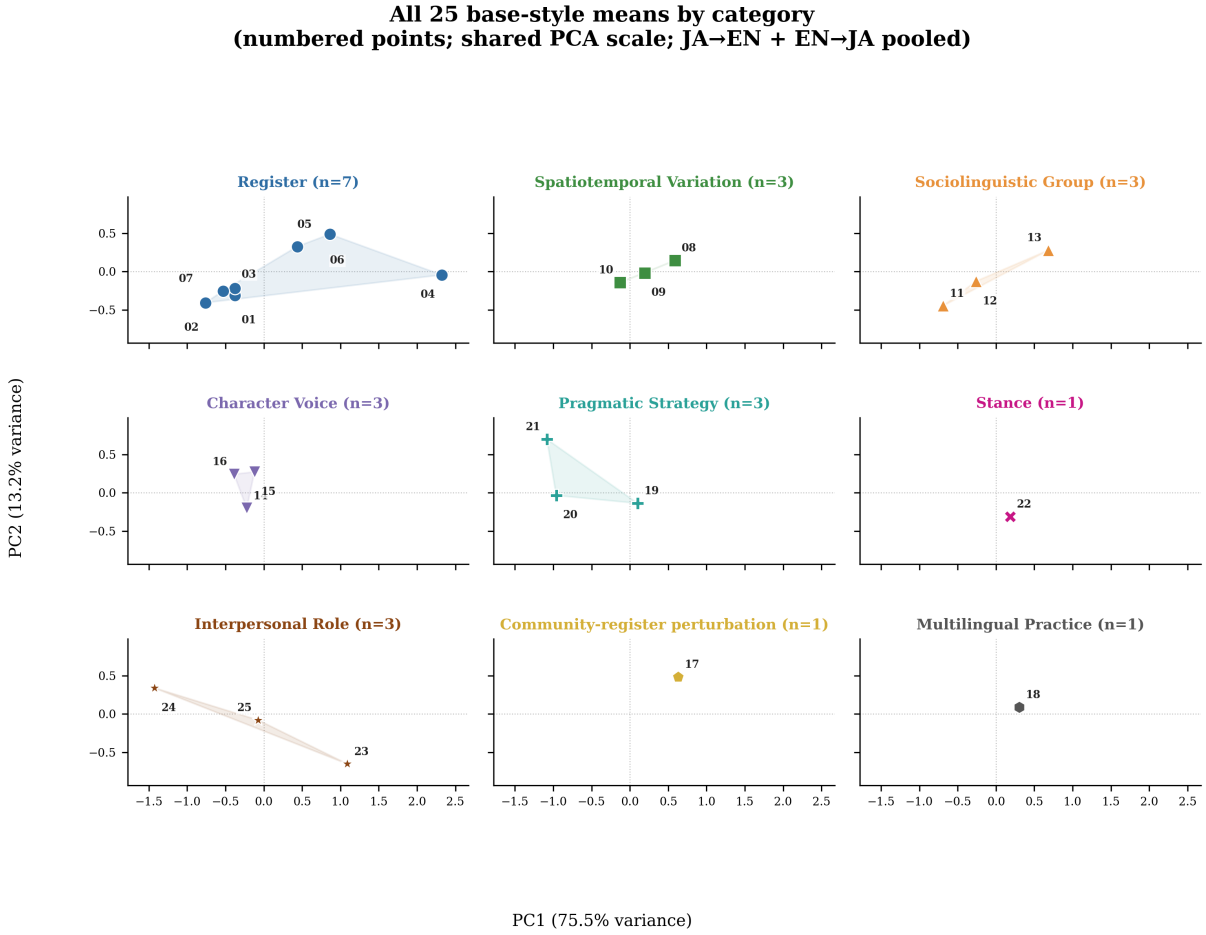


Figure 11: **Per-category view of the 25 base-style means in the full seven-axis SDP space (PCA; Style25, JA↔EN pooled).** The same base-style means as in Figure 10, plotted on an identical PCA scale but split into nine per-category panels. Numbers (01–25) correspond to the style IDs in Table 9. Shaded hulls indicate the spread within each category, making within-category and cross-style variation directly visible; no hull is drawn for categories with a single style ($n = 1$)

HexaBee-4B: Six-Language Translation with Multilingual Answering and Bidirectional Supervision

Iman Barati[1]* **Mohammadmostafa Rostamkhani**[1] **Mohammad Badzohreh**[2]

[1]Iran University of Science and Technology, Tehran, Iran

[2]Tarbiat Modares University, Tehran, Iran

iman_barati@comp.iust.ac.ir mo_rostamkhani97@comp.iust.ac.ir

mohammad.badzohreh@modares.ac.ir

Abstract

Multilingual translation requires a model to preserve meaning while producing text in a requested language. We present HEXABEE-4B, a translator adapted from Gemma 3 4B IT through three stages: Wikipedia continued pretraining, joint multilingual answering and answer-text translation, and bidirectional translation specialization. The intermediate stage connects two tasks through shared multilingual answers: answering a question in the requested language and translating an answer into that language. It comprises 19.46 million answering records, including 5.76 million cross-language question–answer pairs, and 10.70 million answer-text translations. The final stage uses 658,274 forward and reverse synthetic translation examples. On 36,572 synthetic test examples spanning all 30 directions among Arabic, Chinese, English, Hebrew, Persian, and Spanish, HEXABEE-4B achieves direction-macro chrF++ of 73.75, compared with 57.11 for Gemma 3 4B IT and 66.83 for Gemma 3 12B IT. BLEU, TER, COMET, and a matched model-judged subset support the same overall ranking. These results demonstrate improved translation across the six-language set, with chrF++ gains in every evaluated direction over both baselines.

1 Introduction

A multilingual language model can encounter the same information in several languages, yet translation requires more than recognizing that information: its response must preserve the source meaning and follow a specified target language. Multilingual neural translation establishes target-language conditioning as a central design choice (Johnson et al., 2017). More recent work shows that continued training and translation-oriented instruction tuning can adapt general language models into stronger translators (Xu et al., 2024; Li et al., 2024). For a compact instruction-tuned model, the organization of this supervision is therefore an important part of the translation system.

Multilingual instruction tuning supplies a broader source of supervision. Multitask training and multilingual instruction collections improve instruction following across languages (Muennighoff et al., 2023; Singh et al., 2024; Üstün et al., 2024); even limited multilingual instruction data can support cross-language generalization (Shaham et al., 2024). However, multilingual data need not require communication *between* languages. A question and its answer may always use the same language. Explicitly separating their language assignments creates a different task: understand a question in one language and answer it in another. Cross-language instruction tuning and translation-following demonstrations provide precedents for this design (Lin et al., 2025; Ranaldi et al., 2024; Zhu et al., 2024).

We investigate how to organize such answering supervision within a translation-focused adaptation recipe (Figure 1). Consider an English question whose requested answer is Spanish. The model must answer the question in Spanish, whereas an English-to-Spanish translation task must preserve the content of its English input. Our construction connects these tasks through multilingual answer texts: a Spanish answer can supervise both an English question and an English version of that answer. Figure 2 illustrates this shared-target relation. The system instruction specifies the desired response language and, where relevant, response style. Its own wording can vary across languages, but this variation is a supporting data-construction choice. Translation remains the primary task evaluated here.

This separation of language roles is also motivated by multilingual question-answering research. TyDi QA and MKQA broaden the linguistic cov-

*Corresponding author.

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 46–70
November 9, 2026 ©2026 Association for Computational Linguistics

erage of QA evaluation (Clark et al., 2020; Longpre et al., 2021). XHotpotQA makes language assignments to questions and evidence roles explicit when studying cross-lingual knowledge composition (Barati et al., 2026). Our work applies the related principle of separating language roles to training question–answer generation and answer-text translation. The resulting supervision uses a Wikipedia-derived multilingual inventory and does not require retrieving evidence at inference time.

We introduce HEXABEE-4B, built from **Gemma 3 4B IT** (Gemma Team, 2025).[1] Its three stages provide encyclopedic language modeling, multilingual response supervision, and translation specialization. The second stage combines 19,458,332 answering records with 10,695,072 answer-text translations; 5,759,806 answering records cross question and response languages. The third stage uses 658,274 bidirectional synthetic examples. We evaluate the complete model across six languages using five automatic metrics and paired model-based judgments. Our contributions are the three-stage translation recipe, an explicit account of how answering and translation share supervision, and a direction-complete evaluation against public Gemma 3 4B IT and 12B IT. The comparison measures the complete recipe; it does not isolate the effect of an individual training component.

2 Related Work

Multilingual translation and adaptation. Multilingual MT shares parameters across translation directions and uses explicit target-language signals (Johnson et al., 2017). ALMA studies translation-oriented continued training of large language models (Xu et al., 2024), while Li et al. (2024) examine multilingual finetuning with translation instructions. Our staged design combines this translation focus with an intermediate answering task. Its initial Wikipedia adaptation follows the broader motivation of continued pretraining on relevant text (Gururangan et al., 2020). Training on teacher-generated translations is related to sequence-level knowledge distillation (Kim and Rush, 2016).

Multilingual instruction supervision. Multitask finetuning enables cross-lingual generalization (Muennighoff et al., 2023), and Aya provides multilingual instruction data and models (Singh et al., 2024; Üstün et al., 2024). CrossIn combines cross-language instruction–response pairs with instruction translation (Lin et al., 2025). CrossAlpaca uses instruction-following and translation-following demonstrations (Ranaldi et al., 2024); Question Translation Training aligns reasoning questions across languages (Zhu et al., 2024). These studies establish relevant precedents for coupling instruction following and translation. Our contribution is the particular combination of system-directed answering, translations of answer texts, and subsequent bidirectional specialization on Gemma 3 4B IT, together with its measured corpus composition and translation results.

Language roles in question answering. TyDi QA collects information-seeking questions directly in diverse languages (Clark et al., 2020), while MKQA supports comparison across languages through aligned questions and language-independent answer representations (Longpre et al., 2021). XHotpotQA studies mixed-language evidence in multi-hop QA by assigning languages to distinct roles in an evidence-dependency graph (Barati et al., 2026). This provides a conceptual motivation for making language boundaries explicit rather than treating multilinguality only as a collection of monolingual examples. Here, the relevant roles are the question, requested answer, and instruction wording; we use their combinations as generation supervision for a translator. Our experiments concern translation quality, whereas those benchmarks evaluate answering or knowledge composition.

3 Method

3.1 Answering and translation as connected tasks

Let $\mathcal{L} = \{\mathrm{ar}, \mathrm{zh}, \mathrm{en}, \mathrm{he}, \mathrm{fa}, \mathrm{es}\}$ be the six translation languages. We use $s, t \in \mathcal{L}$ for input and response languages. For a content item c, let $q_s(c)$ denote its question in language s, and let $a_t^v(c)$ denote an answer in language t with response variant v (extent or style). Let S denote the *system instruction text*. Specifically, $S_{t,v}^{\mathrm{QA}}$ requests an answer in language t with variant v, and $S_{s,t}^{\mathrm{MT}}$ requests translation from s into t.

[1] Resources: Code; Final model.

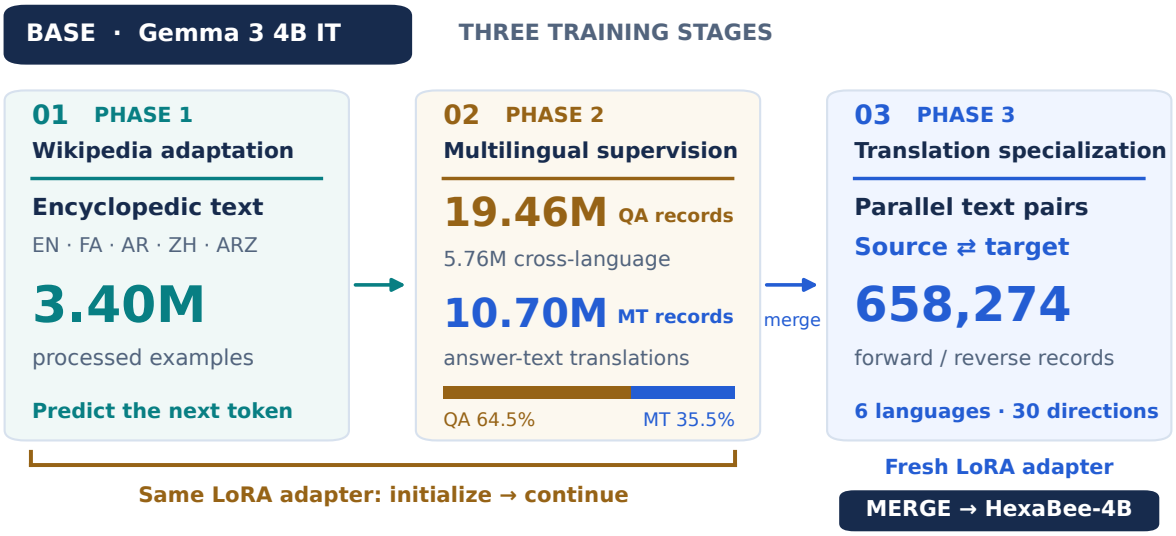


Figure 1: The complete adaptation recipe. Phase 1 applies next-token language modeling to Wikipedia text. Phase 2 continues the same adapter with multilingual answering and answer-text translation. After merging, phase 3 trains a fresh adapter on bidirectional translations and produces the final merged HexaBee-4B model. The base is Gemma 3 4B IT. Phase 1 counts processed training examples; phases 2–3 count SFT records. Examples derived from one source can occur in multiple tasks.

The two supervision relations are

$$\text{QA}: \quad (S^{\text{QA}}_{t,v}, q_s(c)) \mapsto a^v_t(c), \tag{1}$$

$$\text{MT}: \quad (S^{\text{MT}}_{s,t}, a_s(c)) \mapsto a_t(c), \tag{2}$$

where $a_s(c)$ and $a_t(c)$ are the answer variants selected for translation. QA includes same-language and cross-language pairs; MT uses different input and output languages. A shared answer can therefore appear as the target of both tasks. Its input is a question in QA and an answer text in MT. This relation connects response generation to semantic transfer without adding an intermediate generation step.

Instruction pools provide alternative phrasings for response language, extent, style, and translation direction.[2] Cross-language QA explicitly requests the answer language; some same-language examples request only answer extent. A directive's wording language and its requested response language are separate choices. We preserve this distinction when constructing the data, rather than assuming that a prompt written in one language always requests that language. Pool composition and sampling are documented in Appendix D.

3.2 A shared response objective

For either task, let x be the input text and $y = (y_1, \ldots, y_m)$ the target response of m tokens. With

[2]Resources: System prompts.

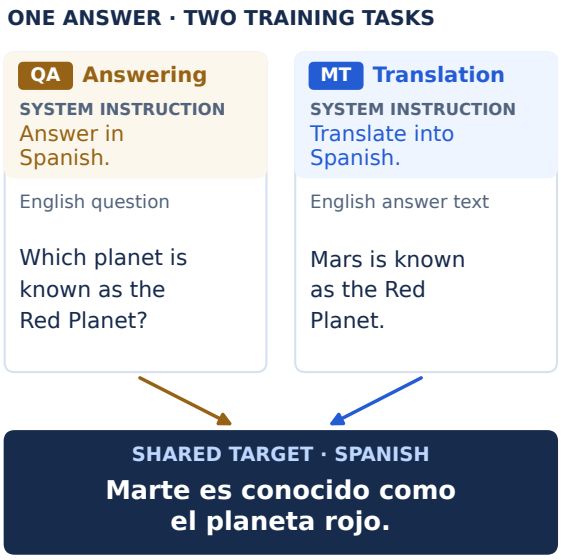


Figure 2: Illustrative English-to-Spanish supervision. The tasks have different inputs and instructions but can share the same target answer. This constructed example explains the method; it is not a quoted training record.

model parameters θ, the response-token negative log-likelihood is

$$\ell_\theta(S, x, y) = -\sum_{k=1}^{m} \log P_\theta(y_k \mid S, x, y_{<k}), \tag{3}$$

where $y_{<k}$ is the response prefix preceding token k. The instruction and input are condition-

ing text; the response supplies supervision. QA and translation examples use this same objective with their respective inputs and targets. Let $\mathcal{D}_{\mathrm{QA}}$ and $\mathcal{D}_{\mathrm{MT}}$ denote the retained QA and translation records. The phase-2 dataset is the multiset union $\mathcal{D}_2 = \mathcal{D}_{\mathrm{QA}} \uplus \mathcal{D}_{\mathrm{MT}}$, so repeated or reused examples retain their contribution to the training mixture. We do not introduce a separate alignment loss between the tasks.

4 Three-Stage Training Data and Adaptation

4.1 Phase 1: Wikipedia continued pretraining

The first stage adapts Gemma 3 4B IT to encyclopedic text before introducing the paired generation tasks. It uses Wikipedia text in English, Persian, Arabic, Chinese, and Egyptian Arabic. These are the languages registered for this stage; the later six-language translation set additionally includes Hebrew and Spanish. The corpus is monolingual text, so this phase provides language-modeling supervision rather than question–answer or translation pairs.

For a tokenized sequence $w = (w_1, \ldots, w_n)$, its autoregressive objective is

$$\ell_\theta^{\mathrm{PT}}(w) = -\sum_{k=2}^{n} \log P_\theta(w_k \mid w_{<k}), \qquad (4)$$

where n is sequence length and $w_{<k}$ contains preceding tokens. In contrast to Equation 3, supervision is applied throughout the text sequence, without a separate requested response. This stage aims to adapt vocabulary use and encyclopedic text modeling to the material used in the next stage, consistent with the motivation of continued pretraining (Gururangan et al., 2020).

A LoRA adapter (Hu et al., 2022) is trained for one epoch on 3,399,664 processed training examples. The available raw source files contain 11,395,061 text rows before preprocessing; these counts have different units. We retain the trained adapter for phase 2.[3] Appendix A gives source-language counts and the repeated-source composition of the original mixture.

4.2 Phase 2: multilingual answering and answer translation

The second stage constructs multilingual question and answer variants from 2,025,730 Wikipedia-derived source rows. The source inventory begins with Persian content and is expanded into the other languages using Gemma 3 27B IT (Gemma Team, 2025). Figure 3 traces this inventory into the two supervision branches. The resulting multilingual inventory, rather than a single language, defines the supervision in Equations 1–2. Construction provenance and translation paths are given in Appendix B.

Phase-2 supervision	Records
Same-language answering	13,698,526
Cross-language answering	5,759,806
Answer-text translation	10,695,072
Total	30,153,404

Table 1: Retained phase-2 supervision. Same/cross refers to question and answer language. Response-style and direction breakdowns are given in the appendix.

For answering, the input is a question and the target is an answer selected in the requested language. The variants include complete and moderate answers, as well as formal and informal responses. For answer-text translation, both sides are answer variants associated with the same content. All constructed answer-translation groups participate in this phase alongside QA; translation is therefore a substantial part of the intermediate training, not only the final specialization.

The retained mixture contains 30,153,404 records[4]: 19,458,332 answering records and 10,695,072 answer-text translations (Table 1). Cross-language answering accounts for 5,759,806 records, or 29.6% of QA supervision. Answering and translation account for 64.5% and 35.5% of the full mixture, respectively. These are corpus proportions, not additional loss weights. The same source item can contribute to several tasks or language combinations.

The QA distribution is asymmetric: the complete-answer family covers 27 cross-language directions, while the combined translation data cover all 30. Retained data also contain empty fields and a small set of conflicting language directives. We document their counts and affected directions in Appendix C. The intended task is language-controlled answering; the recorded corpus is an imperfect realization of that design.

Phase 2 continues the phase-1 adapter for one epoch on the combined answering/translation mix-

[3] Resources: Wikipedia corpus; Phase-1 adapter.

[4] Resources: Phase-2 training data.

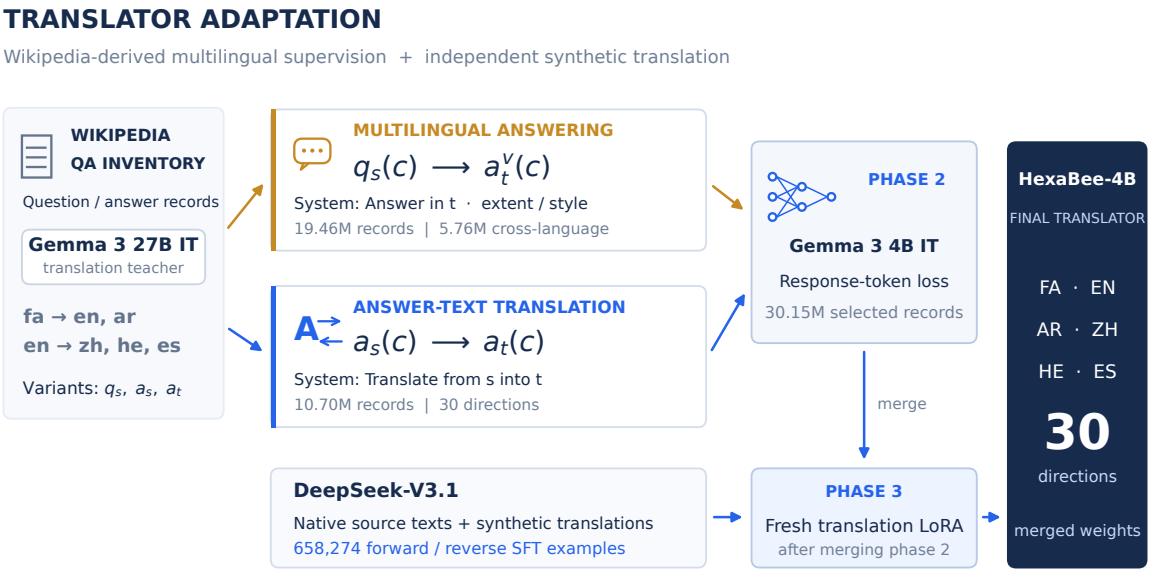


Figure 3: Data and supervision flow in phases 2 and 3. Gemma 3 27B IT expands the Wikipedia-derived inventory into multilingual question and answer variants. Answering and answer-text translation supply complementary phase-2 inputs, with system instructions specifying the requested response. A separate DeepSeek-generated corpus supplies the final bidirectional translation stage. The full checkpoint sequence, including phase 1, is shown in Figure 1.

ture. The resulting adapter is then merged with Gemma 3 4B IT to obtain the multilingual checkpoint used by the final stage.[5] This continuity allows both phases to contribute to the checkpoint before translation specialization.

4.3 Phase 3: bidirectional translation specialization

The third stage uses a separately generated translation corpus. DeepSeek-V3.1 source-generation metadata (DeepSeek-AI, 2025, 2024) identifies the teacher used to produce texts across topics, genres, registers, and complexity levels in the six languages. Each source text is paired with a synthetic translation into a different language. Native source generation broadens the content beyond the Wikipedia-derived answering inventory; its diversity is a construction goal, not evidence of coverage of every deployment domain.

The corpus contains 365,718 aligned pairs, with 60,953 source texts per language. Its stored split contains 329,146 training pairs and 36,572 test pairs. Filtering leaves 329,137 training pairs for bidirectional supervision. Each retained pair supplies two examples: translating the source into its paired target and translating the paired target back into the original source language. Thus the final SFT dataset contains $2 \times 329{,}137 = 658{,}274$ records.[6] Both orientations share an underlying pair. The reverse example reuses the existing texts; it does not require another teacher call.

The use of decoded teacher translations is related to sequence-level distillation (Kim and Rush, 2016). Here, both source generation and target translation are synthetic, so evaluation on the held-out portion measures this particular distribution. The exact split and filtering details are retained in Appendix B.

A fresh LoRA adapter is trained on the merged phase-2 checkpoint using the response objective in Equation 3. After one epoch, it is merged to produce HEXABEE-4B.[7] All three stages use LLaMA-Factory (Zheng et al., 2024); adapter parameterization, optimization settings, and serialization are in Appendix E. The complete sequence is therefore continued pretraining, joint answering/translation SFT, and translation-only SFT, with two adapter merges.

[5] Resources: Phase-2 adapter; Merged multilingual checkpoint.

[6] Resources: Bidirectional translation corpus.

[7] Resources: Phase-3 adapter.

5 Experimental Setup

5.1 Evaluation data and comparison systems

We evaluate stored outputs for all 36,572 held-out synthetic examples, covering the 30 directed language pairs. Each direction contains 1,173–1,276 examples. The comparison includes HEXABEE-4B, public Gemma 3 4B IT, and public Gemma 3 12B IT (Gemma Team, 2025); all three use the same source texts and references. A matched subset of 300 examples, ten per direction, is additionally scored by a model judge.

The archived inference templates differ between HEXABEE-4B and the public baselines. Consequently, results compare the recorded systems, including their prompts, rather than isolating adaptation under an identical prompt. We score outputs as stored, including empty responses and unsolicited text. Inference settings and failure counts are reported in Appendix B.4. No human-authored translation benchmark is evaluated in this study; FLORES provides a complementary evaluation setting for future validation (Goyal et al., 2022).

5.2 Automatic metrics

We report BLEU (Papineni et al., 2002), chrF (Popović, 2015), chrF++ (Popović, 2017), TER (Snover et al., 2006), and COMET (Rei et al., 2020). SacreBLEU provides reproducible lexical-metric settings (Post, 2018); COMET uses the fixed `wmt22-comet-da` model (Unbabel, 2022).[8] TER is lower-is-better; the other reported metrics are higher-is-better. COMET values are multiplied by 100 for display.

Let $\Omega = \{(s,t) \in \mathcal{L}^2 : s \neq t\}$ be the set of 30 directions and M_d the score of a metric computed on direction d. Our primary summary is

$$M_{\text{macro}} = \frac{1}{|\Omega|} \sum_{d \in \Omega} M_d. \tag{5}$$

This gives every direction equal weight. A target-language summary averages its five incoming directions. Pooled corpus scores are different aggregations and are reported separately in Appendix F, together with metric settings, encoder truncation, and every directional result.

5.3 Model judging and uncertainty

Model-based translation evaluation can complement reference-overlap metrics (Kocmi and Federmann, 2023b,a). We use the archived DeepSeek judge in two modes: reference-assisted, with the synthetic reference, and reference-free, with source and candidate only. Both modes produce a holistic score from 0 to 100 and eight rubric dimensions covering accuracy, terminology, fluency, cohesion, context, language adherence, style, and formatting. These are model judgments, not calibrated accuracy percentages or human annotations.

We compare systems on the same 300 examples and estimate 95% intervals for score differences using paired bootstrap resampling within each translation direction (Koehn, 2004). This preserves the matched comparison and gives each direction the same weight. The intervals capture item-sampling uncertainty on this subset; they do not quantify variability across training runs or alternative judges. Appendix B.5 gives the procedure and rubric details.

6 Results and Analysis

6.1 Overall translation quality

HEXABEE-4B obtains the best overall score on all five automatic metrics (Table 2). Direction-macro chrF++ reaches 73.75, compared with 57.11 for Gemma 3 4B IT and 66.83 for Gemma 3 12B IT. This is a gain of 16.64 points over the same-size public base and 6.92 points over the larger comparator. BLEU increases from 40.70 to 62.03 relative to the 4B baseline; TER decreases from 51.40 to 30.05, and COMET increases from 85.88 to 89.86. The agreement between lexical and learned metrics supports improved translation on the evaluated synthetic distribution.

6.2 Language and direction differences

The chrF++ gain is positive in every direction relative to both baselines. Figure 4 shows improvements for each target language, while Appendix F reports all source–target comparisons. Relative to the 4B baseline, the largest directional gain is Chinese-to-Persian (+27.09), and the smallest is English-to-Spanish (+6.64). Thus improvement extends beyond one target language or one script pairing.

Absolute chrF++ remains lower for Chinese-target translations than for the other target groups. This should not be read as a calibrated ranking of language difficulty: script, segmentation, reference phrasing, and the metric's character/word

[8] Resources: Stored model outputs; Evaluation tables and figures.

System	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.
4B IT	40.70	58.91	57.11	51.40	85.88	80.32	81.43
12B IT	50.54	68.70	66.83	38.89	88.20	91.73	91.97
HEXABEE-4B	**62.03**	**75.08**	**73.75**	**30.05**	**89.86**	**92.92**	**94.10**

Table 2: Full-test direction-macro automatic metrics ($N = 36{,}572$) and holistic judge means on 300 matched items. COMET is scaled by 100; TER is lower-is-better. Bold marks the best displayed score and underlining the second distinct score. Each system uses its recorded inference prompt.

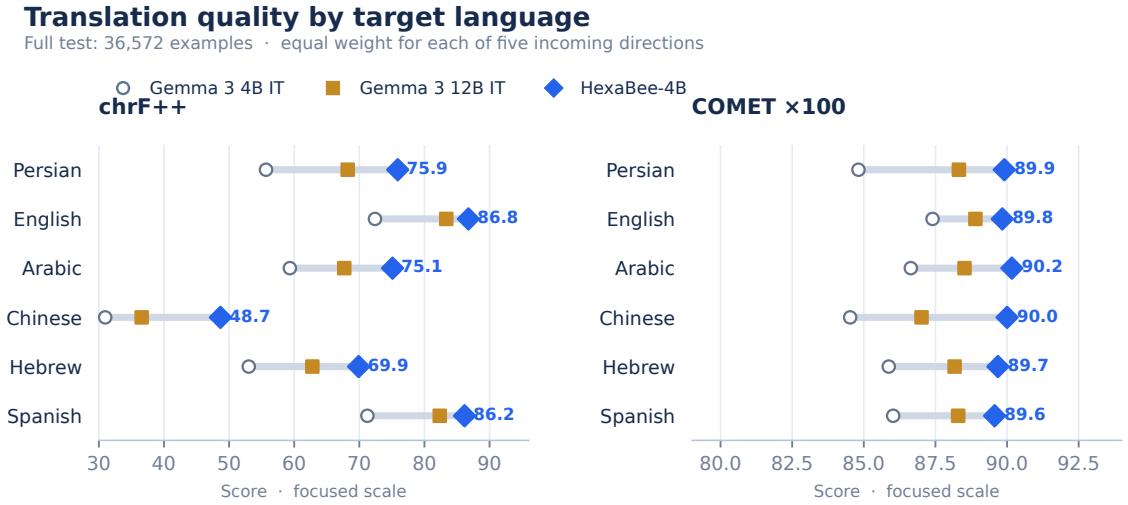


Figure 4: Translation quality by target language. Each point averages five incoming directions on the full test partition. HEXABEE-4B improves both direction-macro chrF++ and COMET for each target language. The focused axis ranges are shown explicitly; marker shapes distinguish systems as well as colors.

components affect cross-language score comparisons. The metric agreement is also not universal at the direction level. For Chinese-to-Hebrew, HEXABEE-4B has TER 90.09, worse than 88.40 for 4B IT and 78.13 for 12B IT, despite higher BLEU and chrF++. TER improves in the other 29 directions. This exception motivates inspecting multiple metrics rather than inferring uniform improvements from a single aggregate.

6.3 Judged quality

Both judge modes follow the automatic-metric ranking. Reference-assisted means are 92.92 for HEXABEE-4B, 80.32 for 4B IT, and 91.73 for 12B IT; reference-free means are 94.10, 81.43, and 91.97. Table 3 reports paired differences and uncertainty. The comparison with the larger model is considerably closer than the comparison with the 4B base.

Against 4B IT, HEXABEE-4B wins, ties, or loses on 214, 54, and 32 reference-assisted examples, respectively. Across rubric dimensions, its terminology mean increases from 75.60 to 90.42, and target-language adherence from 82.88 to 96.15. Terminology nevertheless remains its lowest reference-assisted dimension. These judgments suggest a useful focus for further bilingual error analysis; they do not establish the frequency of specific error categories. Complete rubric results by direction and target language are included in the appendix.

Baseline	Judge view	Gain	95% CI
4B IT	Reference	+12.60	[9.96,15.31]
4B IT	No reference	+12.67	[10.13,15.34]
12B IT	Reference	+1.19	[0.34,2.08]
12B IT	No reference	+2.13	[1.29,3.13]

Table 3: Paired holistic-score differences, HEXABEE-4B minus baseline, on the matched 300-item subset.

7 Discussion

The method combines three forms of supervision with different roles. Continued pretraining adapts language modeling to encyclopedic text. Multilingual answering requires selecting the information requested by a question and expressing it in a specified language; answer-text translation exposes the

model to semantic counterparts in that same response space. Final bidirectional SFT then concentrates training on the translation relation evaluated at test time. Shared answer targets provide a concrete connection between the intermediate tasks, while the final stage uses a separate source-generation process.

The results show that this complete recipe produces a stronger translator on the recorded test set. They do not establish which component causes the gains. A controlled attribution study would keep the Wikipedia-adapted checkpoint, total training budget, final translation corpus, and inference prompts fixed while varying the intermediate answering/translation mixture. Evaluating the intermediate checkpoint would further distinguish improvements before and after specialization. These comparisons are not available here. The present contribution is a fully specified system and its translation evaluation, with corpus counts that make those future comparisons possible.

8 Conclusion

We presented HEXABEE-4B, a six-language translator adapted from Gemma 3 4B IT through Wikipedia continued pretraining, multilingual answering with answer-text translation, and bidirectional translation specialization. The central data design links answering and translation through multilingual answer targets, with 5.76 million cross-language QA records embedded in a 30.15-million-record intermediate mixture. On a 36,572-example synthetic test set, the final system improves all reported aggregate metrics over public Gemma 3 4B IT and 12B IT, including chrF++ gains in every direction. This study establishes a reproducible translation-focused adaptation recipe and a direction-level account of its strengths and remaining weaknesses.

Limitations

The test sources and references are synthetic, and the judge belongs to the same teacher-service family. Reference similarity and judge preferences may favor teacher-like outputs. The results therefore do not establish quality on human-authored benchmarks or out-of-domain text. The record-level split allows topic overlap; two source strings overlap after whitespace normalization, and the audit does not establish absence of semantic duplicates or pretraining exposure. Archived inference prompts differ across systems, so adaptation and prompting are not separated. Some baseline requests encountered context limits; their empty responses remain in the evaluation. COMET truncates long inputs, limiting the text considered by that metric. The judge subset contains only ten examples per direction. No human assessment, controlled mixture ablation, evaluation of intermediate checkpoints, or held-out QA evaluation is available. Training data have uneven language coverage, empty fields, and 1,961 conflicting answer-language directives. Internal validation sets are too small for reliable model selection. These limitations constrain both causal attribution and generalization of the reported gains.

Ethical Considerations

Wikipedia-derived and synthetic data can reflect uneven representation across languages and topics. Generated translations may alter names, factual claims, or register; the model has not been validated for high-stakes translation. Data and model use must respect applicable Wikipedia attribution/share-alike obligations, upstream dataset and teacher-service terms, and Gemma terms. The archived exports do not recover complete article-level provenance, so this paper does not assert a blanket license over all text. We report existing training and generation artifacts without collecting new human annotations. Full directional reporting and retention of empty outputs help make performance differences across languages visible.

References

Iman Barati, Arash Ghafouri, and Behrouz Minaei-Bidgoli. 2026. XHotpotQA: A benchmark for cross-lingual knowledge composition in multi-hop question answering. *Preprint*, arXiv:2608.27481.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI. 2025. DeepSeek-V3.1: Model card. Hugging Face model repository. Accessed 12 September 2026.

Gemma Team. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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

Hugging Face. 2025. PEFT: Parameter-efficient fine-tuning. Software, version 0.17.1.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen. 2024. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions. *Transactions of the Association for Computational Linguistics*, 12:576–592.

Geyu Lin, Bin Wang, Zhengyuan Liu, and Nancy F. Chen. 2025. CrossIn: An efficient instruction tuning approach for cross-lingual knowledge alignment. In *Proceedings of the Second Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation*, pages 12–23, Abu Dhabi. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

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.

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.

Maja Popović. 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.

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.

Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2024. Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7961–7973, Bangkok, Thailand. 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.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2304–2317, Bangkok, Thailand. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John 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.

Unbabel. 2022. wmt22-comet-da: Model card. Hugging Face model repository. Revision 2760a223ac95.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr

Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

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.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. Question translation training for better multilingual reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8411–8423, Bangkok, Thailand. Association for Computational Linguistics.

A Continued-pretraining corpus

The five distinct source files registered for phase 1 contain the raw text rows in Table 4. Four Persian registrations refer to the same source file, so its rows are counted once in this table. The counts were measured from the available raw exports; they are neither unique-article counts nor the number of processed training sequences. The recorded training run reports 3,399,664 examples after preprocessing. No claim is made that the raw row sum alone reconstructs its tokenization, packing, or source repetition.

Language	Raw text rows
English	6,406,711
Persian	802,549
Arabic	1,216,238
Chinese	1,354,407
Egyptian Arabic	1,615,156
Total across distinct source files	11,395,061

Table 4: Available phase-1 Wikipedia sources before training preprocessing. The registered mixture repeats the Persian source four times.

B Construction and inference details

B.1 Multilingual answer inventory

The phase-2 inventory begins with three question formats (general, standard, and title-derived QA). Records are pooled by format and task in the released dataset. Their combined initial row count is 2,025,730. Gemma 3 27B IT (Gemma Team, 2025) generates Persian-to-English and Persian-to-Arabic variants, followed by English-to-Chinese, English-to-Hebrew, and English-to-Spanish variants. Thus part of the inventory is English-pivoted. Equivalence of all generated answer variants has not been exhaustively checked by human annotators. Wikipedia context supports upstream example construction; the final training record supplies only an instruction, input text, and target response.

Question columns supply QA inputs and answer columns supply QA targets. For answer-text MT, both columns contain answer variants. Sampling fractions for QA depend on the source group, direction, and response variant. For answer-text MT, the fraction of available content retained for a direction is

$$f(s,t) = \begin{cases} .70, & \{s,t\} = \{\text{fa},\text{en}\}, \\ .25, & \{s,t\} \cap \{\text{fa},\text{en}\} \neq \varnothing, \\ .08, & \text{otherwise.} \end{cases} \quad (6)$$

Here s, t are input and target languages and $s \neq t$. The first matching case applies. These fractions determine dataset construction; all retained answer-translation datasets are subsequently selected for phase 2. The construction records contain 10,695,112 MT rows; null filtering leaves 10,695,072. Across all tasks, summing retained construction rows reproduces 30,153,404 training records. The public release organizes these into 14 unified task configurations without dropping or resampling examples. Their internal split contains 30,153,102 training and 302 validation examples.

B.2 System-instruction composition

For cross-language complete-answer QA, the constructor samples a completeness directive and an answer-language directive, then combines them in randomized order with a sampled separator. Some same-language examples use only the completeness directive. Style-conditioned examples may add a formal or informal directive. The language in which a sentence is written is distinct from the answer language it requests. The construction uses random seed 42; this value controls reproducibility of sampling choices, not an additional modeling assumption. Prompt pools retain duplicate rows and their sampling mass.

B.3 Synthetic translation generation

Source generation uses a three-level topic taxonomy and attributes for genre, register, style, complexity, length, and optional linguistic/noise conditions. Stored metadata identifies DeepSeek-V3.1 (DeepSeek-AI, 2025); source generation uses temperature 0.5 and top-$p = 0.9$. The translation service is addressed through `deepseek-chat`, without a recoverable immutable serving revision for every call. A different target language is selected for each source. Translation generation uses temperature zero and a 5,000-token completion budget. The teacher is asked to preserve content, names, numbers, formatting, and register; these instructions do not guarantee translation quality.

The aggregate contains 365,718 pairs. A shuffled 90/10 record split produces 329,146 raw train and 36,572 test pairs. No split seed is recorded. Language-label normalization, pair deduplication, and removal of empty/equal texts yield 329,137 forward and 329,137 reverse training records. The net reduction is nine underlying pairs; a per-filter attribution is unavailable. The final selected total is 658,274, with an internal train/validation split of 658,267/7. The stored split memberships are preserved throughout the retrospective evaluation.

B.4 Archived inference

All three systems use temperature-zero decoding with a 5,000-token completion budget. HEXABEE-4B receives a short English direction instruction. The public Gemma models receive a longer English instruction specifying translation and formatting requirements, with quoted source text. The comparison therefore includes prompt differences. The full test contains 31 empty responses for Gemma 3 4B IT and none for the other two systems. Context-limit failures appear in the logs, so the empty responses cannot all be interpreted as linguistic failures. Empty strings and unsolicited wrappers remain in metric inputs.

B.5 Judge rubric and paired intervals

The reference-assisted judge sees source, candidate, and reference; the reference-free judge sees source and candidate. Both modes use a custom holistic 0–100 score and eight component scores, following the general model-based evaluation setting of Kocmi and Federmann (2023b,a). This is not a claim to reproduce GEMBA's exact protocol. Prompts request score caps of 10 for wrong-language output or untranslated copying, 40 for key mistranslations or hallucinations, and 60 for critical entity/number corruption. These are requested constraints, not verified error labels.

Let $J_{d,i}^{m,v}$ be the holistic judge score for system m, direction $d \in \Omega$, item $i \in \{1, \ldots, 10\}$, and view $v \in \{\text{ref}, \text{noref}\}$. For HEXABEE-4B (system h) and baseline b, the estimated score difference is

$$\widehat{\Delta}_v = \frac{1}{30} \sum_{d \in \Omega} \frac{1}{10} \sum_{i=1}^{10} (J_{d,i}^{h,v} - J_{d,i}^{b,v}). \qquad (7)$$

Each bootstrap replicate samples ten item indices with replacement within each direction, using the same sampled indices for both systems. We generate 10,000 replicates with seed 20260912 and use their 2.5th and 97.5th percentiles. These are paired, direction-stratified item-sampling intervals (Koehn, 2004).

C Phase-2 corpus composition

The following counts pool all retained records by question format and task. Summing construction records with non-null training fields exactly reproduces the training mixture. The released data have 14 unified task configurations; merging source partitions changes only packaging, not the examples. Source/target labels for answering tasks are independently checked against the notebook column policies. Counts describe records, not unique articles or independent question concepts.

Construction quality audit. The merger removes null fields but preserves empty strings. The final exports contain 6,742 empty input fields and 6,043 empty response fields (field counts

Subcorpus	Complete QA	Moderate QA	Formal QA	Informal QA	Translation	Total
General QA	3,952,557	208,364	330,194	284,650	2,481,368	7,257,133
Standard QA	7,240,185	300,875	1,540,222	411,033	5,554,592	15,046,907
Title QA	3,835,308	0	646,570	708,374	2,659,112	7,849,364

Table 5: All selected phase-2 data by question format and task, with source partitions pooled. A zero denotes an absent task export.

$s\backslash t$	fa	en	ar	zh	he	es
fa	2,025,720	764,747	202,573	101,287	19,744	1,139
en	1,418,003	2,025,730	101,287	462,076	39,489	0
ar	405,143	202,573	2,025,730	20,258	4,936	0
zh	202,572	101,287	40,514	2,025,730	4,114	822
he	49,362	29,616	9,873	4,936	987,223	0
es	107,249	76,607	15,322	15,322	4,936	1,532,130

Table 6: Complete QA records by input language (rows) and response language (columns). Diagonal cells are same-language answering; translation excludes identity directions.

$s\backslash t$	fa	en	ar	zh	he	es
fa	4,545,437	931,711	202,573	101,287	19,744	1,139
en	2,513,228	2,582,276	101,287	462,076	39,489	0
ar	460,798	202,573	2,025,730	20,258	4,936	0
zh	222,051	101,287	40,514	2,025,730	4,114	822
he	57,710	29,616	9,873	4,936	987,223	0
es	115,597	76,607	15,322	15,322	4,936	1,532,130

Table 7: All QA response variants (complete, moderate, formal, informal) records by input language (rows) and response language (columns). Diagonal cells are same-language answering; translation excludes identity directions.

can overlap within a record); no system field is empty. These entries are included in the recorded export counts. The raw construction-to-export reduction is 69 records, including 40 translation records. Two complete-answer exports contain 822 Chinese-to-Spanish and 1,139 Persian-to-Spanish records whose language-directive pools request English. Both conflicts are verified in stored examples. They account for all 1,961 cross-language Spanish-target complete-answer records. Consequently Spanish cross-language answering with a correctly specified target is not established by this subset, although Spanish is present in same-language QA and translation supervision. The historical model is evaluated as trained; the audit does not retroactively repair its supervision.

D Prompt sampling details

For instruction component g (such as answer language or extent), let $\mathcal{P}_{g,p,t}$ be the list of sentences written in language p for target language t. Let $\mathcal{A}_{g,t}$ contain the languages with available,

$s\backslash t$	fa	en	ar	zh	he	es
fa	0	1,418,003	506,429	506,429	246,806	383,029
en	1,418,003	0	506,433	506,433	246,806	383,033
ar	506,429	506,433	0	162,059	78,978	122,571
zh	506,429	506,433	162,059	0	78,978	122,571
he	246,806	246,806	78,978	78,978	0	78,978
es	383,029	383,033	122,571	122,571	78,978	0

Table 8: Answer-text MT records by input language (rows) and response language (columns). Diagonal cells are same-language answering; translation excludes identity directions.

	Answer-language		Completeness	
Prompt language	Rows	Unique	Rows	Unique
fa	1302	1110	1076	836
en	1302	944	1000	759
ar	1060	817	1032	792
zh	976	755	1024	784
he	914	691	1080	840
es	–	–	1009	728

Table 9: Representative prompt pools before record sampling. Counts use the pool whose target-language label matches its prompt language; completeness text does not itself specify the answer language. Unique counts deduplicate exact sentence strings. Repeated rows remain in the sampler. A dash indicates no such pool.

nonempty pools. The builder normalizes configured nonnegative weights $w_{g,p}$ over $\mathcal{A}_{g,t}$, chooses p, and samples row index j uniformly:

$$P(p, j \mid g, t) = \frac{w_{g,p}}{\sum_{u \in \mathcal{A}_{g,t}} w_{g,u}} \frac{1}{|\mathcal{P}_{g,p,t}|}. \quad (8)$$

Here $p \in \mathcal{A}_{g,t}$ and $1 \le j \le |\mathcal{P}_{g,p,t}|$. The configured available weights have a positive sum. Repeated strings in a list therefore receive repeated sampling mass; uniformity is over rows, not distinct paraphrases. The prompt-generation scripts expand seed sentences with model-generated paraphrases, combine outputs, and substitute localized language names into target placeholders. Appendix C reports pool sizes and exact duplicate counts. Changing p changes how a directive is expressed; changing t changes the required response. Our core supervision concerns the latter.

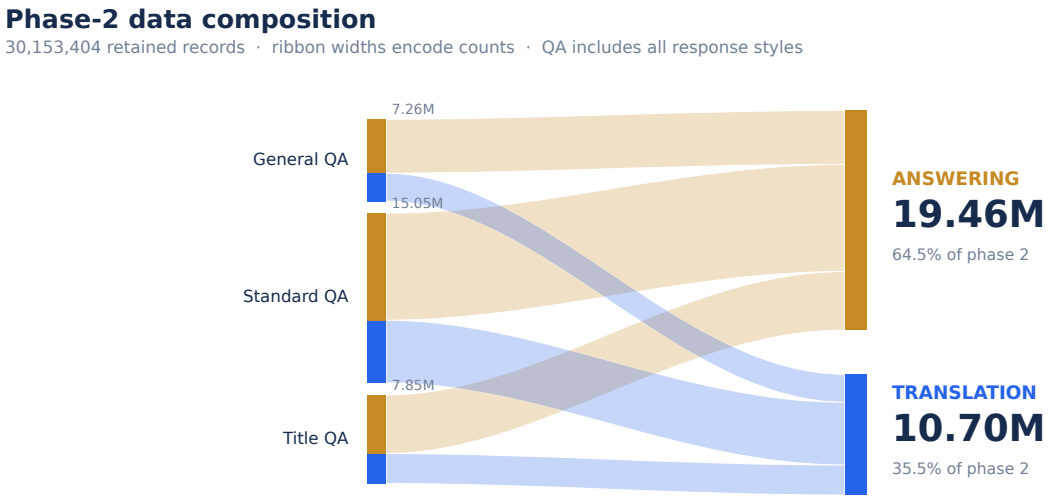


Figure 5: Retained phase-2 records flowing from the three pooled question formats to answering and translation tasks. Ribbon widths encode record counts. Translation and several answering variants are jointly selected; source-row reuse across these tasks means the plotted counts are not independent documents.

D.1 Final-phase translation instructions

For each direction, the constructor chooses a prompt-language pool, then uniformly samples a sentence from that pool. Its principal weights are

$$\pi(p \mid s,t) = \begin{cases} .55, & p = t, \\ .30, & p = s, \\ w_p(s,t), & p \notin \{s,t\}, \end{cases} \tag{9}$$

The residual satisfies $\sum_{p \notin \{s,t\}} w_p = .15$. Remaining weights use a script-based heuristic: Persian/Arabic/Hebrew form an RTL group, English/Spanish a Latin group, and Chinese a CJK group. These are implementation groups, not genealogical language families. The allocation below specifies these weights. Empty pools are excluded and available weights renormalized by the sampler. Seed 42 controls the sampled construction choices; it does not retroactively seed source generation or the earlier raw split.

A final-phase translation record has the training schema `system`, `instruct`, `output`. Per-direction intermediate files retain explicit source/target labels, but the merger keeps the three training fields, drops missing values, and escapes media-like tags as text. Here p denotes the language of the sampled translation instruction and supplies surface-form diversity. It does not add a loss that explicitly equates outputs across prompt languages, and our evaluation does not estimate such invariance.

Let $U = \mathcal{L} \setminus \{s,t\}$, $M \subseteq U$ contain languages sharing an implementation script group with s or t, and $O = U \setminus M$. The constructor reserves 0.55 for t and 0.30 for s. With residual $R = .15$, define

$$g = \min(R/2, .01|O|), \tag{10}$$

$$h = \begin{cases} \min(.10, (R-g)/|M|), & |M| > 0, \\ 0, & |M| = 0, \end{cases} \tag{11}$$

$$r_0 = R - g - |M|h. \tag{12}$$

For $p \in U$, the residual weight is

$$w_p = \mathbf{1}[p \in O]\frac{g}{|O|} + \mathbf{1}[p \in M]h + \frac{r_0}{|U|}, \tag{13}$$

where a fraction on an empty group contributes zero. This matches the implementation's residual redistribution before rounding and final sampler normalization. For Persian-to-English, the unrounded weights in order (fa,en,es,he,zh,ar) are $(.30, .55, .046\overline{6}, .046\overline{6}, .01, .046\overline{6})$. Here $\mathbf{1}[\cdot]$ is the indicator function. With $\mathcal{P}_{p,s,t}$ the nonempty prompt pool, the probability of selecting row j from language pool p is

$$P(j, p \mid s,t) = \frac{\pi(p \mid s,t)}{|\mathcal{P}_{p,s,t}|}, \quad 1 \le j \le |\mathcal{P}_{p,s,t}|. \tag{14}$$

If pools are unavailable, π is renormalized over the available pools. The six-language assignment

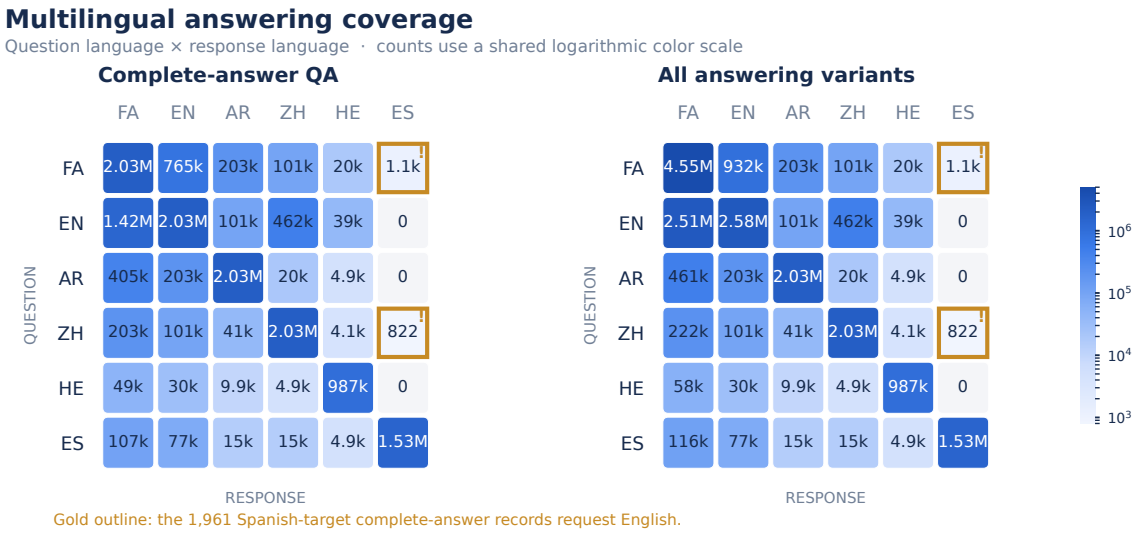


Figure 6: Realized question/answer language coverage. Left: complete-answer QA. Right: all answering variants. Color uses a logarithmic count scale. Gold outlines mark the two Spanish-target cells whose complete-answer directives incorrectly request English (1,961 records). Zero cells indicate absent exported combinations.

has full theoretical support, but observed coverage must be measured from instantiated prompts rather than inferred from the formula.

D.2 Partition audit

Streaming comparison of raw source strings finds zero exact train–test source overlaps. After collapsing whitespace, two test source texts match training sources, corresponding to 0.0055% of the test partition; neither occurs in the 300-item judge subset. This is a narrow source-text audit: it does not prove absence of semantic duplicates, topic overlap, overlap with reverse inputs, or exposure during pretraining. The split is by record rather than by topic or Wikipedia article. We retain this distinction when interpreting generalization.

E Training and serialization

The final stage uses PEFT 0.17.1 (Hugging Face, 2025), Transformers 4.55.4 (Wolf et al., 2020), PyTorch 2.7.1+cu126, Datasets 3.6.0, Tokenizers 0.21.1, and DeepSpeed 0.16.9 with ZeRO stage 0. AdamW (Loshchilov and Hutter, 2019) parameters are $\beta_1 = .9$, $\beta_2 = .999$, and $\epsilon = 10^{-8}$. Training uses 4,096-token sequences, response-only supervision, checkpoints every 600 steps, and internal evaluation every 200 steps. The final adapter has no bias adaptation, DoRA, or rank-stabilized LoRA. The saved configuration of the continued cross-lingual adapter also uses dropout .005 despite a different value in the nearby YAML.

For a text-only example, the serialized prefix is shown schematically below; literal line breaks separate fields:

```
<bos><start_of_turn>user
[system instruction]

[source text]<end_of_turn>
<start_of_turn>model
[target text]<end_of_turn>
```

$$W'_j = W_j + \frac{\alpha}{r} B_j A_j, \tag{15}$$

where $W_j \in \mathbb{R}^{d_{o,j} \times d_{i,j}}$ is projection j with input dimension $d_{i,j}$ and output dimension $d_{o,j}$, $A_j \in \mathbb{R}^{r \times d_{i,j}}$ and $B_j \in \mathbb{R}^{d_{o,j} \times r}$ are trainable factors, r is adapter rank, and α is its scale (Hu et al., 2022). The saved adapters use $r = 256$, $\alpha = 512$, and dropout 0.005. These standard LoRA settings are implementation choices; the method's supervision relations are given in the main text.

The model's 34 text layers have hidden size 2,560, eight query heads, four key/value heads, and MLP intermediate size 10,240. The multimodal checkpoint retains its vision components, although final-stage adaptation affects only text projections. The nominal 4B size identifies the Gemma model family. For the adapted projections, the parameter count is

$$N_{\text{LoRA}} = r \sum_j (d_{i,j} + d_{o,j}) = 476{,}839{,}936. \tag{16}$$

Each layer contributes two $2{,}560 \times 2{,}048$ attention projections, two $2{,}560 \times 1{,}024$ projections, and

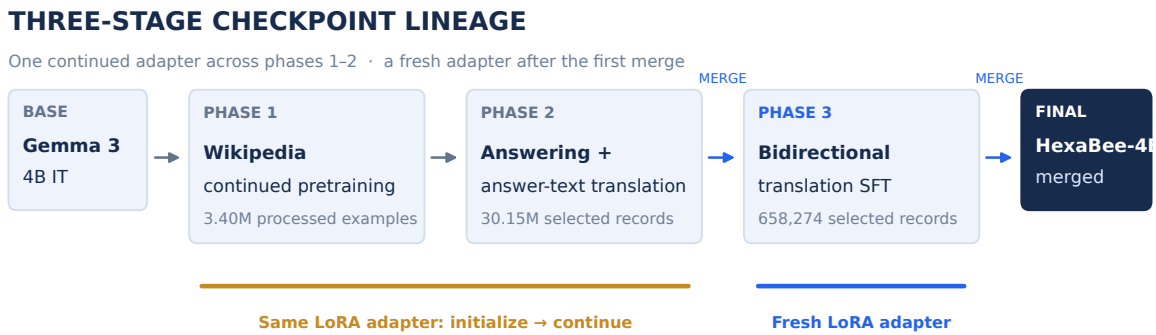


Figure 7: Training lineage. The translation stage starts a new adapter on the merged cross-lingual model. Phase 2 combines system-directed answering with translations of multilingual answer texts; phase 3 specializes the merged model on independent synthetic translations.

Setting	Wikipedia adaptation	Cross-lingual SFT	Translation SFT
Training examples	3,399,664	30,153,102	658,267
Internal validation examples	Not established	302	7
Epochs / optimizer steps	1 / 13,280	1 / 6,544	1 / 3,429
CUDA processes	4	6	2
Per-device batch / accumulation	16 / 4	12 / 64	12 / 8
Effective batch	256	4,608	192
Peak learning rate	10^{-4}	10^{-4}	10^{-4}
Schedule / warmup fraction	cosine / .05	cosine / .05	cosine / .05
Seed	111	42	42
Runtime (hours)	59.99	306.90	27.45
Adapter handling	new on public base	continue previous	new after merge

Table 10: Executed settings from training logs and generated model cards. Counts have stage-specific semantics and should not be added as unique documents. Current nearby YAMLs may describe other runs; logged values take precedence.

three $2{,}560 \times 10{,}240$ MLP projections, up to transposition. Summing their input/output dimensions over 34 layers verifies the adapter count.

F Metrics and results by language

Language codes are ar (Arabic), zh (Chinese), en (English), he (Hebrew), fa (Persian), and es (Spanish). Direction tables keep source and target distinct. Target summaries give each of the five incoming directions equal weight; they are direction-macro summaries, not recomputed pooled corpora. Judge dimension scores remain on a 0–100 scale and are not calibrated accuracy percentages. All raw model responses are retained for scoring, including quotation wrappers. The machine-readable artifacts include metric signatures and sample counts. Writing $M_{s,t}$ for the score of direction (s, t), the target-language summary is

$$M_{\to t} = \frac{1}{5} \sum_{s \in \mathcal{L} \setminus \{t\}} M_{s,t}. \tag{17}$$

Pooled corpus chrF++ is 60.77 for Gemma 3 4B IT, 71.13 for Gemma 3 12B IT, and 77.30 for HexaBee-4B. These differ from the direction-macro values in Table 2. Let $\mathcal{G}$ index the configured n-gram types, each defined by a unit (character or word) and an order. For $g \in \mathcal{G}$, let H_g, R_g, and C_g denote corpus hypothesis, reference, and matched counts. With $E = \{g \in \mathcal{G} : H_g > 0, R_g > 0\}$, the effective-order averages are

$$P = \frac{1}{|E|} \sum_{g \in E} \frac{C_g}{H_g}, \qquad R = \frac{1}{|E|} \sum_{g \in E} \frac{C_g}{R_g}. \tag{18}$$

For chrF++, orders include character 1–6 and word 1–2; chrF uses only the character orders. SacreBLEU then computes

$$\mathrm{chrF}_\beta = 100(1 + \beta^2) \frac{PR}{\beta^2 P + R}, \quad \beta = 2, \tag{19}$$

with a zero score for an empty effective set or zero denominator. It aggregates corpus counts before the harmonic combination, rather than averaging sentence scores. BLEU is computed from

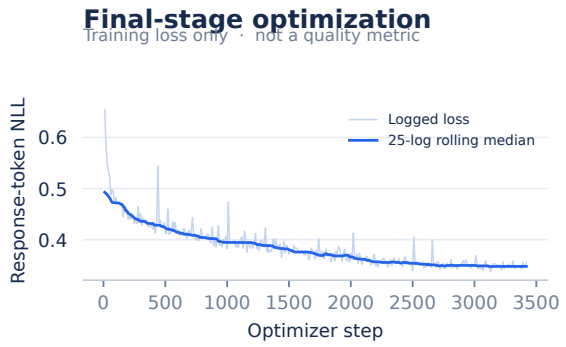


Figure 8: Recorded response-token training loss during final translation adaptation, with a centered rolling median over up to 25 logged values for visual guidance. This is an optimization diagnostic, not a translation-quality or generalization metric.

clipped corpus precisions p_n and total hypothesis/reference lengths c, r:

$$\mathrm{BLEU} = 100\,\mathrm{BP}\exp\left(\frac{1}{4}\sum_{n=1}^{4}\log p_n\right), \quad (20)$$

where $\mathrm{BP} = \min(1, \exp(1 - r/c))$ for $c > 0$, with SacreBLEU's exponential smoothing for zero matches. TER uses the Tercom-style edit search:

$$\mathrm{TER} = 100\frac{N_{\mathrm{sub}} + N_{\mathrm{del}} + N_{\mathrm{ins}} + N_{\mathrm{shift}}}{N_{\mathrm{ref}}}. \quad (21)$$

Here N_{sub}, N_{del}, N_{ins}, and N_{shift} count substitutions, deletions, insertions, and shifts; N_{ref} is the total reference-token count. This search approximates the edit sequence with shifts; the score should not be described as a certified global minimum of all possible edit programs. Corpus and macro aggregation are kept distinct for both metrics.

Let f_ω be the COMET regression model with fixed checkpoint parameters ω and τ_{512} its tokenizer/encoder truncation operation. For N examples, let x_i, $\hat{y}_i$, and y_i denote source, candidate, and reference text, respectively. The scaled corpus mean is

$$\frac{100}{N}\sum_{i=1}^{N} f_\omega(\tau_{512}(x_i), \tau_{512}(\hat{y}_i), \tau_{512}(y_i)). \quad (22)$$

Direction-macro COMET instead averages the 30 direction means. Scaling by 100 does not turn the regression output into an accuracy percentage.

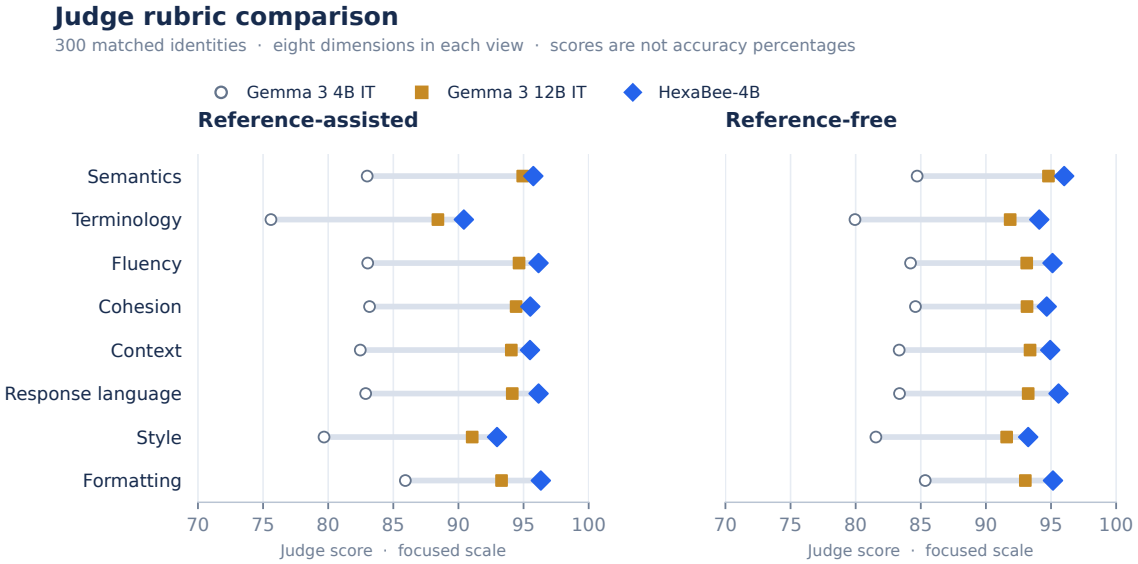


Figure 9: All eight judge dimensions, averaged over the same 300 identities. Both views compare the same three systems; they remain judgments from one evaluator and are not independent human assessments.

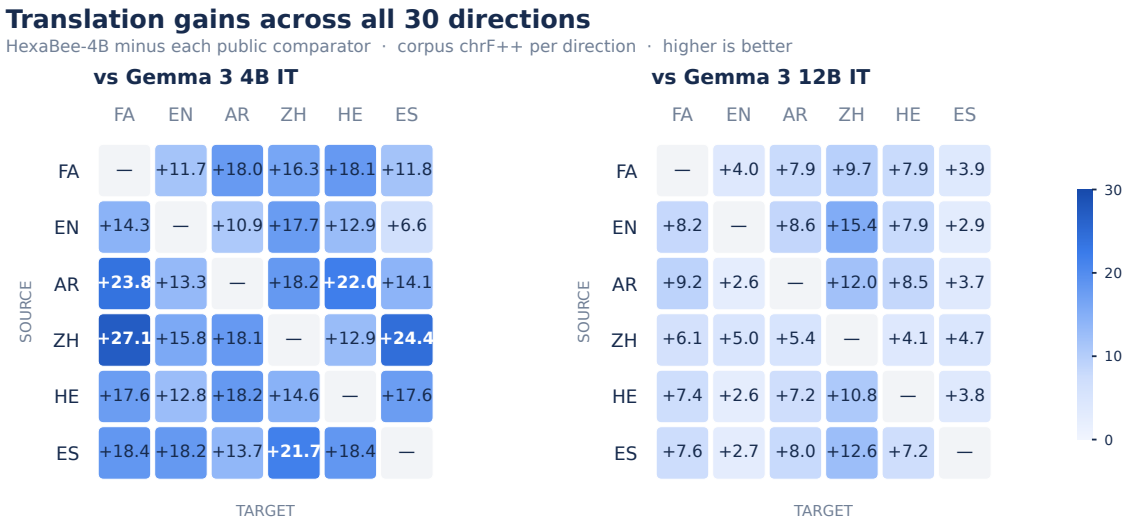


Figure 10: Full-test chrF++ gains of HEXABEE-4B over both public comparison systems across all 30 source–target directions.

Target	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.
All	4B IT	36,572	40.70	58.91	57.11	51.40	85.88	80.32	81.43
	12B IT	36,572	50.54	68.70	66.83	38.89	88.20	91.73	91.97
	HEXABEE-4B	36,572	**62.03**	**75.08**	**73.75**	**30.05**	**89.86**	**92.92**	**94.10**
ar	4B IT	6,120	34.09	62.25	59.33	56.99	86.64	82.56	81.60
	12B IT	6,120	43.14	70.43	67.69	44.60	88.52	89.12	89.78
	HEXABEE-4B	6,120	**56.17**	**77.09**	**75.10**	**33.62**	**90.17**	**92.72**	**93.60**
zh	4B IT	6,135	35.22	31.87	30.97	60.19	84.52	75.22	76.00
	12B IT	6,135	40.15	38.40	36.57	54.28	87.01	89.92	90.82
	HEXABEE-4B	6,135	**54.78**	**50.31**	**48.67**	**38.68**	**90.00**	**91.82**	**92.60**
en	4B IT	6,010	55.28	74.13	72.40	36.35	87.40	86.26	86.84
	12B IT	6,010	67.16	84.77	83.36	22.59	88.90	**94.78**	94.42
	HEXABEE-4B	6,010	**75.06**	**87.68**	**86.76**	**17.00**	**89.83**	93.52	**94.66**
he	4B IT	6,196	31.94	54.98	53.02	59.27	85.87	76.00	78.44
	12B IT	6,196	43.32	64.53	62.78	46.14	88.17	91.08	91.40
	HEXABEE-4B	6,196	**54.71**	**70.90**	**69.90**	**42.17**	**89.68**	**93.16**	**94.30**
fa	4B IT	6,069	32.44	57.57	55.67	59.16	84.81	76.86	77.86
	12B IT	6,069	43.61	70.24	68.22	41.59	88.32	91.04	92.60
	HEXABEE-4B	6,069	**57.43**	**77.34**	**75.90**	**31.68**	**89.91**	**92.06**	**94.34**
es	4B IT	6,042	55.21	72.69	71.24	36.44	86.02	85.04	87.86
	12B IT	6,042	65.85	83.83	82.35	24.13	88.29	**94.44**	92.82
	HEXABEE-4B	6,042	**74.04**	**87.18**	**86.16**	**17.13**	**89.57**	94.24	**95.12**

Table 11: Overall and target-language comparisons. Automatic metrics are direction-macro averages on the full test set; holistic judge means use 300 matched items overall or 50 per target. COMET is scaled by 100; TER is lower-is-better. Bold marks the best displayed score and underlining the second distinct displayed score within each group; ties share rank. The adapted model is tinted for easy comparison. These recorded systems use different inference prompts.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++_{10}	TER_{10} ↓	COMET_{10}
$zh \to ar$	4B IT	1229	23.68	51.49	48.28	80.63	84.07	72.60	71.60	37.49	100.59	80.35
	12B IT	1229	32.53	64.44	60.91	61.20	87.91	**91.60**	92.50	62.38	53.76	87.21
	HEXABEE-4B	1229	**42.46**	**69.06**	**66.34**	**57.30**	**89.15**	90.40	**93.00**	**71.36**	**38.78**	**88.68**
$en \to ar$	4B IT	1213	37.50	66.77	63.77	51.70	87.99	86.50	85.20	65.44	44.06	88.04
	12B IT	1213	42.30	68.56	66.09	47.70	87.47	81.10	82.20	57.25	48.09	85.10
	HEXABEE-4B	1213	**55.60**	**76.61**	**74.66**	**35.72**	**90.06**	**93.40**	**94.10**	**76.86**	**27.55**	**90.48**
$he \to ar$	4B IT	1230	39.08	63.41	61.23	48.02	86.42	78.30	74.70	50.31	52.36	80.29
	12B IT	1230	51.01	74.60	72.25	32.19	89.30	88.40	87.90	64.18	38.88	84.90
	HEXABEE-4B	1230	**63.65**	**80.99**	**79.44**	**23.26**	**90.59**	**92.60**	**93.60**	**69.00**	**31.11**	**86.10**
$fa \to ar$	4B IT	1228	32.88	63.15	60.05	56.62	87.51	88.70	89.80	68.97	38.75	89.71
	12B IT	1228	46.62	72.69	70.14	38.53	89.75	92.60	93.40	73.76	31.76	90.41
	HEXABEE-4B	1228	**61.00**	**79.59**	**78.00**	**24.43**	**91.27**	**94.40**	**95.10**	**81.87**	**20.81**	**91.55**
$es \to ar$	4B IT	1220	37.28	66.41	63.33	47.96	87.21	86.70	86.70	64.12	44.99	88.36
	12B IT	1220	43.22	71.87	69.06	43.37	88.15	91.90	**92.90**	69.86	37.83	88.93
	HEXABEE-4B	1220	**58.14**	**79.19**	**77.07**	**27.40**	**89.78**	**92.80**	92.20	**76.58**	**27.15**	**89.85**

Table 12: All systems translating into Arabic. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++$_{10}$	TER$_{10}$ ↓	COMET$_{10}$
$ar \to zh$	4B IT	1201	36.56	32.17	30.93	58.04	83.81	73.60	73.90	22.92	62.38	81.89
	12B IT	1201	42.36	39.84	37.13	50.06	86.94	91.80	91.90	30.13	51.75	87.06
	HEXABEE-4B	1201	**57.09**	**51.82**	**49.12**	**35.01**	**89.69**	**93.70**	**94.20**	**40.72**	**35.73**	**89.56**
$en \to zh$	4B IT	1276	30.42	27.90	27.56	69.42	83.44	82.30	82.80	29.76	49.49	83.68
	12B IT	1276	32.77	31.48	29.94	67.91	85.07	91.80	93.50	30.05	44.73	87.02
	HEXABEE-4B	1276	**50.33**	**45.82**	**45.30**	**45.39**	**90.02**	**93.00**	**93.60**	**44.30**	**31.80**	**89.95**
$he \to zh$	4B IT	1202	41.02	38.25	37.22	51.04	86.17	76.10	77.20	29.02	66.88	84.29
	12B IT	1202	44.57	42.44	40.97	45.95	87.67	93.00	93.50	38.94	52.52	87.92
	HEXABEE-4B	1202	**58.14**	**53.41**	**51.80**	**33.49**	**89.75**	**94.20**	**94.30**	**50.28**	**34.08**	**90.12**
$fa \to zh$	4B IT	1251	38.95	35.74	34.01	53.12	86.19	74.70	75.00	37.67	48.16	83.33
	12B IT	1251	45.23	42.86	40.58	44.42	88.67	92.10	90.50	46.31	34.15	89.23
	HEXABEE-4B	1251	**57.23**	**52.86**	**50.30**	**34.91**	**90.41**	**94.40**	**94.60**	**58.36**	**23.84**	**91.03**
$es \to zh$	4B IT	1205	29.14	25.27	25.15	69.33	83.01	69.40	71.10	13.03	71.67	79.39
	12B IT	1205	35.84	35.37	34.24	63.07	86.73	80.90	84.70	17.67	62.46	82.49
	HEXABEE-4B	1205	**51.11**	**47.63**	**46.84**	**44.61**	**90.14**	**83.80**	**86.30**	**24.63**	**44.99**	**85.80**

Table 13: All systems translating into Chinese. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++$_{10}$	TER$_{10}$ ↓	COMET$_{10}$
$ar \to en$	4B IT	1200	59.91	75.60	74.32	29.38	86.98	82.00	83.30	64.16	41.19	85.14
	12B IT	1200	70.24	86.13	84.97	21.03	88.74	**92.40**	**94.80**	81.37	23.10	88.22
	HEXABEE-4B	1200	**76.36**	**88.32**	**87.58**	**16.02**	**89.58**	90.10	94.60	**85.46**	**16.95**	**89.33**
$zh \to en$	4B IT	1213	38.89	64.46	62.03	59.53	86.20	64.40	65.90	53.20	73.92	81.00
	12B IT	1213	47.49	75.48	72.87	40.65	87.93	**95.70**	94.90	73.06	39.17	88.17
	HEXABEE-4B	1213	**58.71**	**79.80**	**77.87**	**32.67**	**89.16**	95.20	**95.60**	**78.02**	**30.94**	**89.30**
$he \to en$	4B IT	1185	63.49	77.80	76.72	28.99	87.63	**93.40**	**92.40**	81.10	21.63	87.26
	12B IT	1185	73.72	87.81	86.82	16.72	89.22	91.80	89.20	86.07	14.94	87.78
	HEXABEE-4B	1185	**79.87**	**90.07**	**89.47**	**12.54**	**90.03**	90.80	90.00	**89.03**	**10.89**	**88.69**
$fa \to en$	4B IT	1234	57.78	76.96	75.53	29.13	88.46	95.10	95.40	77.19	27.66	87.40
	12B IT	1234	67.46	84.48	83.21	20.09	89.29	96.60	95.80	80.82	22.40	87.88
	HEXABEE-4B	1234	**76.31**	**88.01**	**87.22**	**14.41**	**90.16**	**96.70**	**96.70**	**85.43**	**16.07**	**88.65**
$es \to en$	4B IT	1178	56.34	75.84	73.42	34.73	87.71	96.40	97.20	86.98	16.42	89.07
	12B IT	1178	76.88	89.96	88.91	14.43	89.31	**97.40**	**97.40**	88.29	13.85	89.11
	HEXABEE-4B	1178	**84.06**	**92.21**	**91.63**	**9.37**	**90.24**	94.80	96.40	**91.34**	**9.19**	**89.94**

Table 14: All systems translating into English. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++$_{10}$	TER$_{10}$ ↓	COMET$_{10}$
$ar \to he$	4B IT	1255	33.86	55.36	53.52	55.20	85.90	70.40	66.30	48.01	52.39	84.15
	12B IT	1255	48.39	68.91	67.01	37.34	88.91	91.70	93.20	70.71	30.88	88.60
	HEXABEE-4B	1255	**61.41**	**76.53**	**75.50**	**28.87**	**91.03**	**93.00**	**94.70**	**78.93**	**20.92**	**90.49**
$zh \to he$	4B IT	1232	19.81	39.55	38.42	88.40	82.50	85.00	87.00	53.38	61.20	85.05
	12B IT	1232	27.88	48.10	47.27	**78.13**	85.34	90.30	88.60	58.72	53.58	86.56
	HEXABEE-4B	1232	**35.82**	**51.34**	**51.36**	90.09	**86.74**	**93.70**	**94.70**	**63.72**	**46.14**	**87.24**
$en \to he$	4B IT	1216	36.21	61.46	59.13	49.77	87.17	71.20	77.20	59.56	44.59	86.58
	12B IT	1216	44.79	65.92	64.17	41.62	88.31	92.60	92.10	64.07	38.82	88.78
	HEXABEE-4B	1216	**56.73**	**73.34**	**72.04**	**32.13**	**89.85**	**93.40**	**92.80**	**71.85**	**28.26**	**89.52**
$fa \to he$	4B IT	1238	33.83	58.93	56.54	52.35	87.53	70.40	75.60	54.20	51.63	87.48
	12B IT	1238	45.84	69.08	66.79	38.09	89.74	90.10	91.20	72.00	32.67	90.98
	HEXABEE-4B	1238	**58.38**	**76.09**	**74.67**	**33.36**	**90.94**	**92.10**	**94.50**	**79.33**	**23.28**	**92.35**
$es \to he$	4B IT	1255	35.99	59.59	57.50	50.63	86.26	83.00	86.10	65.78	53.33	88.65
	12B IT	1255	49.71	70.65	68.68	35.51	88.55	90.70	91.90	73.23	29.21	89.87
	HEXABEE-4B	1255	**61.20**	**77.18**	**75.92**	**26.40**	**89.85**	**93.60**	**94.80**	**80.26**	**21.41**	**90.87**

Table 15: All systems translating into Hebrew. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++$_{10}$	TER$_{10}$ ↓	COMET$_{10}$
$ar \to fa$	4B IT	1228	32.96	57.46	55.29	56.93	84.60	73.20	75.10	58.50	48.14	84.77
	12B IT	1228	45.58	71.82	69.85	40.84	88.30	**89.80**	92.60	70.16	34.38	89.77
	HexaBee-4B	1228	**62.29**	**80.41**	**79.05**	**23.95**	**90.46**	89.50	**95.10**	**80.22**	**21.59**	**91.29**
$zh \to fa$	4B IT	1208	19.17	41.95	40.29	90.09	81.51	51.60	52.70	32.82	98.86	79.60
	12B IT	1208	32.94	63.68	61.27	55.33	87.85	89.40	91.50	63.25	49.32	89.01
	HexaBee-4B	1208	**45.05**	**69.11**	**67.38**	**55.17**	**89.04**	**91.00**	**93.10**	**69.82**	**36.28**	**90.41**
$en \to fa$	4B IT	1239	34.35	62.41	60.48	52.63	86.10	89.70	88.80	64.17	48.39	87.44
	12B IT	1239	42.64	68.48	66.60	41.52	88.24	91.70	91.90	67.09	42.60	88.90
	HexaBee-4B	1239	**56.25**	**76.24**	**74.81**	**29.15**	**90.13**	**93.00**	**93.80**	**76.46**	**27.69**	**90.36**
$he \to fa$	4B IT	1198	40.23	64.44	62.67	44.97	86.19	90.40	91.80	63.53	43.59	86.47
	12B IT	1198	50.15	74.71	72.89	33.55	88.98	93.40	94.80	72.82	32.88	88.59
	HexaBee-4B	1198	**63.73**	**81.50**	**80.26**	**23.03**	**90.18**	**94.50**	**95.90**	**78.81**	**22.59**	**89.28**
$es \to fa$	4B IT	1196	35.51	61.60	59.63	51.17	85.67	79.40	80.90	55.27	55.07	84.60
	12B IT	1196	46.74	72.49	70.47	36.71	88.22	90.90	92.20	65.24	43.73	88.41
	HexaBee-4B	1196	**59.82**	**79.44**	**78.03**	**27.11**	**89.71**	**92.30**	**93.80**	**74.01**	**31.84**	**90.33**

Table 16: All systems translating into Persian. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

Direction	System	N	BLEU	chrF	chrF++	TER↓	COMET	Ref. J.	No-ref. J.	chrF++$_{10}$	TER$_{10}$ ↓	COMET$_{10}$
$ar \to es$	4B IT	1257	59.24	74.94	73.59	30.24	85.60	88.20	91.10	82.38	20.28	87.14
	12B IT	1257	68.19	85.36	84.08	20.85	87.93	**94.10**	**96.30**	86.38	14.91	88.39
	HexaBee-4B	1257	**77.23**	**88.55**	**87.73**	**14.41**	**89.37**	91.80	94.20	**88.63**	**12.20**	**88.66**
$zh \to es$	4B IT	1173	32.40	55.60	53.72	68.49	83.08	81.90	86.20	63.96	53.96	86.22
	12B IT	1173	48.63	75.81	73.37	44.50	87.27	93.60	92.90	75.41	35.64	88.58
	HexaBee-4B	1173	**58.16**	**80.03**	**78.11**	**31.49**	**89.03**	**94.10**	**94.90**	**81.32**	**24.58**	**89.61**
$en \to es$	4B IT	1188	66.60	84.25	82.63	23.34	88.08	93.20	94.90	81.53	22.75	87.87
	12B IT	1188	73.09	87.64	86.37	17.24	88.72	**94.30**	95.90	84.14	19.54	87.99
	HexaBee-4B	1188	**79.33**	**90.12**	**89.27**	**12.45**	**89.61**	93.90	**96.20**	**86.92**	**14.98**	**88.84**
$he \to es$	4B IT	1183	58.27	71.77	70.79	30.88	85.57	78.40	77.20	58.37	49.60	82.44
	12B IT	1183	71.49	85.66	84.60	17.18	88.34	96.00	86.70	85.00	30.21	87.60
	HexaBee-4B	1183	**79.39**	**89.05**	**88.43**	**11.91**	**89.55**	**96.50**	**97.00**	**89.36**	**11.67**	**89.92**
$fa \to es$	4B IT	1241	59.54	76.89	75.45	29.25	87.78	83.50	89.90	77.75	27.47	86.37
	12B IT	1241	67.83	84.70	83.34	20.90	89.22	94.20	92.30	79.56	24.97	86.03
	HexaBee-4B	1241	**76.09**	**88.16**	**87.27**	**15.38**	**90.28**	**94.90**	**93.30**	**84.07**	**20.04**	**87.36**

Table 17: All systems translating into Spanish. Columns through COMET use the full direction-specific test set (N). The last 5 columns use the ten shared judge-subset items per direction. Bold/underline mark first/second within the same direction and metric; TER is lower-is-better.

		Reference-assisted judge								Reference-free judge							
Dir.	System	Sem.	Term.	Flu.	Coh.	Ctx.	Lang.	Style	Fmt.	Sem.	Term.	Flu.	Coh.	Ctx.	Lang.	Style	Fmt.
$zh \to ar$	4B IT	72.0	66.5	73.8	72.1	71.8	73.9	68.8	73.5	74.1	69.3	74.0	72.7	72.7	74.8	71.5	75.8
	12B IT	**94.8**	87.5	94.2	92.8	93.8	94.6	90.5	93.0	**95.1**	92.0	91.8	92.6	92.8	93.6	91.6	94.0
	HexaBee-4B	92.4	**89.3**	**95.2**	**93.3**	**93.9**	**94.8**	**91.3**	**96.0**	95.0	**92.1**	**94.3**	**93.8**	**94.5**	**95.0**	**92.9**	**95.0**
$en \to ar$	4B IT	91.0	80.2	90.2	91.0	90.1	89.2	85.5	91.5	88.0	79.7	88.5	88.6	86.6	89.0	85.0	91.5
	12B IT	83.3	76.0	94.3	94.5	82.8	84.5	81.3	95.0	94.0	88.5	93.4	92.2	93.2	84.0	91.2	94.5
	HexaBee-4B	**95.7**	**88.4**	**97.6**	**96.1**	**95.9**	**97.1**	**93.2**	**97.3**	**96.2**	**95.0**	**94.3**	**94.9**	**95.3**	**94.9**	**92.2**	**95.5**
$he \to ar$	4B IT	79.1	73.8	80.3	80.3	79.5	80.1	77.0	88.0	77.0	71.2	77.4	78.6	76.1	77.0	75.5	79.8
	12B IT	93.8	83.2	92.4	91.2	91.6	90.3	88.2	88.5	90.7	85.5	88.9	89.5	89.7	88.1	88.7	88.5
	HexaBee-4B	**96.3**	**89.7**	**96.0**	**94.7**	**92.8**	**95.6**	**92.0**	**96.0**	**95.3**	**93.7**	**94.8**	**94.2**	**93.9**	**95.1**	**91.9**	**93.9**
$fa \to ar$	4B IT	92.7	84.0	93.8	92.9	91.3	93.2	89.9	94.0	93.2	88.1	91.1	91.2	91.1	91.1	88.4	90.0
	12B IT	96.5	87.1	95.7	95.5	94.8	95.4	92.8	94.5	95.8	93.4	94.1	94.2	94.3	95.0	92.9	94.0
	HexaBee-4B	**98.3**	**91.7**	**98.2**	**97.5**	**96.5**	**97.9**	**95.0**	**97.9**	**97.0**	**95.1**	**95.8**	**95.3**	**96.2**	**95.8**	**94.2**	**94.4**
$es \to ar$	4B IT	90.8	81.0	89.6	90.4	88.7	87.9	86.9	88.0	89.8	83.1	87.1	88.8	89.4	88.2	86.9	90.5
	12B IT	95.5	87.6	**96.1**	**95.3**	94.7	**95.3**	**92.0**	93.5	**95.7**	**93.6**	**92.9**	92.8	**94.5**	93.8	**91.4**	**94.5**
	HexaBee-4B	**95.9**	**89.8**	95.8	**95.3**	**94.8**	95.1	91.3	**95.0**	94.2	91.0	**92.9**	**93.3**	92.7	**94.0**	90.5	94.3

Table 18: Every judge dimension for translations into Arabic, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Dir.	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
$ar \to zh$	4B IT	75.8	72.4	74.9	73.7	74.5	75.1	72.1	74.5	76.7	72.9	75.3	74.4	74.8	75.3	72.4	74.0
	12B IT	95.6	88.5	94.1	93.8	94.7	93.6	90.6	91.5	95.1	91.8	94.4	92.4	93.3	92.4	90.8	91.5
	HEXABEE-4B	**95.9**	**92.1**	**96.5**	**95.3**	**95.6**	**96.3**	**92.5**	**96.4**	**96.0**	**94.0**	**95.4**	**94.1**	**94.5**	**95.7**	**92.5**	**95.0**
$en \to zh$	4B IT	84.8	80.5	80.7	81.1	83.1	82.5	80.0	92.5	86.0	81.7	90.9	93.5	84.2	82.7	81.2	94.5
	12B IT	95.4	**90.4**	91.0	92.4	93.5	92.3	88.8	90.5	96.0	**93.7**	93.2	93.4	94.3	93.9	92.2	93.5
	HEXABEE-4B	**96.3**	90.0	**95.7**	**94.1**	**96.0**	**96.2**	**91.7**	**96.0**	**96.2**	92.5	**95.2**	**94.0**	**94.7**	**95.5**	**92.4**	**95.0**
$he \to zh$	4B IT	80.6	69.1	80.5	80.4	80.4	82.3	77.9	81.5	89.5	79.4	92.8	91.0	90.1	82.7	89.2	93.5
	12B IT	96.0	**91.9**	94.4	94.3	94.9	93.7	91.2	93.0	**96.1**	**94.0**	93.6	93.5	**94.4**	94.1	92.9	94.7
	HEXABEE-4B	**97.1**	91.3	**96.5**	**95.9**	**95.8**	**96.4**	**94.1**	**97.0**	**96.1**	**94.0**	**95.1**	**94.5**	94.2	**95.5**	**93.4**	**95.0**
$fa \to zh$	4B IT	76.5	70.7	75.2	75.9	75.6	75.7	72.0	75.5	85.9	82.3	85.4	85.0	75.6	75.7	74.2	84.8
	12B IT	94.8	90.0	94.9	93.4	93.2	95.3	90.4	94.0	93.8	87.5	93.1	92.5	93.3	94.0	90.8	92.0
	HEXABEE-4B	**97.0**	**92.8**	**96.4**	**95.9**	**95.3**	**96.5**	**94.0**	**96.3**	**96.1**	**93.6**	**96.0**	**94.9**	**95.6**	**95.9**	**94.6**	**94.5**
$es \to zh$	4B IT	72.6	63.3	69.8	71.1	70.6	68.6	66.4	65.0	74.6	68.5	72.0	72.3	72.7	72.5	70.7	70.0
	12B IT	82.7	79.1	82.3	82.4	82.8	82.3	79.8	83.5	86.3	85.5	85.6	85.0	85.8	89.0	84.3	84.0
	HEXABEE-4B	**86.5**	**82.6**	**86.7**	**84.9**	**85.7**	**86.4**	**83.7**	**85.5**	**87.4**	**86.2**	**89.8**	**86.8**	**87.1**	**89.4**	**85.2**	**88.0**

Table 19: Every judge dimension for translations into Chinese, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Dir.	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
$ar \to en$	4B IT	83.3	75.0	85.0	84.0	83.7	85.1	80.6	86.5	86.2	82.8	84.4	84.3	83.7	85.2	81.3	85.5
	12B IT	**95.1**	**89.0**	95.5	94.5	94.5	95.5	91.2	93.0	96.3	94.4	**95.5**	**95.3**	**95.4**	95.7	**93.7**	94.5
	HEXABEE-4B	91.2	87.4	**96.9**	**96.9**	**96.3**	**97.1**	**94.1**	**97.0**	**96.6**	**94.7**	95.2	94.5	**95.4**	**96.2**	93.5	**95.1**
$zh \to en$	4B IT	65.1	61.5	64.1	65.8	64.9	63.8	62.0	75.5	68.0	65.0	75.0	75.4	65.8	66.6	65.3	76.0
	12B IT	**97.9**	**95.3**	95.9	**96.9**	**96.7**	96.1	**94.4**	95.2	**97.1**	**96.4**	94.8	94.8	95.0	96.2	93.1	94.5
	HEXABEE-4B	97.5	93.6	**96.8**	96.4	**96.7**	**96.7**	94.1	**96.0**	**97.1**	**96.4**	**96.8**	**95.4**	**95.9**	**97.1**	**94.6**	**95.8**
$he \to en$	4B IT	**95.6**	**88.8**	96.4	95.7	**95.6**	97.1	92.6	95.0	**93.2**	**90.7**	95.8	**95.0**	**94.5**	**96.2**	93.6	94.0
	12B IT	92.4	86.3	97.1	97.2	95.1	96.8	93.5	96.0	90.3	86.7	**95.9**	94.7	90.5	95.2	93.2	94.3
	HEXABEE-4B	91.4	88.4	**98.4**	**98.2**	92.7	**98.9**	**95.5**	**98.5**	90.9	89.1	94.9	94.8	91.1	95.9	**93.7**	**96.3**
$fa \to en$	4B IT	97.5	94.0	96.9	97.0	97.3	97.0	94.1	94.0	96.5	95.8	94.7	95.9	96.2	95.5	95.0	95.0
	12B IT	**98.4**	**97.5**	97.6	97.7	98.7	97.7	**95.1**	95.8	97.8	96.4	95.6	96.0	96.9	96.7	94.7	94.5
	HEXABEE-4B	97.9	97.3	**99.2**	**97.9**	**98.9**	**99.3**	**95.1**	**98.0**	**98.2**	**97.7**	**96.7**	**96.7**	**97.4**	**97.4**	**96.1**	**95.7**
$es \to en$	4B IT	99.1	96.3	**97.8**	98.3	98.4	**98.2**	95.1	96.7	98.3	98.6	97.3	97.3	97.9	97.5	96.6	96.0
	12B IT	**99.7**	**98.9**	97.0	**99.3**	**99.6**	97.8	**95.5**	97.9	**99.0**	**99.0**	97.3	97.6	**98.7**	97.3	**96.8**	96.3
	HEXABEE-4B	96.2	95.0	97.5	97.4	96.6	97.6	93.6	**98.0**	97.3	96.4	**97.5**	**97.7**	97.1	**97.6**	96.6	**97.5**

Table 20: Every judge dimension for translations into English, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Dir.	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
$ar \to he$	4B IT	72.5	64.0	72.0	72.5	72.0	72.0	68.5	83.0	69.4	62.5	69.5	69.9	69.2	70.0	66.6	69.5
	12B IT	96.0	86.3	94.5	94.4	95.0	95.0	90.0	94.5	96.3	92.9	95.2	93.9	94.1	94.5	92.3	92.1
	HEXABEE-4B	**96.3**	**88.5**	**95.6**	**96.2**	**96.2**	**95.6**	**93.6**	**97.5**	**97.6**	**94.8**	**95.8**	**94.9**	**96.5**	**95.9**	**93.8**	**96.5**
$zh \to he$	4B IT	89.0	80.8	87.7	88.7	88.7	87.4	86.0	88.0	90.4	84.8	87.5	87.9	89.0	89.1	86.5	90.0
	12B IT	93.3	86.8	92.9	91.8	92.3	92.9	90.1	91.2	92.7	87.2	89.0	91.2	88.7	89.5	86.9	91.0
	HEXABEE-4B	**97.4**	**90.5**	**95.9**	**95.6**	**96.0**	**96.5**	**93.3**	**97.3**	**96.4**	**94.8**	**95.0**	**95.3**	**95.1**	**95.6**	**93.2**	**96.0**
$en \to he$	4B IT	72.0	62.5	79.5	80.0	74.5	78.5	72.5	85.5	82.2	69.5	79.5	81.7	80.0	80.0	78.5	87.0
	12B IT	96.6	88.1	94.8	94.8	95.9	93.8	92.1	93.0	95.0	91.1	91.6	**93.2**	93.3	92.4	90.9	93.5
	HEXABEE-4B	**96.9**	**91.2**	**95.6**	**95.7**	**96.6**	**95.5**	**92.4**	**97.0**	**95.6**	**92.5**	**92.8**	93.1	**93.9**	**93.8**	**91.8**	**95.5**
$fa \to he$	4B IT	72.0	61.5	76.3	77.5	71.5	76.3	72.5	90.5	79.0	70.8	78.6	79.2	78.1	80.5	76.1	82.0
	12B IT	95.1	84.0	94.6	94.2	92.3	94.1	**91.3**	**95.5**	94.7	90.2	92.4	91.7	92.6	92.5	90.8	94.0
	HEXABEE-4B	**95.4**	**89.0**	**95.3**	**94.4**	**94.9**	**95.5**	91.1	95.1	**96.6**	**94.9**	**94.6**	**94.7**	**95.9**	**95.7**	**92.5**	**95.6**
$es \to he$	4B IT	88.3	78.7	85.7	87.6	86.1	84.7	83.5	93.0	88.7	82.2	87.2	88.8	88.2	87.6	86.7	93.0
	12B IT	94.0	86.5	94.2	93.1	93.8	94.2	90.4	94.0	94.2	90.0	93.4	92.6	92.3	93.1	90.8	93.0
	HEXABEE-4B	**97.2**	**92.2**	**96.0**	**95.2**	**95.1**	**95.7**	**93.0**	**97.5**	**97.3**	**95.3**	**95.3**	**95.2**	**95.8**	**95.5**	**93.6**	**96.0**

Table 21: Every judge dimension for translations into Hebrew, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Dir.	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
$ar \to fa$	4B IT	77.0	68.8	73.4	74.8	75.5	75.3	71.0	78.0	77.6	74.1	76.0	76.3	76.2	76.6	74.3	75.5
	12B IT	**93.2**	85.4	94.1	93.0	91.1	94.1	89.3	94.5	95.4	91.0	95.0	93.5	94.1	94.8	91.8	94.0
	HEXABEE-4B	92.2	**85.7**	**96.4**	**94.5**	**94.6**	**96.4**	**92.0**	**97.7**	**97.0**	**95.2**	**96.5**	**96.4**	**96.8**	**96.1**	**95.3**	**96.5**
$zh \to fa$	4B IT	51.5	45.0	53.0	51.0	51.5	51.5	49.0	64.5	55.0	49.5	55.0	55.0	55.5	54.0	53.0	56.5
	12B IT	91.4	84.8	93.3	92.4	92.3	93.0	90.0	93.0	94.3	89.8	92.3	93.1	93.6	93.6	90.8	92.5
	HEXABEE-4B	**93.6**	**86.4**	**95.1**	**93.1**	**94.3**	**95.4**	**92.9**	**94.5**	**95.2**	**92.6**	**94.5**	**93.8**	**94.6**	**94.9**	**92.8**	**96.0**
$en \to fa$	4B IT	94.1	85.7	92.6	92.9	91.0	91.4	88.4	92.0	93.2	87.2	88.7	91.6	90.1	89.4	87.3	91.5
	12B IT	95.4	88.0	93.4	93.8	94.0	93.4	90.7	92.5	95.1	92.3	91.5	92.6	92.7	92.8	89.4	93.3
	HEXABEE-4B	**96.7**	**90.3**	**94.4**	**95.2**	**95.2**	**94.3**	**91.8**	**93.5**	**96.3**	**94.5**	**93.4**	**94.0**	**94.1**	**94.1**	**92.1**	**94.0**
$he \to fa$	4B IT	94.8	84.9	92.8	93.3	94.5	93.1	90.2	93.5	94.7	90.9	93.1	93.4	93.4	93.1	91.4	93.5
	12B IT	97.0	89.4	95.5	95.8	96.1	95.6	93.0	94.5	96.5	94.3	94.6	95.5	96.0	94.8	94.4	94.5
	HEXABEE-4B	**97.7**	**90.9**	**97.5**	**96.9**	**96.7**	**97.3**	**94.3**	**96.9**	**98.0**	**96.9**	**96.3**	**95.7**	**96.8**	**96.5**	**94.5**	**95.0**
$es \to fa$	4B IT	82.1	74.4	80.2	81.2	81.4	80.7	80.0	80.5	84.8	79.4	81.9	83.6	82.7	81.9	80.4	83.0
	12B IT	95.5	86.7	93.6	93.7	93.9	92.7	90.8	91.0	95.4	91.8	91.8	**93.5**	94.1	92.1	91.2	94.0
	HEXABEE-4B	**96.5**	**88.2**	**94.3**	**95.4**	**95.6**	**94.7**	**92.9**	**96.0**	**96.2**	**94.4**	**93.5**	93.4	**94.8**	**94.7**	**92.2**	**95.0**

Table 22: Every judge dimension for translations into Persian, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Dir.	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
$ar \to es$	4B IT	91.8	80.7	94.1	92.2	92.3	93.0	89.1	92.5	94.4	87.8	94.3	94.0	93.4	93.5	92.5	95.8
	12B IT	**97.4**	**89.5**	**98.0**	**97.1**	**96.9**	**97.3**	**93.9**	93.0	**98.5**	**96.5**	96.6	**96.7**	**97.5**	96.6	**95.9**	96.0
	HEXABEE-4B	95.0	85.5	96.8	95.8	96.1	96.1	93.8	**96.8**	96.2	91.6	**97.1**	95.7	95.3	**97.1**	94.3	**96.1**
$zh \to es$	4B IT	86.1	78.0	85.0	84.7	85.4	84.9	81.7	83.5	87.6	87.3	85.6	86.4	86.7	86.4	84.6	85.5
	12B IT	96.4	90.6	95.7	95.1	95.5	95.5	92.0	93.5	95.8	93.3	92.9	93.6	93.2	93.3	91.5	93.6
	HEXABEE-4B	**97.5**	**91.4**	**96.6**	**95.2**	**96.8**	**96.3**	**94.3**	**96.5**	**96.5**	**95.7**	**94.8**	**95.1**	**95.1**	**95.9**	**94.0**	**94.8**
$en \to es$	4B IT	96.9	89.8	95.5	95.8	95.7	94.6	92.9	93.5	96.9	95.6	95.3	95.3	96.6	95.8	94.4	95.0
	12B IT	**98.2**	90.7	**97.3**	97.2	**97.2**	96.4	**94.6**	94.5	**98.3**	**97.3**	95.5	**97.0**	**97.5**	96.2	95.2	**97.5**
	HEXABEE-4B	97.3	**91.6**	96.9	**97.3**	97.1	**96.8**	94.4	**97.5**	97.5	96.4	**96.9**	96.2	96.6	**97.0**	**95.6**	96.2
$he \to es$	4B IT	80.6	75.0	79.7	80.1	79.6	79.3	76.9	95.5	78.2	76.8	77.7	77.7	77.3	78.2	75.9	80.5
	12B IT	**99.2**	93.6	**98.9**	**98.5**	98.6	98.3	**95.7**	95.7	87.8	87.2	86.8	86.8	87.5	86.7	86.1	85.5
	HEXABEE-4B	99.1	**97.5**	98.3	98.2	**99.0**	**98.9**	94.6	**99.0**	**98.1**	**98.0**	**98.2**	**97.6**	**98.1**	**98.4**	**95.8**	**95.8**
$fa \to es$	4B IT	84.6	81.2	94.5	93.2	88.7	93.9	87.3	92.5	92.5	90.9	94.8	92.9	92.4	94.9	**91.6**	91.0
	12B IT	97.5	94.0	96.3	96.8	95.5	96.5	92.5	94.0	95.0	92.6	**95.3**	93.3	92.4	**95.5**	91.5	91.0
	HEXABEE-4B	**98.1**	**94.2**	**96.4**	**97.0**	**97.1**	**97.7**	**93.3**	**96.1**	**96.2**	**93.8**	94.0	**94.1**	**93.1**	95.1	**91.6**	**94.5**

Table 23: Every judge dimension for translations into Spanish, ten shared items per direction. Sem.: semantic accuracy; Term.: terminology; Flu.: fluency; Coh.: cohesion; Ctx.: contextual appropriateness; Lang.: target-language adherence; Fmt.: formatting. Scores are model judgments on 0–100 scales, not validated accuracy percentages. Bold/underline mark first/second within each direction and view; ties at displayed precision share rank.

		Reference-assisted judge								Reference-free judge							
Target	**System**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**	**Sem.**	**Term.**	**Flu.**	**Coh.**	**Ctx.**	**Lang.**	**Style**	**Fmt.**
ar	4B IT	85.1	77.1	85.5	85.3	84.3	84.9	81.6	87.0	84.4	78.3	83.6	84.0	83.2	84.0	81.5	85.5
	12B IT	92.8	84.3	94.5	93.9	91.5	92.0	89.0	92.9	94.3	90.6	92.2	92.3	92.9	90.9	91.2	93.1
	HEXABEE-4B	**95.7**	**89.8**	**96.6**	**95.4**	**94.8**	**96.1**	**92.6**	**96.4**	**95.5**	**93.4**	**94.4**	**94.3**	**94.5**	**95.0**	**92.3**	**94.6**
zh	4B IT	78.1	71.2	76.2	76.4	76.8	76.8	73.7	77.8	82.5	77.0	83.3	83.2	79.5	77.8	77.5	83.4
	12B IT	92.9	88.0	91.3	91.3	91.8	91.4	88.2	90.5	93.5	90.5	92.0	91.4	92.2	92.7	90.2	91.1
	HEXABEE-4B	**94.6**	**89.8**	**94.4**	**93.2**	**93.7**	**94.4**	**91.2**	**94.2**	**94.4**	**92.1**	**94.3**	**92.9**	**93.2**	**94.4**	**91.6**	**93.5**
en	4B IT	88.1	83.1	88.0	88.2	88.0	88.2	84.9	89.5	88.4	86.6	89.4	89.6	87.6	88.2	86.4	89.3
	12B IT	**96.7**	**93.4**	96.6	97.1	**96.9**	96.8	93.9	95.6	**96.1**	94.6	95.8	95.7	95.3	96.2	94.3	94.8
	HEXABEE-4B	94.8	92.3	**97.8**	**97.4**	96.2	**97.9**	**94.5**	**97.5**	96.0	**94.9**	**96.2**	**95.8**	**95.4**	**96.8**	**94.9**	**96.1**
he	4B IT	78.8	69.5	80.2	81.3	78.6	79.8	76.6	88.0	81.9	74.0	80.5	81.5	80.9	81.4	78.9	84.3
	12B IT	95.0	86.3	94.2	93.7	93.9	94.0	90.8	93.6	94.6	90.3	92.3	92.5	92.2	92.4	90.3	92.7
	HEXABEE-4B	**96.6**	**90.3**	**95.7**	**95.4**	**95.8**	**95.8**	**92.7**	**96.9**	**96.7**	**94.5**	**94.7**	**94.6**	**95.4**	**95.3**	**93.0**	**95.9**
fa	4B IT	79.9	71.8	78.4	78.6	78.8	78.4	75.7	81.7	81.1	76.2	78.9	80.0	79.6	79.0	77.3	80.0
	12B IT	94.5	86.9	94.0	93.7	93.5	93.8	90.8	93.1	95.3	91.8	93.0	93.6	94.1	93.6	91.5	93.7
	HEXABEE-4B	**95.3**	**88.3**	**95.5**	**95.0**	**95.3**	**95.6**	**92.8**	**95.7**	**96.5**	**94.7**	**94.8**	**94.7**	**95.4**	**95.3**	**93.4**	**95.3**
es	4B IT	88.0	80.9	89.8	89.2	88.3	89.1	85.6	91.5	89.9	87.7	89.5	89.3	89.3	89.8	87.8	89.6
	12B IT	**97.7**	91.7	**97.2**	**96.9**	96.7	96.8	93.7	94.1	95.1	93.4	93.4	93.5	93.6	93.7	92.0	92.7
	HEXABEE-4B	97.4	**92.0**	97.0	96.7	**97.2**	**97.2**	**94.1**	**97.2**	**96.9**	**95.1**	**96.2**	**95.7**	**95.6**	**96.7**	**94.3**	**95.5**

Table 24: All eight judge dimensions in both views, aggregated by target language, with 50 shared examples per target. All directions receive equal weight.

Split	System	N	Src.	Cand.	Ref.
Full	4B IT	36,572	7757	8240	7789
Full	12B IT	36,572	7757	8334	7789
Full	HEXABEE-4B	36,572	7757	7743	7789
Judge	4B IT	300	51	52	52
Judge	12B IT	300	51	53	52
Judge	HEXABEE-4B	300	51	51	52

Table 25: Texts exceeding the 512-token COMET encoder limit. Src., Cand., and Ref. denote source, candidate, and reference. Counts can overlap within an example.

Metric settings. SacreBLEU 2.5.1: one reference, mixed-case BLEU, exponential smoothing, no effective order, `zh` tokenization for Chinese targets and `13a` otherwise. chrF and chrF++ use character order six, $\beta = 2$, and respectively zero/two word orders, excluding character whitespace. TER uses lowercase Tercom tokenization with normalization, punctuation retained, and Asian support. The compiled TER beam kernel preserves the reference operation tie order, beam, shift limits, and prefix cache; exact edit/length totals match SacreBLEU on all 900 judge-subset outputs, alongside 2,000 exact score/trace cache checks and 250 candidate-order checks. The compiled complete search additionally matches 500 randomized cases; the reference tokenizer and search semantics are retained.

COMET. We use `Unbabel/wmt22-comet-da` with unbabel-comet 2.2.7, official checkpoint revision `2760a223ac95`, and verified SHA-256. Inference uses float32 (batch 32 for full-test scoring, 16 for the judge subset). Values in tables are raw scores multiplied by 100 and are not accuracy percentages. The encoder truncates each source, candidate, and reference separately at 512 tokens, so long-document suffixes are not assessed by this metric.

G Evaluation integrity and reproducibility

The parser used truth-value fallback for the score field, which turned some explicit zero scores into nulls in intermediate records. We reparse the original judge JSON and recover 11 reference-view and 13 reference-free zeros for the 4B baseline, plus one reference-free zero for the 12B baseline. All 300 items then have valid scores in both views. Dropping these entries would inflate baseline means. The reparsed means agree with the final flattened score files.

Content identity is the SHA-256 hash of a canonical JSON list containing source language, target language, source text, and reference. We require equality of identity sets before comparing full-test files and before pairing judge scores. File row indices are not stable identities: asynchronous processing and sampling changed order across models. For judge recovery, an explicit JSON field `score: 0` remains zero rather than being treated as a missing value.

Our audit scripts are read-only with respect to the original training and evaluation artifacts. They preserve null/empty outputs, pin metric settings, save directional sufficient statistics where applicable, and produce content-hash IDs without copying raw evaluation text into the paper package. The local evidence directory records the actual training/merge paths and sanitized snapshots for author verification; those private paths are omitted from this release. The paper reports a retrospective analysis of existing translation outputs and newly computed metrics, not new training runs or newly collected human labels.

SerialDistill: Retrieval-Conditioned On-Policy Distillation for Data-Scarce Serial Subtitle Translation

Yiran Zhang[1] **Ziqi Zhang**[2]
[1]Tianjin University of Finance & Economics, Tianjin, China
[2]Harbin Institute of Technology, Harbin, China
[1]shiro027@stu.tjufe.edu.cn [2]2023110966@stu.hit.edu.cn

Abstract

Serial subtitle translation requires work-specific terminology and translation conventions to be maintained across episodes, despite scarce bilingual supervision. Previously finalized subtitles provide both adaptation data and translation memory, but our experiments show that the benefit of combining supervised adaptation and retrieval varies across series. We introduce **SerialDistill**, which combines LoRA-based domain adaptation with retrieval-conditioned on-policy distributional distillation (OPD). A Qwen3-8B student generates translation trajectories, while a frozen Qwen3-14B teacher provides token-level supervision under the same bilingual context retrieved exclusively from previously finalized episodes. On Japanese-to-Chinese translation across two anime series, SerialDistill improves over the non-distilled LoRA-8B+RAG baseline by up to **3.80 chrF and 4.15 BLEU**, measured on pooled held-out episodes. The student also outperforms the larger retrieval-augmented teacher, which has not undergone domain adaptation. These results support retrieval-conditioned distillation as an additional adaptation stage for data-scarce serial subtitle translation.

1 Introduction

Serial subtitle translation is a data-scarce domain-adaptation task: only a few thousand work-specific bilingual subtitle pairs are available in our setting, yet translation must remain consistent in character names, terminology, and recurring expressions across episodes. As episodes are finalized, their subtitles form a growing translation memory that is limited in quantity but closely matched to subsequent episodes. We therefore ask how a compact model can effectively exploit this scarce supervision and historical bilingual context for Japanese-to-Chinese subtitle translation.

Supervised adaptation can encode work-specific patterns through low-rank adaptation (LoRA) (Hu et al., 2022), while retrieval-augmented generation (RAG) makes earlier translations explicitly available during generation. However, the gains from their combination vary across series. Combining LoRA and RAG improves translation on the primary series, but does not consistently outperform either component alone on a second series. This motivates investigating whether teacher supervision can further improve a domain-adapted student operating with historical bilingual context.

To address this question, we develop **SerialDistill**, which adapts on-policy distributional distillation (Agarwal et al., 2024) to serial subtitle translation. As illustrated in Figure 1, SerialDistill conditions both student generation and teacher supervision on episode-causal bilingual memory retrieved exclusively from previously finalized episodes. A LoRA-adapted student generates translation trajectories under this historical context, and a frozen teacher provides next-token distributions along the same student-generated prefixes under identical context. This couples the distillation process to the retrieval conditions used at inference time.

We evaluate SerialDistill on two anime series using a Qwen3-8B student and a Qwen3-14B teacher (Yang et al., 2025). On the pooled held-out sets, SerialDistill improves over LoRA-8B+RAG and achieves the highest chrF, BLEU, and COMET scores among the evaluated systems on both series. These results support the additional value of retrieval-conditioned distillation.

Our contributions are threefold:

- We study data-scarce serial subtitle translation with bilingual memory restricted to previously finalized episodes.
- We introduce SerialDistill, combining domain adaptation with retrieval-conditioned on-policy distributional distillation.
- We demonstrate improvements over adapta-

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 71–83
November 9, 2026 ©2026 Association for Computational Linguistics

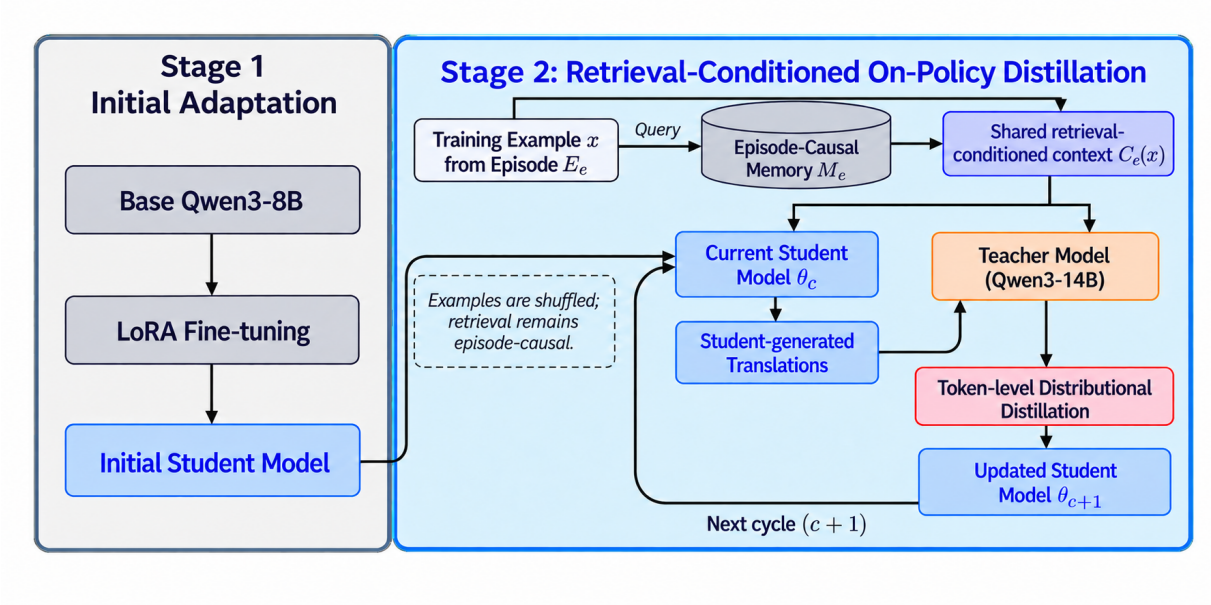


Figure 1: Overview of SerialDistill. Qwen3-8B is first adapted with LoRA, then refined through retrieval-conditioned on-policy distributional distillation. Training examples are shuffled, while each instance accesses only episode-causal memory from previously finalized episodes.

tion, retrieval, and their direct combination on the pooled held-out sets of two anime series.

2 Related Work

2.1 Large Language Models for Machine Translation

Large language models (LLMs) have been evaluated for translation across language pairs and prompting settings (Hendy et al., 2023; Jiao et al., 2023). For few-shot translation, example quality and domain relevance influence performance (Vilar et al., 2023; Agrawal et al., 2023). The use of contextual information has been explored for document-level and subtitle translation (Wang et al., 2023; Pramodya et al., 2025). However, effectively exploiting broader context remains challenging (Jin et al., 2023). Studies of subtitle translation have also examined context-dependent phenomena such as deixis, ellipsis, and lexical cohesion (Voita et al., 2019). Our setting is a domain-specific, low-resource case in which earlier subtitles remain directly relevant to later episodes.

2.2 Domain Adaptation and Parameter-Efficient Fine-Tuning

Fine-tuning on in-domain data is a standard approach to domain adaptation in neural machine translation (NMT) (Saunders, 2022). Parameter-efficient alternatives include lightweight adapter modules for pretrained language models and translation systems (Houlsby et al., 2019; Bapna and Firat, 2019b). LoRA represents parameter updates with low-rank matrices (Hu et al., 2022). In subtitle translation, LoRA can internalize recurring terminology, conventions, and style; we pair this parameter adaptation with retrieval to preserve access to historical translations.

2.3 Retrieval-Augmented Machine Translation

Retrieval-augmented machine translation supplements a model with relevant bilingual examples or external knowledge at inference time; earlier subtitles can therefore serve as a translation memory. Earlier retrieval-based MT methods reuse translation pieces or retrieve phrase-level context from bilingual memory (Zhang et al., 2018; Bapna and Firat, 2019a). At the token level, kNN-MT augments predictions with a datastore of translation examples and supports domain adaptation through the choice of datastore (Khandelwal et al., 2021). More recent work has studied retrieval for machine translation in semantic, document-level, and external-knowledge settings (Bouthors et al., 2024; Peng et al., 2025; Wang et al., 2025; Lu et al., 2025). Serial subtitles are a natural test bed because finalized episodes often reuse terminology and expressions. Our work uses this translation memory to condition both teacher and student during distillation.

2.4 Knowledge Distillation and On-Policy Distillation

Knowledge distillation transfers information from a stronger teacher through generated sequences or predictive distributions (Hinton et al., 2015). Sequence-level distillation trains on teacher translations (Kim and Rush, 2016), whereas token-level methods preserve richer distributional information. Recent LLM distillation methods also study the choice of distribution-matching objectives and the use of student-generated outputs (Gu et al., 2024; Ko et al., 2024). Offline methods typically supervise fixed teacher trajectories; on-policy distillation instead lets the student generate trajectories and obtains supervision on the states it visits (Agarwal et al., 2024). Our work applies on-policy distributional distillation to data-scarce serial subtitle translation. We combine a domain-adapted student with bilingual retrieval restricted to previously finalized episodes, using the same retrieved context for teacher and student.

3 Method

3.1 Task Definition and Overview

Given a Japanese subtitle sequence

$$X = (x_1, \ldots, x_n), \tag{1}$$

the model generates the corresponding Chinese subtitle sequence

$$Y = (y_1, \ldots, y_n). \tag{2}$$

Here, x_i and y_i denote aligned subtitle segments. Within an individual target segment y, y_t denotes its t-th token.

We consider a serial translation setting in which each training instance is associated with an episode-causal retrieval memory. When translating an example from episode e, the retrieval memory contains only finalized Japanese–Chinese subtitle pairs from episodes $1, \ldots, e-1$. Target subtitles from the current and future episodes are excluded from retrieval. Source-side subtitles from future episodes are also excluded from retrieval in this evaluated setting.

Our framework, **SerialDistill**, consists of two stages. First, Qwen3-8B is adapted to the target series through supervised fine-tuning (SFT) with LoRA. Second, the resulting domain-adapted student is further refined through retrieval-conditioned on-policy distributional distillation from a frozen Qwen3-14B teacher.

This episode-causal construction aligns retrieval-conditioned distillation with the assumed serial translation workflow: the retrieved bilingual evidence excludes the current and future episodes. This restriction applies to retrieval memory; it does not impose an episode-by-episode order on parameter updates.

3.2 Domain Adaptation with LoRA

We first adapt Qwen3-8B using the available in-domain bilingual subtitles. LoRA is used for parameter-efficient supervised fine-tuning, while the underlying pretrained model remains frozen.

The student is optimized with the standard autoregressive response-token objective:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t} \log p_\theta(y_t \mid y_{<t}, x). \tag{3}$$

This stage adapts the student to work-specific terminology, translation conventions, and recurring linguistic patterns. The resulting Qwen3-8B LoRA model serves as the initialization for the subsequent distillation stage.

3.3 Episode-Causal Subtitle Memory

Let the finalized bilingual subtitle set for episode j be

$$D_j = \{(x_i^{(j)}, y_i^{(j)})\}_{i=1}^{N_j}, \tag{4}$$

where $x_i^{(j)}$ and $y_i^{(j)}$ denote the Japanese source and finalized Chinese translation of subtitle i.

When processing episode e, the available translation memory is

$$M_e = \bigcup_{j<e} D_j. \tag{5}$$

Only subtitle pairs originating from episodes earlier than e are eligible for retrieval; pairs from the current and future episodes are excluded.

For a current Japanese subtitle x, the retriever returns up to K highest-scoring bilingual matches from M_e:

$$R_e(x) = \mathrm{TopK}_{(x',y')\in M_e} \mathrm{BM25}(x, x'). \tag{6}$$

The retrieved bilingual examples and the current source subtitle are then combined into a single model context:

$$C_e(x) = \mathrm{Prompt}\left(R_e(x), x\right). \tag{7}$$

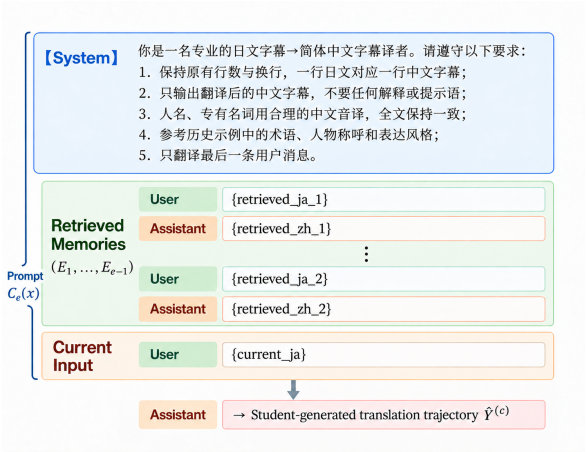


Figure 2: Role-structured construction of the episode-causal retrieval prompt. Retrieved bilingual examples are represented as User–Assistant dialogue turns, followed by the current Japanese input. A fixed bilingual demonstration is omitted for clarity.

Figure 2 illustrates the role-structured prompt construction used to instantiate $C_e(x)$.

This construction makes retrieval strictly causal with respect to episode order while allowing the bilingual memory to grow naturally as more episodes become historical.

Training instances are shuffled for OPD; episode order constrains retrieval availability rather than optimization order.

3.4 Retrieval Conditioning

SerialDistill uses a frozen Qwen3-14B model as the teacher and the LoRA-adapted Qwen3-8B model as the student.

During distillation, both models receive the same current source subtitle and the same causal retrieval context. Their predictive distributions are therefore

$$p_T\left(y_t \mid C_e(x), y_{<t}\right) \tag{8}$$

for the teacher, and

$$p_\theta\left(y_t \mid C_e(x), y_{<t}\right) \tag{9}$$

for the student.

The teacher contributes the predictive behavior of the larger model, while the student begins with work-specific knowledge already acquired through SFT. The teacher and student operate under identical retrieval evidence, so teacher supervision is conditioned on information that is also available to the student.

3.5 Retrieval-Conditioned Distillation

Figure 3 summarizes the distillation cycle. Let θ_c denote the student parameters and B_c the batch of training examples at the beginning of cycle c. For each example $x \in B_c$, we first construct its episode-causal retrieval context $C_e(x)$. The current student then samples a fresh translation trajectory

$$\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim \pi_{\theta_c}^{\text{rollout}}(\cdot \mid C_e(x)). \tag{10}$$

Here, $\pi_{\theta_c}^{\text{rollout}}$ denotes the current student's sampling policy with temperature 0.7 and top-p 0.9. Teacher and student distributions used in the KL objective are computed at temperature 1 over the full vocabulary, without nucleus truncation. Generated token IDs and teacher targets are held fixed during backpropagation.

The frozen teacher evaluates the same student-generated prefixes under the identical retrieval context:

$$p_T^t(v) = p_T(v \mid C_e(x), \hat{y}_{<t}), \tag{11}$$

$$p_{\theta_c}^t(v) = p_{\theta_c}(v \mid C_e(x), \hat{y}_{<t}). \tag{12}$$

At each response position t, let S_t denote the teacher's top-K_T token set. Both models are compressed using the same set. For $m \in \{T, \theta_c\}$, we define a distribution over $S_t \cup \{\text{other}\}$:

$$q_m^t(v) = \begin{cases} p_m^t(v), & v \in S_t, \\ 1 - \sum_{u \in S_t} p_m^t(u), & v = \text{other}. \end{cases} \tag{13}$$

The retained probabilities are not renormalized; the remaining probability mass forms the residual category.

The loss for cycle c is computed as the token-weighted mean KL divergence over all valid response tokens in the cycle:

$$\mathcal{L}_c = \frac{\sum_{x \in B_c} \sum_{t \in \mathcal{T}_x} D_{\text{KL}}\left(q_T^{x,t} \,\|\, q_{\theta_c}^{x,t}\right)}{\sum_{x \in B_c} |\mathcal{T}_x|}, \tag{14}$$

where $\mathcal{T}_x$ contains response positions whose generated token IDs are not special tokens. Prompt, padding, and special-token positions, including EOS, are excluded from the loss.

Gradients are accumulated over the cycle and normalized by its total number of valid response tokens. After gradient clipping, we perform one AdamW update of the student's LoRA parameters, yielding θ_{c+1}. The student backbone and teacher

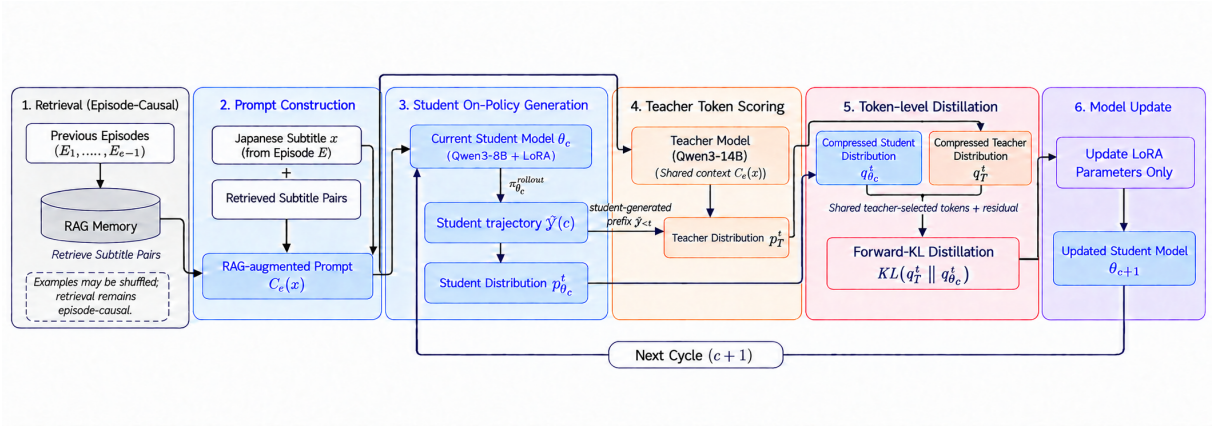


Figure 3: Overview of retrieval-conditioned on-policy distillation. Student and teacher share the same episode-causal RAG context; the teacher provides token-level supervision along student-generated trajectories, and the updated student generates fresh trajectories in the next optimization cycle.

remain frozen, and optimizer state is preserved across cycles.

After this update, the next cycle processes a new batch of training examples and generates fresh trajectories using the updated student $p_{\theta_{c+1}}$. No replay buffer or fixed teacher-generated trajectory set is used.

SerialDistill is therefore on-policy at the cycle level: each batch is generated by the current student, while retrieval remains episode-causal for every training instance.

4 Experimental Setup

4.1 Data and Splits

We use Japanese–Chinese subtitles from a 12-episode anime series. Lyric cues are excluded from the primary-series experiments. Dataset usage is summarized in Table 1.

OPD examples from EP2–9 are shuffled across episodes, with retrieval restricted to earlier finalized episodes for each example. At evaluation time, the memory grows with episode order: EP10, EP11, and EP12 use finalized subtitles from EP1–9, EP1–10, and EP1–11, respectively. Earlier evaluation episodes enter the bilingual memory only after they are treated as finalized, and no further parameter updates are performed. For each evaluated episode, reference translations from that episode and all future episodes are excluded from its retrieval context. EP11–12 are not used for adaptation, distillation, or configuration selection. These boundaries prevent target-episode reference leakage while permitting reuse of previously finalized translations in the sequential workflow.

Use	Episodes	Pairs
SFT / initial memory	1–9	2,687
OPD	2–9	2,448
Development	10	346
Held-out test	11 / 12	277 / 212

Table 1: Primary-series episode split and retained non-lyric pairs.

System	Base model	LoRA	RAG	Distillation
Base-8B	Qwen3-8B	–	–	–
LoRA-8B	Qwen3-8B	✓	–	–
RAG-8B	Qwen3-8B	–	✓	–
LoRA-8B+RAG	Qwen3-8B	✓	✓	–
14B+RAG Teacher	Qwen3-14B	–	✓	–
SerialDistill	Qwen3-8B	✓	✓	OPD

Table 2: Summary of evaluated systems.

Cross-series replication. To examine whether the main finding transfers beyond the primary series, we independently replicate the SerialDistill pipeline on a second 12-episode anime series, *Gimai Seikatsu*, using publicly available Japanese–Chinese subtitles released by Haruhana Funsub. The 12-episode dataset follows the same split protocol as the primary experiment: Episodes 1–9 provide LoRA adaptation and retrieval memory, EP2–9 provide OPD training instances, EP10 is the development set, and EP11–12 are held-out tests. Held-out episodes are evaluated only after training and configuration are fixed.

4.2 Models and Systems

The evaluated systems are summarized in Table 2.

The Qwen3-14B teacher remains frozen, and SerialDistill is initialized from the domain-adapted

Qwen3-8B LoRA model. LoRA-8B+RAG uses the same initial adapter without distillation. Both series use this student–teacher configuration.

4.3 Training Configuration

The primary-series experiments are run on a single GPU. We first adapt the student to the target series with LoRA and then refine the same adapter through retrieval-conditioned on-policy distillation. During distillation, each optimization cycle uses fresh student-generated trajectories, while the student backbone and teacher remain frozen. Hardware, software versions, and training hyperparameters are reported in Appendix A.3.

4.4 Retrieval and Inference

Both series use sentence-level BM25 and the episode-causal memory protocol described above. The number of retrieved examples is specified in Appendix A.3. The shared translation instruction contains one fixed bilingual demonstration in addition to any retrieved examples. We compare dense semantic retrieval and structured word-level memory on the primary-series development set in Section 5.4.

Evaluation uses deterministic decoding with thinking disabled and a maximum of 128 new tokens; OPD training uses sampled trajectories.

4.5 Evaluation

We report corpus chrF (Popović, 2015), BLEU (Papineni et al., 2002) computed with SacreBLEU and Chinese tokenization (Post, 2018), and mean segment COMET using `Unbabel/wmt22-comet-da` (Rei et al., 2020). Major comparisons additionally use two-sided paired bootstrap resampling for significance testing (Koehn, 2004). For the primary-series teacher comparison and both second-series comparisons in Table 5, we use 2,000 paired resamples of subtitle segments with random seed 42.

5 Results

5.1 Main Results

Table 3 summarizes the main results on the EP10 development set and the held-out EP11–12 test sets. On EP10, both LoRA and retrieval improve over the base model, and their combination improves all three metrics over either component alone.

Retrieval-conditioned on-policy distillation further improves over LoRA-8B+RAG by 0.72 chrF and 1.20 BLEU on the pooled held-out set (Table 5). SerialDistill also surpasses the larger 14B+RAG teacher across chrF, BLEU, and COMET on both EP11 and EP12.

The same trend holds on the pooled 489-example test set. SerialDistill exceeds the teacher by 3.60 chrF, 6.80 BLEU, and 0.0067 COMET, with the pooled differences reaching statistical significance (Table 5). This consistency across episodes and metrics indicates that the improvement is not driven by a single test split.

5.2 Cross-Series Replication

To test whether the main SerialDistill result is specific to the primary series, we independently replicate the pipeline on *Gimai Seikatsu*. As shown in Table 4, SerialDistill achieves the strongest pooled performance and preserves the student-over-teacher pattern on both held-out episodes.

On the pooled EP11–12 set, SerialDistill improves over the corresponding non-distilled LoRA-8B+RAG baseline by 3.80 chrF and surpasses the 14B+RAG teacher by 2.46 chrF. All three metric improvements over LoRA-8B+RAG are significant at $p < .001$; against the teacher, chrF and BLEU are significant at $p < .001$ and COMET at $p = .001$. Table 5 summarizes the pooled gains over the non-distilled student and the teacher for both series.

5.3 Teacher Retrieval Ablation

We examine teacher-side retrieval on the primary series. The ablated variant removes retrieved examples only from the teacher input during distillation; student retrieval and the episode-causal evaluation protocol are retained in both variants.

As shown in Table 6, teacher retrieval improves chrF by 1.09 and BLEU by 1.12. The chrF gain is statistically significant ($p = .014$; two-sided paired bootstrap, 2,000 resamples, seed 42). These results support the value of providing historical bilingual evidence to the teacher during distillation.

5.4 Retrieval Strategy Comparison

To assess the choice of retrieval strategy, Table 7 compares sentence-level BM25, dense semantic retrieval, and word-level structured memory using Qwen3-8B without LoRA on the primary-series development set. The difference between BM25 and dense retrieval is not statistically significant, whereas the structured memory does not improve over the simpler sentence-level approach.

System	Development: EP10 chrF	BLEU	COMET	Held-out Test: EP11 chrF	BLEU	COMET	Held-out Test: EP12 chrF	BLEU	COMET	Held-out Test: EP11+12 pooled chrF	BLEU	COMET
Base-8B	24.60	28.43	.8286	23.80	28.43	.8207	23.87	27.62	.8153	23.83	28.07	.8184
LoRA-8B	32.19	37.18	.8569	36.93	42.38	.8753	37.75	42.48	.8708	37.29	42.43	.8733
RAG-8B	33.71	38.26	.8590	38.39	37.70	.8735	34.60	39.32	.8702	36.75	38.39	.8721
LoRA-8B+RAG	35.59	40.58	.8645	43.26	47.32	.8867	38.13	43.48	.8785	40.98	45.61	.8832
14B+RAG Teacher	34.34	37.66	.8630	39.41	40.64	.8806	36.46	39.24	.8754	38.11	40.01	.8784
SerialDistill	**36.94**	**42.48**	**.8723**	**43.42**	**48.00**	**.8883**	**39.55**	**44.78**	**.8809**	**41.70**	**46.81**	**.8851**

Table 3: Main results on EP10 development and EP11/12 held-out tests. The pooled column is computed on the concatenated 489-example held-out set.

System	Development: EP10 chrF	BLEU	COMET	Held-out Test: EP11 chrF	BLEU	COMET	Held-out Test: EP12 chrF	BLEU	COMET	Held-out Test: EP11+12 pooled chrF	BLEU	COMET
Base-8B	16.98	17.99	.7877	19.04	20.94	.8017	18.26	19.86	.7945	18.69	20.45	.7983
LoRA-8B	22.03	25.39	**.8162**	19.87	22.93	.8177	22.50	26.31	.8179	21.03	24.48	.8178
RAG-8B	21.61	23.16	.8066	21.22	23.49	.8086	25.11	28.05	.8215	22.94	25.55	.8146
LoRA-8B+RAG	23.76	**27.33**	.8117	19.56	22.76	.8056	24.35	28.29	.8215	21.68	25.31	.8130
14B+RAG Teacher	22.16	23.02	.8097	22.18	22.99	.8115	24.07	25.54	.8166	23.02	24.12	.8139
SerialDistill	**23.85**	27.08	.8136	**22.70**	**25.83**	**.8183**	**28.99**	**33.64**	**.8306**	**25.48**	**29.46**	**.8240**

Table 4: Cross-series replication on *Gimai Seikatsu*. EP10 is the development set, EP11 and EP12 are held-out test sets, and the pooled column is computed on the concatenated 672-example EP11–12 test set.

These results suggest that increasing retrieval complexity does not necessarily improve translation quality in this setting. Because BM25 achieves comparable downstream performance with a simpler retrieval mechanism, we use sentence-level BM25 in the final SerialDistill system.

5.5 Qualitative Analysis

To complement the quantitative comparisons, Figure 4 presents three illustrative translations from LoRA-8B+RAG and SerialDistill. In (a), SerialDistill recovers the omitted description of the project's scale. In (b), it preserves the source statement about insufficient awareness of a crisis rather than recasting it as the addressee's failure to notice the crisis. In (c), it translates the reference to a battlefield smell without adding the unsupported specification of blood.

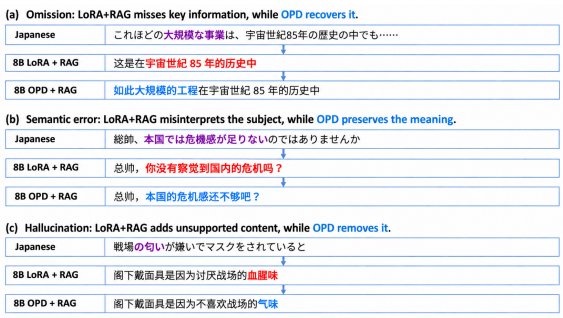


Figure 4: Qualitative examples comparing LoRA-8B+RAG and SerialDistill (labeled OPD in the figure). The examples illustrate omission, semantic error, and unsupported additions in the adapted student output, together with the corresponding SerialDistill corrections.

6 Discussion

6.1 Student–Teacher Performance Comparison

On the pooled held-out sets, SerialDistill's advantage over the 14B+RAG teacher is observed under two different starting conditions. On the primary series, LoRA-8B+RAG already outperforms the teacher before distillation and improves further afterward. On *Gimai Seikatsu*, the same baseline starts below the teacher in chrF and COMET, whereas the distilled student surpasses it on all three metrics. In both cases, the student combines prior adaptation to the target series with supervision from the larger teacher, which has not undergone domain-specific fine-tuning.

This pattern suggests that the teacher's own translation scores do not fully capture its usefulness as a source of supervision. In SerialDistill, the teacher provides next-token guidance along the adapted student's generated prefixes under the same retrieved bilingual context. Such guidance may complement the translation patterns acquired during supervised adaptation, helping the student improve beyond both its initial performance and the teacher's independently generated translations.

The examples in Figure 4 illustrate changes accompanying the aggregate gains: recovery of omitted information, preservation of intended meaning, and avoidance of unsupported details.

Series	Comparison baseline	chrF Δ	chrF p	BLEU Δ	BLEU p	COMET Δ	COMET p
Primary	LoRA-8B+RAG	+0.72	$< .001$	+1.20	$< .001$	+.0019	$< .001$
	14B+RAG Teacher	+3.60	$< .001$	+6.80	$< .001$	+.0067	.042
Gimai Seikatsu	LoRA-8B+RAG	+3.80	$< .001$	+4.15	$< .001$	+.0110	$< .001$
	14B+RAG Teacher	+2.46	$< .001$	+5.34	$< .001$	+.0102	.001

Table 5: SerialDistill gains on the pooled EP11–12 test sets (489 and 672 examples for the primary and second series, respectively). Positive Δ indicates improvement over the comparison baseline. The p values are from two-sided paired bootstrap tests. Deltas use unrounded scores where available; the primary LoRA-8B+RAG comparison uses the reported baseline scores.

Teacher retrieval	chrF	BLEU	COMET
Without	40.17	45.06	.8811
With	**41.26**	**46.18**	**.8819**

Table 6: Teacher-retrieval ablation on the pooled primary-series EP11–12 test set. These training runs are independent of the main experiment in Table 3.

Retrieval	chrF	COMET
Sentence BM25	33.71	.8590
Dense top-2	**33.81**	**.8602**
Structured word-level	32.47	.8575

Table 7: Retrieval comparison for Qwen3-8B without LoRA on the primary-series EP10 development set. Bold marks the highest score in each column.

6.2 Implications for Serial Translation

Combining adaptation and retrieval improves over either component alone on the primary-series pooled test set, while its benefits vary across metrics on *Gimai Seikatsu*. Distillation adds gains in both settings, yielding the highest pooled scores on all three metrics in each series.

Previously finalized subtitles support adaptation and retrieval, while teacher supervision enables further training without additional parallel examples. The gains observed with a few thousand training pairs in each series support this approach to data-scarce domain adaptation.

At inference time, the 8B student translates with retrieval alone. The final system retains the model size and inference structure of LoRA-8B+RAG, fitting a workflow that reuses completed subtitles for subsequent episodes.

7 Conclusion

SerialDistill combines LoRA adaptation with retrieval-conditioned on-policy distillation in a serial subtitle translation workflow. On both evaluated Japanese-to-Chinese anime series, it improves over LoRA-8B+RAG and surpasses the unadapted 14B+RAG teacher on pooled held-out chrF, BLEU, and COMET. Previously finalized subtitles supply adaptation data and episode-causal retrieval memory, while teacher guidance adds supervision without additional parallel examples. The resulting 8B student translates with retrieval alone at inference time. These findings support distillation as an additional adaptation stage under the evaluated bilingual-history setting.

Limitations

Our experiments evaluate Japanese-to-Chinese subtitle translation in two anime series, using finalized translations of earlier episodes as retrieval memory. Evaluation across additional language pairs, genres, and translation-memory conditions would help establish the broader applicability of SerialDistill.

Due to time constraints, we have not conducted cold-start experiments in which no previously translated episodes are available for training or retrieval. Our results therefore apply to settings with an existing work-specific bilingual history. Evaluating performance from an initially empty memory as translations accumulate is an important direction for future work.

The evaluation combines automatic MT metrics with qualitative analysis of individual translations; a systematic human evaluation remains to be conducted. Such an evaluation would assess terminology consistency and context-sensitive meaning across episodes, alongside translation adequacy and fluency, and quantify the occurrence of the improvements illustrated by the qualitative examples.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier

Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of ACL*, pages 8857–8873.

Ankur Bapna and Orhan Firat. 2019a. Non-parametric adaptation for neural machine translation. In *Proceedings of NAACL-HLT*, pages 1921–1931.

Ankur Bapna and Orhan Firat. 2019b. Simple, scalable adaptation for neural machine translation. In *Proceedings of EMNLP-IJCNLP*, pages 1538–1548.

Maxime Bouthors, Josep Crego, and François Yvon. 2024. Retrieving examples from memory for retrieval augmented neural machine translation: A systematic comparison. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3022–3039.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*.

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:2302.09210.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*, pages 2790–2799.

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

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is ChatGPT a good translator? yes with GPT-4 as the engine. ArXiv:2301.08745.

Linghao Jin, Jacqueline He, Jonathan May, and Xuezhe Ma. 2023. Challenges in context-aware neural machine translation. In *Proceedings of EMNLP*, pages 15246–15263.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of EMNLP*, pages 1317–1327.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of ICML*, pages 24872–24895.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Yichen Lu, Wei Dai, Jiaen Liu, Ching Wing Kwok, Zongheng Wu, Xudong Xiao, Ao Sun, Sheng Fu, Jianyuan Zhan, Yian Wang, Takatomo Saito, and Sicheng Lai. 2025. ViDove: A translation agent system with multimodal context and memory-augmented reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 228–243.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Ziqian Peng, Rachel Bawden, and François Yvon. 2025. Self-retrieval from distant contexts for document-level machine translation. In *Proceedings of the Tenth Conference on Machine Translation*, pages 220–240.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation*, pages 186–191.

Ashmari Pramodya, Yusuke Sakai, Justin Vasselli, Hidetaka Kamigaito, and Taro Watanabe. 2025. Translating movie subtitles by large language models using movie-meta information. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 315–330.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of EMNLP*, pages 2685–2702.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. ArXiv:2104.06951, version 2.

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 ACL*, pages 15406–15427.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of ACL*, pages 1198–1212.

Jiaan Wang, Fandong Meng, Yingxue Zhang, and Jie Zhou. 2025. Retrieval-augmented machine translation with unstructured knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5858–5871.

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 EMNLP*, pages 16646–16661.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of NAACL-HLT*, pages 1325–1335.

A Appendix

A.1 Detailed Dataset Statistics

Episode	Primary series	*Gimai Seikatsu*	Experimental role
1	239	282	SFT and initial retrieval memory
2	288	423	SFT, OPD, and retrieval memory
3	335	470	SFT, OPD, and retrieval memory
4	305	335	SFT, OPD, and retrieval memory
5	295	313	SFT, OPD, and retrieval memory
6	309	300	SFT, OPD, and retrieval memory
7	252	453	SFT, OPD, and retrieval memory
8	348	354	SFT, OPD, and retrieval memory
9	316	349	SFT, OPD, and retrieval memory
10	346	344	Development
11	277	360	Held-out test
12	212	312	Held-out test

Table 8: Episode-level dataset statistics. Counts are computed from the dataset files and corresponding audit records.

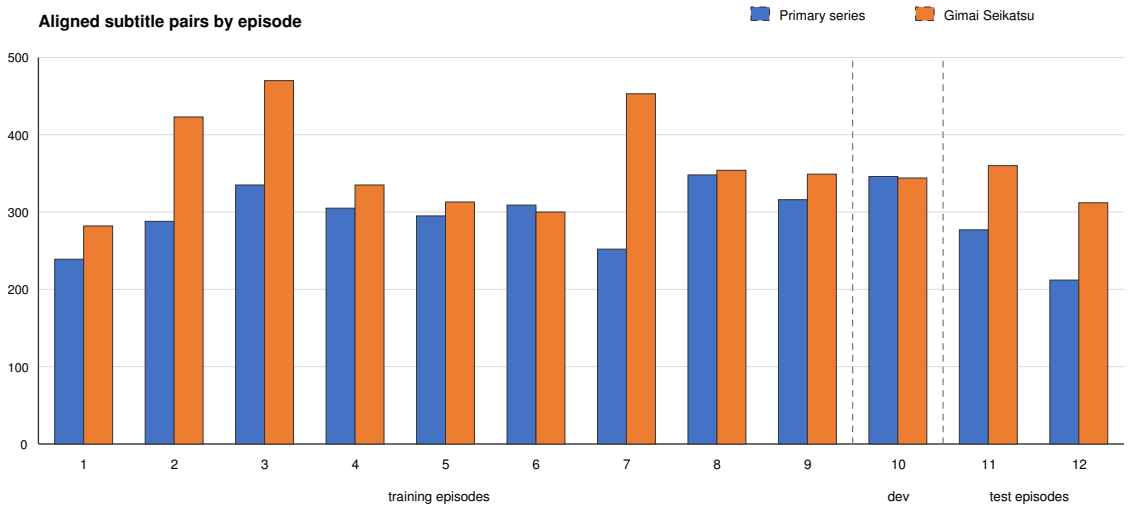


Figure 5: Episode-level distribution of usable aligned subtitle pairs in the two series. Dashed boundaries separate the training, development, and held-out test portions.

This appendix reports the episode-level statistics underlying the two-series experiments. Table 8 gives the exact counts of usable aligned subtitle pairs, and Figure 5 shows how these counts vary across episodes in the two series. For the primary series, the counts exclude lyric cues; for *Gimai Seikatsu*, they correspond to the valid aligned pairs recorded by the dataset audit.

A.2 On-Policy Distillation Diagnostics

Table 9 summarizes the primary-series OPD trace in consecutive cycle ranges, while Figure 6 shows the cycle-level fluctuations in KL and gradient norm. Each row is computed from the saved per-cycle log. KL is token-weighted within each cycle; the table reports arithmetic means of the cycle-level KL values and pre-clipping gradient norms. Coverage is the teacher probability mass retained by its top-64 token set, shared with the student; mean coverage averages the cycle means, and minimum coverage is the lowest token-level value in the range. These diagnostics describe optimization behavior and were not used to select a checkpoint on the held-out test episodes.

Mean KL decreases from 1.4092 in Cycles 1–10 to 0.8537 in Cycles 71–77. At the same time, the retained top-64 set accounts for more than 99.99% of the teacher probability mass on average in every range.

Cycles	Examples	Mean KL	Mean grad. norm	Mean coverage (%)	Min. coverage (%)	Response tokens
1–10	320	1.4092	2.3147	99.9946	96.9318	2,274
11–20	320	1.2840	2.1207	99.9961	97.2676	2,181
21–30	320	1.1872	2.1202	99.9954	98.2997	2,182
31–40	320	1.2077	2.2150	99.9949	94.4799	2,412
41–50	320	1.0819	1.9129	99.9961	98.2002	2,131
51–60	320	1.0099	1.8003	99.9967	98.8558	2,184
61–70	320	0.9189	1.7747	99.9950	95.2910	2,128
71–77	208	0.8537	1.6327	99.9970	98.6739	1,363

Table 9: Primary-series OPD diagnostics aggregated from the 77 saved optimization cycles. The final range contains 208 examples because the last cycle contains the remaining 16 of the 2,448 training instances.

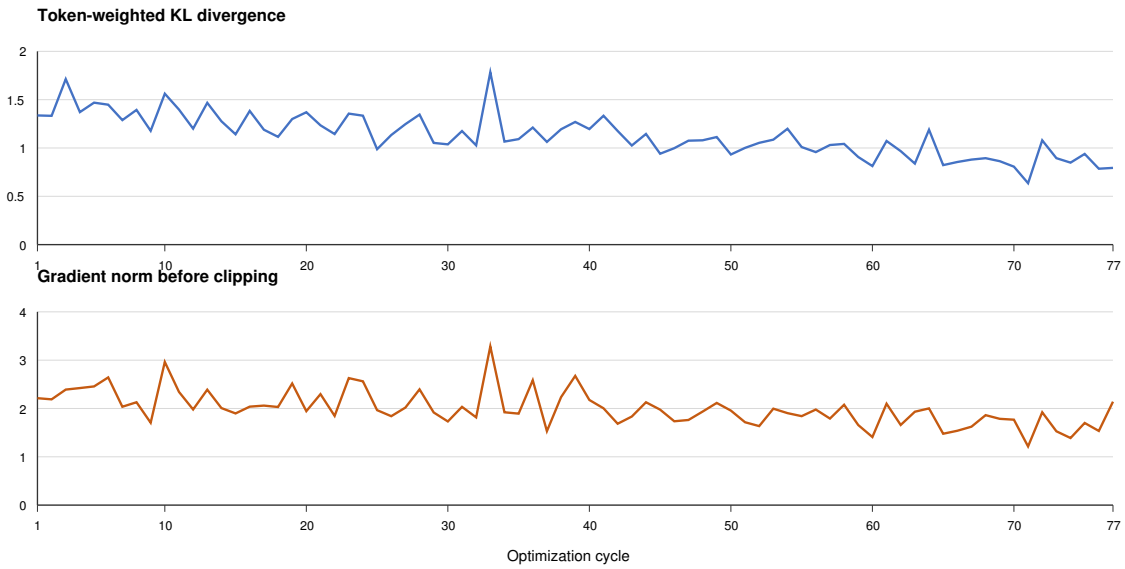


Figure 6: Cycle-level primary-series OPD optimization traces. The upper panel shows the token-weighted KL objective; the lower panel shows the gradient norm before clipping.

A.3 Training and Retrieval Configuration

Table 10 lists the hardware, software, and training settings for the primary series, together with the retrieval setting shared by both series. Dataset sizes and splits are given in Tables 1 and 8.

Setting	Value
Primary series: hardware and software	
GPU	One NVIDIA GeForce RTX 5090 (32 GB)
Numerical precision	bf16
Gradient checkpointing	Enabled
PyTorch	2.11.0+cu128
Transformers	5.16.1
PEFT	0.20.0
Primary series: supervised adaptation	
LoRA configuration	Rank 16; $\alpha = 32$; dropout 0.05
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
Training epochs	1
Optimizer	AdamW
Learning rate	2×10^{-4}
Batch size	1
Gradient accumulation	8 steps
Maximum sequence length	512 tokens
Learning-rate schedule	Cosine
Warm-up steps	10
Gradient clipping	1.0
Primary series: on-policy distillation	
Optimizer	AdamW
Learning rate	5×10^{-6}
Optimization cycles	77
Examples per cycle	Up to 32
Updates per cycle	1
Rollout micro-batch size	4
Teacher scoring / student update micro-batch size	1 / 1
Teacher distribution representation	Top $K_T = 64$ tokens plus residual mass
Sampling temperature	0.7
Sampling top-p	0.9
Maximum new tokens	128
Both series: retrieval	
Maximum retrieved bilingual examples	$K = 2$

Table 10: Primary-series implementation settings and the retrieval size shared by both series.

Towards a Rubric-based LLM-as-a-Judge for Patent Claim Translation Evaluation

Yingyi Fu and Takehito Utsuro
Deg. Prog. Sys.&Inf. Eng.,
Grad. Sch. Sci.&Tech.,
University of Tsukuba

Masaaki Nagata
NTT Communication Science Laboratories,
NTT Corporation, Japan

Abstract

Existing automatic machine translation evaluation metrics, such as COMETKiwi and MetricX, showed relatively low correlations with human evaluation in the WAT2025 Patent Claim Translation/Evaluation Tasks. We propose a rubric-based LLM-as-a-Judge method for patent claim translation evaluation. Based on a patent-specific rubric consisting of five evaluation dimensions with corresponding checklist items, we instruct a GPT model to assess translation quality. On a publicly available human evaluation dataset derived from the WAT2025 Patent Claims Translation/Evaluation Tasks test set, the proposed method achieves the highest Pearson and Spearman correlation coefficients with human evaluation among all evaluated methods.

1 Introduction

Patent claim translation is more difficult than general translation because of complex sentence structures, long sentences, and highly specialized technical terminology. Evaluating patent claim translations is also challenging because seemingly minor errors, such as the omission of a limitation or an incorrect element relationship, may substantially change the meaning or scope of the claimed invention.

Although research on patent translation has a long history, research specifically focusing on patent claim translation has been relatively limited. In 2007, a bilingual Japanese–English patent parallel corpus was created by Utiyama and Isahara (2007) for the NTCIR-6 patent retrieval task (Fujii et al., 2007). More recently, Nagata et al. (2024) built JaParaPat, a Japanese–English parallel patent corpus containing over 300 million sentence pairs, aligning every part of the patent documents including titles, abstracts, descriptions, and claims. Several previous studies on patent translation have focused on constructing patent corpora rather than developing patent-translation-specific methods, e.g., COPPA V2.0 (Junczys-Dowmunt et al., 2016), ParaPat (Soares et al., 2020) and JaParaPat. Moreover, relatively little research has focused specifically on patent claim translation evaluation. In 2025, Nakazawa et al. (2025) organized the first shared task focusing on patent claim translation at WAT2025. However, the organizers of WAT2025 reported that automatic metrics, such as COMETKiwi (Rei et al., 2023) and MetricX (Juraska et al., 2024), showed low correlations with human evaluation. Correlations between general machine translation (MT) LLM-as-a-Judge such as GEMBA (Kocmi and Federmann, 2023b,a) and human evaluation were not reported.

To develop an automatic patent claim translation evaluation method that correlates better with human evaluation, we propose an LLM-as-a-Judge method to assess translation quality based on a patent-specific rubric and checklist. The experimental results show that the proposed method achieves higher correlations with WAT2025 human evaluation than COMETKiwi, MetricX, GEMBA-DA and GEMBA-MQM.

In summary, (1) we develop a rubric-based LLM-as-a-Judge method specific to patent claim translation evaluation, which achieves the highest correlations with human evaluation among the evaluated methods. (2) We evaluate several automatic metrics on WAT2025 human evaluation data, including LLM-as-a-Judge methods and neural metrics such as COMETKiwi and MetricX. (3) We compare the performance of GPT-5.2 and GPT-4 using the same LLM-as-a-Judge methods.

2 Related Work

2.1 Automatic Evaluation of MT

BLEU (Papineni et al., 2002) is a traditional automatic MT evaluation metric that measures MT quality based on modified n-gram precision and a

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 84–103
November 9, 2026 ©2026 Association for Computational Linguistics

brevity penalty. Rei et al. (2020) proposed COMET, a neural MT evaluation metric based on multilingual pretrained language models. COMET uses information from the source text and the reference translation to evaluate MT quality and achieves a high correlation with human judgments. Rei et al. (2022) then presented COMETKiwi, a reference-free MT evaluation metric based on the COMET framework. Freitag et al. (2022) reported that neural evaluation metrics such as COMET outperform overlap-based metrics such as BLEU.

2.2 GEMBA-DA and GEMBA-MQM

To investigate the use of LLMs for MT quality assessment, Kocmi and Federmann (2023b) developed GEMBA. One of its prompt variants, GEMBA-DA, uses a GPT model to directly assess translation quality by assigning a score.

Unlike GEMBA-DA, GEMBA-MQM (Kocmi and Federmann, 2023a) uses a GPT model to identify and classify translation errors into error categories and severity levels based on the Multidimensional Quality Metrics (MQM) framework used by Freitag et al. (2021). Specifically, GEMBA-MQM classifies translation errors into categories such as accuracy, fluency, and terminology. It also marks each translation error as critical, major, or minor, with default weights of 25, 5, and 1, respectively. Table 1 shows the error severity levels and categories. The prompt for GEMBA-MQM includes three few-shot examples.

Both methods take the source text and the MT output as input. GEMBA-DA can optionally use a reference translation, whereas GEMBA-MQM is reference-free. Both methods use predefined prompts. The output of GEMBA-DA is a direct assessment (DA) score, whereas the outputs of GEMBA-MQM are error annotations and a negative score calculated based on the number of translation errors and their severity levels.

2.3 LiTransProQA

Zhang et al. (2025) proposed LiTransProQA, an LLM-based question-answering metric for literary translation evaluation. It provides a prompting template and a list of evaluation questions. The prompt instructs the LLM to answer the questions with "yes," "no," or "maybe," which are mapped to scores of 1, 0, and 0.5, respectively. LiTransProQA achieves adequacy performance comparable to that of trained linguistic students, showing its potential for literary translation evaluation. Inspired by LiTransProQA, we develop a patent-claim-specific checklist with three answer options and a score-mapping rule with the assistance of coding agents.

Error Severity	Error Category
Critical	• Accuracy • Fluency • Terminology • Style • Locale convention (omitted from the general GEMBA-MQM prompt) • Other • Non-translation
Major	Same as above
Minor	Same as above

Table 1: GEMBA-MQM Error Severity Levels and Categories.

2.4 Patent Claims Translation/Evaluation Task at WAT2025

Nakazawa et al. (2025) organized the WAT2025 Patent Claims Translation/Evaluation Tasks, the first shared task concentrating on patent claim translation.

The test set consisted of claims from applications without a corresponding application in the target language when the source texts were distributed to the participants and translations from six MT systems. Three of them were submissions from participants: an open-weight LLM trained using continual pretraining and supervised fine-tuning on JaParaPat (Azami et al., 2025), and two GPT-5-based systems using prompt engineering (Edamatsu et al., 2025). The other three anonymous MT systems were a general MT system, a patent-specific MT system, and an open-weight LLM.

A subset of the test data was manually evaluated using human evaluation criteria modified from the MQM-based framework of Freitag et al. (2021) for patent translation. However, the organizers also reported issues with the quality of the MQM-based human annotations. Therefore, in this study, we use the human evaluation scores and evaluate their correlations with automatic evaluation scores.

3 Proposed Method

3.1 Prompt for Proposed Method

The proposed method, Patent-Rubric-Checklist, assesses translation quality using a rubric specific to patent claim translation evaluation.

Notably, the evaluation dimensions and the checklist items are mainly developed with the assistance

of Copilot. We instructed Copilot to design a checklist for patent claim translation evaluation with reference to LiTransProQA by providing general translation criteria such as semantic equivalence and fluency, as well as patent-specific constraints including the number of claim elements and their logical relationships, the structure of the claim as a single sentence, and consistency of terminology. Moreover, we tried to improve the prompt with ChatGPT because we assumed that it is stronger than Copilot.

First, the prompt for the proposed method instructs a GPT model to evaluate each checklist item according to the rubric. As shown in Figure 1, the first part of the prompt describes the task setting for patent claim translation evaluation and specifies the required evaluation procedure. Specifically, the prompt instructs the GPT model to answer each checklist item as follows:

- "yes": no problem
- "partial": minor or weakening problem
- "no": clear or serious problem

Second, the prompt presents the rubric consisting of the five evaluation dimensions and their corresponding checklist items for patent claim translation evaluation, as shown in Figure 2. Lines 23–27 define Dimension [A] Content Fidelity, which evaluates the preservation of the core inventive idea and technical functions or effects, as well as the absence of hallucinated technical content. Lines 29–33 define Dimension [B] Element Coverage, which evaluates the translation of the claim elements, as well as the preservation of element relationships, quantities, ranges, or conditions. Lines 35–42 define Dimension [C] Claim Scope & Logical Constraints, which evaluates the preservation of claiming scope, the correct attachment of constraint clauses, and the preservation of coordination and modification relations. Lines 44–50 define Dimension [D] Claim Form & Syntactic Requirements, which evaluates whether the translation is written as a single sentence claim, preserves the claim type, maintains consistent antecedent basis, and avoids narrative or explanatory prose style. Lines 52–56 define Dimension [E] Fluency & Grammaticality, which evaluates the grammar correctness, stylistic appropriateness, and readability.

Third, the prompt presents the input and the output formats, shown as Figure 3. The prompt instructs the GPT model to return the answers to the checklist items with justifications and an overall explanation in JSON format.

3.2 Scoring Process

We map the checklist answers to 1.0 for "yes," 0.5 for "partial," and 0.0 for "no." For each evaluation dimension, we average the mapped scores of its checklist items to obtain a dimension score, denoted by A through E in Equation 1. We then aggregate the dimension scores using the following weighted formula:

$$\text{rule_based_score} = 100 \times \big(0.25A + 0.25B + 0.25C + 0.15D + 0.10E\big) \quad (1)$$

Specifically, the weights assigned to each dimension in Equation 1 were suggested by Copilot.

4 Experiment Setup

4.1 Dataset

In this study, we use a manually annotated dataset[1] derived from the WAT2025 patent claims translation evaluation test set[2]. The test set contains 26 documents with 70 claims for the Japanese-to-English (JA–EN) direction and 30 documents with 81 claims for the English-to-Japanese (EN–JA) direction. The JA–EN set contains outputs from six MT systems and the EN–JA set contains outputs from four MT systems. No reference translations are provided for either language direction. The human evaluation dataset contains scores and annotations for the MT outputs of 13 documents with 28 claims in each language direction from the test set described above.

4.2 Prompting Methods

In our experiments, we use three prompting methods to evaluate the translation quality. Both GEMBA-DA and GEMBA-MQM are existing LLM-as-a-Judge methods for MT evaluation developed by Kocmi and Federmann (2023b,a), whereas Patent-Rubric-Checklist is developed in this study for patent claim translation evaluation.

In summary:

- **GEMBA-DA:** Directly assigns a score ranging from 0 to 100 based on meaning and grammar (baseline).

[1] https://github.com/Tzawa/pat-claim-trans
[2] WAT2025 Patent Claims Translation/Evaluation Tasks

```
You are an expert evaluator of MACHINE TRANSLATION OF PATENT CLAIMS.

Your task is to evaluate the translation quality using a
rubric-based evaluation framework,
implemented explicitly as a CHECKLIST.

This is NOT general prose.
The input is a PATENT CLAIM, which has strict legal and structural constraints.

You must:
(1) Answer ALL checklist questions below using:
    - "yes" (no problem)
    - "partial" (minor or weakening problem)
    - "no" (clear or serious problem)
(2) For each question, provide a short justification.
(3) Provide a brief explanation of the evaluation result.
(4) Evaluate SOURCE vs HYPOTHESIS only; no reference is provided.

```

Figure 1: Prompt for Patent-Rubric-Checklist, Part 1: Task Setting and Evaluation Procedure.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[A] CONTENT FIDELITY (Meaning Preservation)

A1. Is the core inventive idea (what is being claimed) preserved?
A2. Are the technical functions or effects preserved without distortion?
A3. Is the translation free from hallucinated technical content not present in the
    source?

[B] ELEMENT COVERAGE (Completeness)

B1. Are all essential claim elements translated (no omissions)?
B2. Are relationships between elements (having, connected to, arranged on, etc.)
    preserved?
B3. Are quantities, ranges, or conditions (e.g. "at least one", numerical ranges)
    preserved?

[C] CLAIM SCOPE & LOGICAL CONSTRAINTS (CRITICAL)

C1. Is the claiming scope preserved?
    (e.g., "comprising" vs "consisting of", "characterized in that")
C2. Are constraint clauses correctly attached?
    (e.g., "wherein", "前記〇〇が〜")
C3. Are coordination vs modification relations preserved?
    (e.g., A and B that each have C, NOT A and (B that has C))

[D] CLAIM FORM & SYNTACTIC REQUIREMENTS

D1. Is the translation written as a SINGLE SENTENCE claim?
D2. Is the claim type preserved (apparatus / method / composition, etc.)?
D3. Is antecedent basis consistent?
    (a/the/said, 前記〜)
D4. Does the translation avoid narrative or explanatory prose style?

[E] FLUENCY & GRAMMATICALITY

E1. Is the translation grammatically correct in the target language?
E2. Is it stylistically appropriate for patent claims?
E3. Is the sentence readable and not excessively awkward or ambiguous?

```

Figure 2: Prompt for Patent-Rubric-Checklist, Part 2: Rubric and Checklist Items.

```
--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "A1": {{"answer": "yes", "justification": "Short justification."}},
    "A2": {{"answer": "yes", "justification": "Short justification."}},
    "A3": {{"answer": "yes", "justification": "Short justification."}},
    "B1": {{"answer": "yes", "justification": "Short justification."}},
    "B2": {{"answer": "yes", "justification": "Short justification."}},
    "B3": {{"answer": "yes", "justification": "Short justification."}},
    "C1": {{"answer": "yes", "justification": "Short justification."}},
    "C2": {{"answer": "yes", "justification": "Short justification."}},
    "C3": {{"answer": "yes", "justification": "Short justification."}},
    "D1": {{"answer": "yes", "justification": "Short justification."}},
    "D2": {{"answer": "yes", "justification": "Short justification."}},
    "D3": {{"answer": "yes", "justification": "Short justification."}},
    "D4": {{"answer": "yes", "justification": "Short justification."}},
    "E1": {{"answer": "yes", "justification": "Short justification."}},
    "E2": {{"answer": "yes", "justification": "Short justification."}},
    "E3": {{"answer": "yes", "justification": "Short justification."}}
  }},
  "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 3: Prompt for Patent-Rubric-Checklist, Part 3: Input and Output Formats.

- **GEMBA-MQM:** Detects translation errors, labels their category and severity, and calculates a score based on error severities (baseline).

- **Patent-Rubric-Checklist:** Instructs a GPT model to evaluate translation quality based on a rubric with five evaluation dimensions and corresponding checklist items. Then maps the checklist answers to numerical values and calculates the final score outside the LLM using predefined weights (proposed method).

The figures in Appendix A provide the detailed prompts for GEMBA-DA and GEMBA-MQM.

To facilitate comparison with the human evaluation score, GEMBA-DA score, and our proposed method's score, we convert the negative raw GEMBA-MQM score into a final score as follows:

$$\text{final_score} = 100 + 4 \times \text{GEMBA-MQM_score} \quad (2)$$

where final_score denotes the final score, and GEMBA-MQM_score denotes the negative raw score calculated from the GEMBA-MQM annotations.

Furthermore, we use GPT-5.2 (OpenAI, 2025) and GPT-4 (OpenAI et al., 2024) as judge models for different prompting methods.

4.3 Evaluation Metrics

We use Pearson's correlation coefficient[3] and Spearman's rank correlation coefficient[4] as evaluation metrics in our experiments. Pearson correlation measures the linear relationship between automatic evaluation scores and human evaluation scores, while Spearman correlation measures the monotonic relationship between them.

Furthermore, we conduct permutation tests[5] for both Pearson and Spearman correlations to compare our proposed method with GEMBA-DA and GEMBA-MQM. All three methods use GPT-5.2.

[3] SciPy documentation for `pearsonr`
[4] SciPy documentation for `spearmanr`
[5] SciPy documentation for `permutation_test`

5 Results and Analysis

5.1 Correlations with Human Evaluation

We compute Pearson and Spearman correlations between human evaluation scores and the scores produced by GEMBA-DA, GEMBA-MQM, and our proposed method, Patent-Rubric-Checklist, using both GPT-4 and GPT-5.2. We also conduct the same correlation analysis for reference-free MetricX-24 and COMETKiwi. Specifically, we use `Unbabel/wmt23-cometkiwi-da-xl`[6] and `google/metricx-24-hybrid-xl-v2p6`[7] in our experiments. For MetricX-24, we negate the scores when computing correlations so that higher scores indicates better quality. As shown in Table 2 , for the JA–EN direction, all correlations between automatic evaluation scores and human evaluation scores are positive and statistically significant ($p < 0.05$ or $p < 0.01$). However, for the EN–JA direction, the Spearman correlation of COMETKiwi and both the Pearson and Spearman correlation for reference-free MetricX-24 and all the LLM-as-a-Judge methods using GPT-4, are not statistically significant. In particular, although GEMBA-MQM using GPT-4 shows negative Pearson and Spearman correlations, neither is statistically significant. Our proposed method, Patent-Rubric-Checklist using GPT-5.2, achieves the highest Pearson and Spearman correlations in both translation directions. Additionally, we observe that results for the JA–EN direction yield higher correlations than those for EN–JA direction. We also observe that GPT-5.2 yields higher correlations than GPT-4 for all prompting methods.

We conduct permutation tests between the proposed method and GEMBA-MQM using GPT-5.2, the proposed method and GEMBA-DA using GPT-5.2, as well as reference-free MetricX-24 and COMETKiwi. For the JA–EN direction, the difference in Spearman correlation between Patent-Rubric-Checklist and GEMBA-DA, both using GPT-5.2, is statistically significant ($p < 0.01$), whereas the difference in Pearson correlation is not. Similarly, the difference in Spearman correlation between Patent-Rubric-Checklist and GEMBA-MQM, both using GPT-5.2, is statistically significant ($p < 0.05$), whereas the difference in Pearson correlation coefficient is not. The differences in Pearson and Spearman correlations between reference-free MetricX-24 and COMETKiwi are not statistically significant.

For the EN–JA direction, the differences in Pearson and Spearman correlations are not statistically significant between Patent-Rubric-Checklist and either GEMBA-DA or GEMBA-MQM, all using GPT-5.2. The differences in both correlations between reference-free MetricX-24 and COMETKiwi are not statistically significant either.

5.2 Dimension Ablation Study

To further analyze the performance of Patent-Rubric-Checklist using GPT-5.2, we conduct a dimension ablation study by computing the Pearson and Spearman correlations for prompt variants containing only one dimension. Specifically, to analyze the effect of each dimension, we remove the other four dimensions and their corresponding checklist items from the prompt shown in Figure 2. We also revise the output format shown in Figure 3 to match the revised rubric. The prompt variants with a single dimension are provided in Figures 6–11 in Appendix A. Additionally, we set the weight of the single dimension to 1.0. We make these modifications with the assistance of ChatGPT.

Table 3 shows the Pearson and Spearman correlations between human evaluation scores and scores calculated using prompt variants with GPT-5.2. In both translation directions, the prompt variant containing only dimension A achieves the lowest Pearson and Spearman correlations. Surprisingly, although dimensions D and E are assigned relatively lower weights in Equation 1, the prompt variant containing only dimension E achieves the highest Pearson correlation, while the variant containing only dimension D achieves the highest Spearman correlation in both translation directions.

We additionally present case studies illustrating differences between the judgments of dimensions A–C and D–E in Appendix B. The single-dimension scores for dimensions D and E correlate more strongly with human evaluation scores, suggesting that the human evaluation may be more sensitive to the quality aspects captured by dimensions D–E (claim form, syntactic requirements, and fluency & grammaticality) than to those captured by dimensions A–C (content fidelity, element coverage, and claim scope & logical constraints). However, based on the available data and results, we cannot determine why the single-dimension scores for dimensions D and E achieve higher correlations with human evaluation scores.

[6] Unbabel/wmt23-cometkiwi-da-xl
[7] google/metricx-24-hybrid-xl-v2p6

Methods	Pearson's r		Spearman's ρ	
	ja–en	en–ja	ja–en	en–ja
COMETKiwi	0.2880**	0.1860*	0.2703**	0.1810
MetricX-24 (reference-free)	0.2436**	0.1397	0.2945**	0.1770
GEMBA-DA (GPT-4)	0.3753**	0.1391	0.3664**	0.0806
GEMBA-DA (GPT-5.2)	0.4685**	0.1996*	0.4210**	0.3017**
GEMBA-MQM (GPT-4)	0.2989**	-0.0297	0.4535**	-0.1291
GEMBA-MQM (GPT-5.2)	0.5533**	0.2461**	0.4741**	0.3135**
Patent-Rubric-Checklist (GPT-4)	0.1906*	0.1329	0.2691**	0.1034
Patent-Rubric-Checklist (GPT-5.2)	**0.6214****	**0.3767****	**0.5931****	**0.4417****

$^{*}p < 0.05$, $^{**}p < 0.01$.

Table 2: Pearson and Spearman Correlations between Automatic Evaluation Scores and Human Evaluation Scores.

Rubric Setting	Pearson's r		Spearman's ρ	
	ja–en	en–ja	ja–en	en–ja
Full	0.6214**	0.3767**	0.5931**	0.4417**
Only A	0.4331**	0.2853**	0.4788**	0.3862**
Only B	0.4856**	0.3352**	0.4878**	0.4039**
Only C	0.5621**	0.3835**	0.5477**	0.4619**
Only D	0.6275**	**0.5239****	0.6050**	**0.5032****
Only E	**0.6314****	0.3535**	**0.6102****	0.3874**

$^{*}p < 0.05$, $^{**}p < 0.01$.

Table 3: Dimension Ablation Study of Patent-Rubric-Checklist (GPT-5.2) using Pearson and Spearman Correlations.

5.3 Comparison of Evaluation Results Using GPT-4 and GPT-5.2

We observe that GPT-5.2 achieves higher Pearson and Spearman correlations with human evaluation than GPT-4 for all LLM-as-a-Judge methods. Specifically, for the JA–EN direction, Patent-Rubric-Checklist using GPT-5.2 achieves Pearson and Spearman correlations that are 0.4308 and 0.3240 higher, respectively, than those achieved using GPT-4. We find that Patent-Rubric-Checklist using GPT-4 assigns a score of 100 to all translations produced by MT systems 2, 3, and 5, indicating that all checklist answers are "yes".

For the EN–JA direction, Patent-Rubric-Checklist using GPT-5.2 achieves Pearson and Spearman correlations that are 0.2438 and 0.3383 higher, respectively, than those achieved using GPT-4. We find that Patent-Rubric-Checklist using GPT-4 assigns a score of 100 to all translations produced from MT systems 1 and 2.

In contrast, when using GPT-5.2, we do not observe Patent-Rubric-Checklist assigning a score of 100 to all translations from any single MT system in either translation direction. Furthermore, the overall proportion of translations receiving a score of 100 using GPT-4 as the judge model is notably higher than that using GPT-5.2 (92.26% vs. 2.98% for JA–EN and 98.21% vs. 5.36% for EN–JA).

Table 4 presents an example of the checklist answers produced by the two GPT models and corresponding externally calculated rule-based scores for the same translation. For the same MT output, GPT-4 answers "yes" to all checklist items, while GPT-5.2 answers "yes" to only one of them. Consequently, the rule-based scores calculated from the two GPT models' checklist answers are substantially different. In particular, although the English translation consists of multiple sentences, GPT-4 fails to detect this error when answering checklist item D1.

6 Discussion

The novelty of our proposed method is that we prompt the GPT model with a rubric consisting of five evaluation dimensions and corresponding checklist items designed specifically for patent claim translation evaluation. The prompt for our proposed method is for Japanese–English translation evaluation. To adapt the prompt to other language pairs, it is important to design a rubric that reflects the language-pair-specific characteristics for patent claim translation evaluation. For example, we can provide patent claims, examples of translation errors or human evaluation criteria in the target language pair to the LLM and instruct it to analyze important evaluation aspects and adapt the rubric, including its checklist items, to the target language pair. The results of this study suggest that designing a rubric with the assistance of coding agents may be a practical approach to patent claim translation evaluation, since documents explaining the guidelines for filing U.S. patent applications and differences between Japanese and U.S. patents are widely available on the internet.

Moreover, the MT outputs in the human evaluation data provided by the WAT2025 organizers include translations generated by a GPT-5-based

Source Text
[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。
MT Output
[Claim 1] A propulsion device comprising: a cylindrical portion arranged with its axis extending in a forward-rearward direction submerged below the water surface, with one end positioned upstream and the other end positioned downstream to form a flow channel; an axis portion arranged inside the cylindrical portion and extending along the axis direction; a propeller mounted on the axis portion, having multiple propeller blades extending radially along the axis direction and arranged circumferentially around the axis, capable of rotating about the axis; and multiple struts positioned either forward or rearward of the propeller within the flow channel, extending radially and arranged circumferentially to support the axis portion. The multiple struts extend vertically and are arranged symmetrically with respect to a vertical symmetry line passing through the axis direction. At least one angle interval between the struts differs from other angle intervals between the struts. At least one strut is positioned on both sides of the vertical direction relative to a horizontal plane extending horizontally through the axis direction.
Answer to D1 from Patent-Rubric-Checklist (GPT-4)

`"D1": {`
`  "answer": "yes",`
`  "justification": "The translation is written as a single sentence claim."`
`},`

Answer to D1 from Patent-Rubric-Checklist (GPT-5.2)

`"D1": {`
`  "answer": "no",`
`  "justification": "The English claim is written as multiple sentences; patent claims should be a single sentence, and the source is a single-sentence claim."`
`},`

Checklist Answers from Patent-Rubric-Checklist (GPT-4) and Rule-Based Score

`A1: yes, A2: yes, A3: yes,`
`B1: yes, B2: yes, B3: yes,`
`C1: yes, C2: yes, C3: yes,`
`D1: yes, D2: yes, D3: yes, D4: yes,`
`E1: yes, E2: yes, E3: yes`
`rule_based_score: 100`

Checklist Answers from Patent-Rubric-Checklist (GPT-5.2) and Rule-Based Score

`A1: partial, A2: partial, A3: partial,`
`B1: partial, B2: partial, B3: yes,`
`C1: partial, C2: partial, C3: partial,`
`D1: no, D2: yes, D3: partial, D4: partial,`
`E1: partial, E2: partial, E3: partial`
`rule_based_score: 54.1666666666666`

Table 4: Example Answers from the Proposed Method Using GPT-4 and GPT-5.2.

system (Edamatsu et al., 2025; Nakazawa et al., 2025). This raises the possibility that the LLM-as-a-Judge using GPT-4 answered "yes" to every checklist item when evaluating outputs from these systems.

7 Conclusion

This study proposed a rubric-based LLM-as-a-Judge method specific to patent claim translation evaluation. Specifically, our proposed method instructs a GPT model to evaluate MT outputs based on a rubric consisting of five evaluation dimensions and corresponding checklist items. On both the JA–EN and the EN–JA test sets show that our proposed method using GPT-5.2 achieves higher correlations with human evaluation than GEMBA-DA and GEMBA-MQM using either GPT-4 or GPT-5.2, suggesting the potential of the rubric-based LLM-as-a-Judge method for patent claim translation evaluation. In the dimension ablation study, we find that the prompt variants containing only dimension D and E, which have relatively lower weights in the proposed method, achieve the highest Spearman and Pearson correlations, respectively. We also observe that GPT-5.2 achieves higher correlations with human evaluation than GPT-4 for all LLM-as-a-Judge methods. In particular, all translations from some MT systems in each translation direction receive a score of 100 under the proposed method using GPT-4, indicating that GPT-4 answers every checklist item with "yes". In contrast, no similar pattern is observed in the results obtained using GPT-5.2. Moreover, we observe that all automatic evaluation methods yield lower correlations on the EN–JA test set than on the JA–EN test set.

Future work will focus on improving the design of the rubric and checklist items to better align with human evaluation. We will also optimize the dimension weights to develop a scoring rule that yields higher correlation with human evaluation. For example, we could use a held-out set to optimize the dimension weights of the rubric.

Limitations

Although this study suggests the potential of a rubric-based LLM-as-a-Judge method for patent claim translation evaluation, several limitations remain.

First, the dataset used in this study does not include reference translations, therefore, the performance of the proposed method when reference translations are available remains unknown. Second, the dataset is limited, only consisting of 28 claims with translations from six MT systems for the JA–EN direction and 28 claims with translations from four MT systems for the EN–JA direction. Third, we run each prompt only once on the test set, therefore, we cannot assess the stability of the LLM-as-a-Judge methods. Finally, we use only GPT-4 and GPT-5.2 as the judge models. Consequently, the performance of the proposed method using open-weight LLMs such as Qwen and Llama remains unknown.

References

Haruto Azami, Yin Zhang, Futo Kajita, Nobuyori Nishimura, and Takehito Utsuro. 2025. UTSK25 at WAT2025 patent claims translation/evaluation task. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 30–34, Mumbai, India. Association for Computational Linguistics.

Taishi Edamatsu, Isao Goto, and Takashi Ninomiya. 2025. Ehime-U system with judge and refinement, specialized prompting, and few-shot for the patent claim translation task at WAT 2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 16–29, Mumbai, India. Association for Computational Linguistics.

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.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Atsushi Fujii, Makoto Iwayama, and Noriko Kando. 2007. Overview of the patent retrieval task at the ntcir-6 workshop. In *NTCIR*.

Marcin Junczys-Dowmunt, Bruno Pouliquen, and Christophe Mazenc. 2016. Coppa v2. 0: Corpus of parallel patent applications building large parallel corpora with gnu make. In *4th Workshop on Challenges in the Management of Large Corpora Workshop Programme*, pages 15–19.

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 and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

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.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Kazuhiro Kasada, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2025. Findings of the first patent claims translation task at WAT2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 1–15, Mumbai, India. Association for Computational Linguistics.

OpenAI. 2025. Introducing GPT-5.2. `https://openai.com/index/introducing-gpt-5-2/`.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de

Souza, and André F. T. 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.

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.

Felipe Soares, Mark Stevenson, Diego Bartolome, and Anna Zaretskaya. 2020. ParaPat: The multi-million sentences parallel corpus of patents abstracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3769–3774, Marseille, France. European Language Resources Association.

Masao Utiyama and Hitoshi Isahara. 2007. A Japanese-English patent parallel corpus. In *Proceedings of Machine Translation Summit XI: Papers*, Copenhagen, Denmark.

Ran Zhang, Wei Zhao, Lieve Macken, and Steffen Eger. 2025. LiTransProQA: An LLM-based literary translation evaluation metric with professional question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29099–29121, Suzhou, China. Association for Computational Linguistics.

A Prompts Used for Each Method in Experiments

This appendix presents the prompts used for GEMBA-DA, GEMBA-MQM and dimension ablation study in our experiments, as shown in Figures 4–11

B Case Studies for Dimension Ablation

We select two cases for each translation direction based on the scores obtained using the single-dimension prompt variants: one where the average single-dimension score across dimensions A–C exceeds that across dimensions D–E by the largest margin, and one in which the average single-dimension score across dimensions D–E exceeds that across dimensions A–C by the largest margin. The five dimensions A–E concern [A] Content Fidelity, [B] Element Coverage, [C] Claim Scope & Logical Constraints, [D] Claim Form & Syntactic Requirements, and [E] Fluency & Grammaticality. Tables 5–8 present the four selected examples. All calculations use unrounded values, while the scores reported in the tables are rounded to two decimal places.

In both cases where the average score for dimensions D–E is lower than that for dimensions A–C, the translation fails to preserve the single-sentence structure of the patent claim (`D1: no`). Moreover, in the EN–JA example where the average score for dimensions A–C is higher than that for dimensions D–E (Table 7), the same sentence-splitting also affects the judgment for C1, resulting in an answer of "partial", because, according to the LLM's explanation, splitting the claim may marginally blur the intended single-claim constraint framing.

```
Score the following translation from {source_lang} to{target_lang} on a continuous scale from 0 to 100, where a score of zero means "no meaning preserved" and score of one hundred means "perfect meaning and grammar".

{source_lang} source: "{source_seg}"
{target_lang} translation: "{target_seg}"
Score:
```

Figure 4: Prompt for GEMBA-DA.

```
(system) You are an annotator for the quality of machine translation. Your task is to identify errors and assess the quality of the translation.

(user) {source_lang} source:
```{source_seg}```
{target_lang} translation:
```{target_seg}```

Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error.
Each error is classified as one of three categories: critical, major, and minor.
Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what
the text is trying to say is still understandable. Minor errors are technically errors,
but do not disrupt the flow or hinder comprehension.
```

Figure 5: Prompt for GEMBA-MQM.

```
You are an expert evaluator of MACHINE TRANSLATION OF PATENT CLAIMS.
Your task is to evaluate the translation quality using a
rubric-based evaluation framework,
implemented explicitly as a CHECKLIST.
This is NOT general prose.
The input is a PATENT CLAIM, which has strict legal and structural constraints.

You must:
(1) Answer ALL checklist questions below using:
    - "yes" (no problem)
    - "partial" (minor or weakening problem)
    - "no" (clear or serious problem)
(2) For each question, provide a short justification.
(3) Provide a brief explanation of the evaluation result.
(4) Evaluate SOURCE vs HYPOTHESIS only; no reference is provided.

```

Figure 6: Shared Part of the Single-Dimension Prompts for Patent-Rubric-Checklist.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[A] CONTENT FIDELITY (Meaning Preservation)

A1. Is the core inventive idea (what is being claimed) preserved?
A2. Are the technical functions or effects preserved without distortion?
A3. Is the translation free from hallucinated technical content not present in the
    source?

--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "A1": {{"answer": "yes", "justification": "Short justification."}},
    "A2": {{"answer": "yes", "justification": "Short justification."}},
    "A3": {{"answer": "yes", "justification": "Short justification."}}
  }},
  "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 7: Prompt for Patent-Rubric-Checklist with Only Dimension A: Rubric, Checklist Items and Input/Output Format.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[B] ELEMENT COVERAGE (Completeness)

B1. Are all essential claim elements translated (no omissions)?
B2. Are relationships between elements (having, connected to, arranged on, etc.)
    preserved?
B3. Are quantities, ranges, or conditions (e.g. "at least one", numerical ranges)
    preserved?

--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "B1": {{"answer": "yes", "justification": "Short justification."}},
    "B2": {{"answer": "yes", "justification": "Short justification."}},
    "B3": {{"answer": "yes", "justification": "Short justification."}}
    }},
  "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 8: Prompt for Patent-Rubric-Checklist with Only Dimension B: Rubric, Checklist Items and Input/Output Format.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[C] CLAIM SCOPE & LOGICAL CONSTRAINTS (CRITICAL)

C1. Is the claiming scope preserved?
    (e.g., "comprising" vs "consisting of", "characterized in that")
C2. Are constraint clauses correctly attached?
    (e.g., "wherein", "前記◯◯が〜")
C3. Are coordination vs modification relations preserved?
    (e.g., A and B that each have C, NOT A and (B that has C))

--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "C1": {{"answer": "yes", "justification": "Short justification."}},
    "C2": {{"answer": "yes", "justification": "Short justification."}},
    "C3": {{"answer": "yes", "justification": "Short justification."}}
  }},
 "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 9: Prompt for Patent-Rubric-Checklist with Only Dimension C: Rubric, Checklist Items and Input/Output Format.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[D] CLAIM FORM & SYNTACTIC REQUIREMENTS

D1. Is the translation written as a SINGLE SENTENCE claim?
D2. Is the claim type preserved (apparatus / method / composition, etc.)?
D3. Is antecedent basis consistent?
(a/the/said, 前記〜)
D4. Does the translation avoid narrative or explanatory prose style?

--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "D1": {{"answer": "yes", "justification": "Short justification."}},
    "D2": {{"answer": "yes", "justification": "Short justification."}},
    "D3": {{"answer": "yes", "justification": "Short justification."}},
    "D4": {{"answer": "yes", "justification": "Short justification."}}
  }},
 "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 10: Prompt for Patent-Rubric-Checklist with Only Dimension D: Rubric, Checklist Items and Input/Output Format.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[E] FLUENCY & GRAMMATICALITY

E1. Is the translation grammatically correct in the target language?
E2. Is it stylistically appropriate for patent claims?
E3. Is the sentence readable and not excessively awkward or ambiguous?

--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
  "E1": {{"answer": "yes", "justification": "Short justification."}},
  "E2": {{"answer": "yes", "justification": "Short justification."}},
  "E3": {{"answer": "yes", "justification": "Short justification."}}
  }},
  "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}},
```

Figure 11: Prompt for Patent-Rubric-Checklist with Only Dimension E: Rubric, Checklist Items and Input/Output Format.

Source Text

[請求項 3] Ｎ本の前記ストラットのそれぞれが、前記軸線回りの領域を前記周方向に均等に分割する（Ｎ × ２ − １）本の分割線のいずれかに配置され、１以上Ｎ未満であるＮ’に対して、前記ストラット同士の角度間隔は、Ｎ’ × ３６０／（Ｎ × ２ − １）度のいずれかである、請求項１又は２に記載の推進装置。

MT Output

[Claim 3] In a propulsion device according to Claims 1 or 2, each of the N aforementioned struts is positioned along one of the (N×2 − 1) dividing lines that evenly divide the area around the axis line in a circumferential direction. For N' where 1 ≤ N' < N, the angular spacing between the struts is either N'×360/(N×2 − 1) degrees.

Dimension A

Checklist answers: `A1: partial, A2: yes, A3: yes`

Overall explanation: The translation largely preserves the claim' s technical meaning and mathematical constraints. The main issue is a minor loss of legal precision regarding whether N' is explicitly an integer/natural number, which could slightly broaden interpretation.

Rule-based score: 83.33

Dimension B

Checklist answers: `B1: yes, B2: yes, B3: yes`

Overall explanation: The translation preserves the claim' s structural elements, inter-element relationships, and mathematical/conditional limitations. Any wording differences (e.g., "area around the axis line") do not materially change the technical/legal constraints expressed in the source.

Rule-based score: 100.00

Dimension C

Checklist answers: `C1: partial, C2: yes, C3: yes`

Overall explanation: The translation is structurally faithful and preserves the main logical constraints of the claim. The only notable issue is a slight potential weakening/ambiguity in expressing the "one of" set-selection over N' values as "either" and as a per-N' phrasing.

Rule-based score: 83.33

Dimension D

Checklist answers: `D1: no, D2: yes, D3: partial, D4: partial`

Overall explanation: The translation generally preserves the technical/legal content and claim type, but it violates a key formal requirement by breaking the claim into two sentences. Minor issues also exist with antecedent tightness and slightly explanatory phrasing.

Rule-based score: 50.00

Dimension E

Checklist answers: `E1: yes, E2: partial, E3: partial`

Overall explanation: The translation is grammatically correct and largely understandable, but it is slightly non-idiomatic for patent-claim style and introduces mild ambiguity/awkwardness in key geometric phrasing and in expressing the set of allowable angular spacings.

Rule-based score: 66.67

Score Summary

A–C average: 88.89

D–E average: 58.33

Gap: 30.56

Human evaluation score: 50

Table 5: A JA–EN Case Where the Average Single-Dimension Score for Dimensions A–C Is Higher than That for Dimensions D–E.

Source Text

[請求項 1] 特定の種別の光の方路切替が可能な第１振分部と、前記第１振分部で取り扱うことができる特定の種別の光と、前記第１振分部で取り扱うことができない特定の種別の光の方路切替が可能な第２振分部と、を備え、前記第２振分部は、入力された前記光を、前記第１振分部を経由する第１経路と、前記第１振分部を経由しない第２経路の少なくともいずれかの経路に出力する光出力部と、前記第２経路上に設けられ、Ｎ（Ｎは１以上の整数）個のポートとｋ（ｋは２以上の整数）個のポートとを有し、前記Ｎ個のポートのいずれかのポートに入力された前記光を、他の前記第２振分部が接続されている前記ｋ個のポートのいずれかのポートから出力するＮ × ｋマトリックス光スイッチと、を備え、前記光出力部は、端末間において波長単位の光パスを確立する場合には入力された前記光を少なくとも前記第１経路に出力し、光ファイバ単位の光パスを確立する場合には入力された前記光を少なくとも前記第２経路に出力する、を備える光ノード装置。

MT Output

1. An optical node apparatus comprising: a first distribution unit configured to switch an optical path of a specific type of light; and a second distribution unit configured to switch optical paths of: (i) a specific type of light that can be handled by the first distribution unit; and (ii) a specific type of light that cannot be handled by the first distribution unit, the second distribution unit comprising: a light output unit configured to output the input light to at least one of: (i) a first path that passes through the first distribution unit; and (ii) a second path that does not pass through the first distribution unit; and an N x k matrix optical switch provided on the second path and having N (N is an integer equal to or greater than 1) ports and k (k is an integer equal to or greater than 2) ports, each of the k ports being connected to another second distribution unit, the N x k matrix optical switch being configured such that light input to any one of the N ports is output from any one of the k ports; wherein: when establishing an optical path between terminals on a wavelength basis, the light output unit is configured to output the input light to at least the first path; and when establishing an optical path on an optical fiber basis, the light output unit is configured to output the input light to at least the second path.

Dimension A

Checklist answers: `A1: yes, A2: partial, A3: partial`

Overall explanation: The translation largely preserves the claimed apparatus and its conditional routing behavior, with only a minor fidelity risk where the connectivity of the k ports is overstated (universalized) compared to the Japanese, slightly narrowing/altering the technical scope.

Rule-based score: 66.67

Dimension B

Checklist answers: `B1: partial, B2: partial, B3: partial`

Overall explanation: The translation largely preserves the claim’s elements and structure, but it introduces a potentially material scope change by strengthening the connectivity requirement for the k ports (from ’any of the k ports’ to ’each of the k ports’), which can affect legal completeness and relationships.

Rule-based score: 50.00

Dimension C

Checklist answers: `C1: partial, C2: partial, C3: yes`

Overall explanation: The translation mostly preserves the claim’s structure and functional limitations, but it introduces a notable scope/logic deviation around the N×k switch’s k-port connectivity/output constraint, potentially altering the claimed network connection requirement.

Rule-based score: 66.67

Dimension D

Checklist answers: `D1: yes, D2: yes, D3: partial, D4: yes`

Overall explanation: Overall, the translation conforms well to patent-claim form and preserves the apparatus nature and structure. The main issue is a minor antecedent-basis looseness around “the input light,” which could be tightened for stricter claim clarity.

Rule-based score: 87.50

Dimension E

Checklist answers: `E1: yes, E2: yes, E3: partial`

Overall explanation: The translation is fluent and largely conforms to patent-claim English, with correct grammar and conventional claim drafting patterns. Minor awkward constructions introduce some interpretive ambiguity, but do not render the claim unreadable.

Rule-based score: 83.33

Score Summary

A–C average: 61.11

D–E average: 85.42

Gap: -24.31

Human evaluation score: 60

Table 6: A JA–EN Case Where the Average Single-Dimension Score for Dimensions D–E Is Higher than That for Dimensions A–C.

Source Text

1. A system including a first electronic device and a cell phone, the first electronic device including a processor and a wired or wireless interface to a network, said first electronic device being adapted to respond to user instructions by changing device state, the cell phone being equipped with a camera and a touchscreen, characterized in that the first electronic device lacks buttons by which the user instructions can be entered, the cell phone being adapted to communicate with the first electronic device over said network, the cell phone further including software instructions adapted to present on the cell phone touchscreen a user interface including Up and Down buttons by which a user can enter user control instructions for transmission to said first electronic device to control said device state, the software instructions also including instructions to obtain identification information corresponding to the first electronic device by decoding information from a 2D machine readable code on the first electronic device that is sensed by a camera of the cell phone, to thereby authorize the cell phone to interact with the first electronic device.

MT Output

1. 第 1 の電子装置と携帯電話を含むシステムであって、第 1 の電子装置はプロセッサと、ネットワークへの有線または無線インターフェースを備えており、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ことで応答するように構成され、携帯電話にはカメラとタッチスクリーンが装備されている。本システムは、第 1 の電子装置がユーザ指示を入力するためのボタンを有しておらず、携帯電話が当該ネットワークを介して第 1 の電子装置と通信するように構成されている点を特徴とする。さらに、携帯電話には、携帯電話のタッチスクリーン上にユーザインターフェースを表示するように適応されたソフトウェア指示が含まれており、このユーザインターフェースには上下ボタンが設けられており、ユーザはこのボタンを用いて第 1 の電子装置に送信して装置の状態を制御するためのユーザ制御指示を入力することができる。これらのソフトウェア指示には、携帯電話のカメラによって検知される第 1 の電子装置上の 2 次元機械読み取り可能コードから情報を解読することにより、第 1 の電子装置に対応する識別情報を取得する指示も含まれており、これにより携帯電話が第 1 の電子装置と相互作用することを許可する。

Dimension A

Checklist answers: `A1: yes, A2: yes, A3: yes`

Overall explanation: 翻訳は請求項の主要構成要件と機能的関係（UI による制御指示送信、2D コード読取りによる識別情報取得と認可）を適切に保持しており、意味の逸脱や不要な技術的付加は認められない。

Rule-based score: 100.00

Dimension B

Checklist answers: `B1: yes, B2: yes, B3: yes`

Overall explanation: The Japanese claim translation is complete and preserves the structural relationships and functional limitations of the English claim. No material loss of elements, conditions, or inter-element dependencies is observed.

Rule-based score: 100.00

Dimension C

Checklist answers: `C1: partial, C2: yes, C3: yes`

Overall explanation: The Japanese translation preserves the main claim scope and the logical attachment of limitations, with correct nesting of functional features. The only notable issue is a mild structural shift by splitting the claim into multiple sentences around the “characterized” portion, which could marginally blur the intended single-claim constraint framing.

Rule-based score: 83.33

Dimension D

Checklist answers: `D1: no, D2: partial, D3: partial, D4: no`

Overall explanation: 技術内容自体は概ね対応しているものの、訳文が複数文に分割され、説明調の接続語を多用しているため、特許請求の範囲としての単一文・形式要件を満たさず、法的明確性（前記関係の厳密さ）も弱まっている。

Rule-based score: 25.00

Dimension E

Checklist answers: `E1: yes, E2: partial, E3: partial`

Overall explanation: 文法的には概ね良好で内容も追える一方、請求項としては文分割と直訳調表現により、クレームの一体性・限定関係の明確さがやや弱まっている。

Rule-based score: 66.67

Score Summary

A–C average: 94.44

D–E average: 45.83

Gap: 48.61

Human evaluation score: 45

Table 7: An EN–JA Case Where the Average Single-Dimension Score for Dimensions A–C Is Higher than That for Dimensions D–E.

Source Text

1. A method for unmanned aerial vehicle handover, comprising: determining, by a source base station, that one or more base stations meet one or more handover conditions based on a measurement report from an unmanned aerial vehicle; determining, by the source base station, a candidate base station that has completed a handover preparation among the one or more base stations meeting the one or more handover conditions; handing over, by the source base station, the unmanned aerial vehicle to the candidate base station meeting the one or more handover conditions; determining, by the source base station, one or more accessible base stations based on a flight path of the unmanned aerial vehicle and geographical location information of the one or more accessible base stations; selecting, by the source base station, a base station from the one or more accessible base stations based on a flight speed, a flight altitude, or the flight speed and the flight altitude of the unmanned aerial vehicle, wherein the selected base station is to perform a handover preparation operation, and the accessible base station is a base station that the unmanned aerial vehicle is able to access when flying according to the flight path; performing, by the source base station, a handover preparation operation for handing over the base station that is to perform the handover preparation operation; and determining, by the source base station, the base station that has successfully performed the handover preparation as the candidate base station that has completed the handover preparation.

MT Output

１つ又は複数の基地局が無人航空機からの測定レポートに基づいて１つ又は複数のハンドオーバ条件を満たすことを決定することと、前記１つ又は複数のハンドオーバ条件を満たす１つ又は複数の基地局のうち、ハンドオーバ準備を完了した候補基地局を決定することと、前記１つ又は複数のハンドオーバ条件を満たす前記候補基地局に前記無人航空機をハンドオーバすることと、前記無人航空機の飛行経路及び１つ又は複数のアクセス可能な基地局の地理的位置情報に基づいて、前記１つ又は複数のアクセス可能な基地局を決定することと、前記無人航空機の飛行速度、飛行高度、又は前記飛行速度及び前記飛行高度に基づいて、前記１つ又は複数のアクセス可能な基地局から１つの基地局を選択することであって、前記選択された基地局は、ハンドオーバ準備動作を実行する基地局であり、前記アクセス可能な基地局は、前記飛行経路に従って飛行するときに前記無人航空機がアクセス可能な基地局である、選択することと、前記ハンドオーバ準備動作を実行する基地局に対してハンドオーバ準備動作を実行することと、前記ハンドオーバ準備を完了した前記候補基地局として、前記ハンドオーバ準備を正常に完了した基地局を決定することと、を含む、無人航空機ハンドオーバ方法。

Dimension A

Checklist answers: `A1: partial, A2: partial, A3: yes`

Overall explanation: The translation largely preserves the claimed method steps and technical dependencies, but it weakens fidelity by dropping explicit attribution to the source base station and by slightly distorting the phrasing of the handover-preparation step, which could impact claim clarity/scope.

Rule-based score: 66.67

Dimension B

Checklist answers: `B1: partial, B2: partial, B3: yes`

Overall explanation: The Japanese translation largely covers the claim steps and preserves key conditions and quantifiers, but it weakens legal precision by omitting the repeated limitation that actions are performed by the source base station and by introducing some ambiguity in the handover-preparation-for-handover relationship.

Rule-based score: 66.67

Dimension C

Checklist answers: `C1: partial, C2: partial, C3: partial`

Overall explanation: The translation largely preserves the claim' s overall structure and major limitations, but a critical mid-to-late step concerning the handover preparation operation is rendered with an altered object/relationship, and the 'wherein' attachments are somewhat entangled, potentially affecting claim logic and scope.

Rule-based score: 50.00

Dimension D

Checklist answers: `D1: yes, D2: yes, D3: partial, D4: yes`

Overall explanation: 全体として単一文の方法クレームとしての体裁は保たれており、列挙構造も適切です。一方で「前記」の付与が初出要素に及ぶ箇所があり、先行詞の厳密性という特許請求項特有の要件に軽微な弱点があります。

Rule-based score: 87.50

Dimension E

Checklist answers: `E1: partial, E2: yes, E3: partial`

Overall explanation: 特許請求項としての体裁と用語運用は概ね適切で、全体の意味も追える一方、いくつかの箇所で係り受け・目的語関係が不自然で可読性と明確性がやや低下している。

Rule-based score: 66.67

Score Summary

A–C average: 61.11

D–E average: 77.08

Gap: -15.97

Human evaluation score: 65

Table 8: An EN–JA Case Where the Average Single-Dimension Score for Dimensions D–E Is Higher than That for Dimensions A–C.

Machine-Translated Bengali in the Wild: A Provenance Benchmark and a Detector-Flagged Shift Toward Translation in Common Crawl

Mantaqa Abedin
BRAC University
Dhaka, Bangladesh
mantaqa.abedin@g.bracu.ac.bd

Jeba Afia Khan
BRAC University
Dhaka, Bangladesh
jeba.afia.khan@g.bracu.ac.bd

Abstract

Low-resource Asian machine translation draws on monolingual web text from Common Crawl. We ask what share of Bengali Common Crawl a detector flags as machine output, and whether it moves with translation or generation. We introduce BanglaGen-Bench, a 2,700-document Bengali benchmark with human-written, English-to-Bengali machine-translated and LLM-generated classes, and a stylometric detector that needs no Bengali-specific tooling. Translatedness did not transfer between the two NMT systems we tested: a detector trained only on NLLB-200 output recognises mBART-50 output 5.67% of the time, and the reverse 0.00%, while mixed-system training reaches 99.3% on held-out documents of a seen system. A detector trained on dedicated NMT catches an LLM prompted to translate 20 times in 100, against 97 when the same model generates. Across four snapshots (2018–2026), the detector-flagged non-human share rises from 2.13% to 6.20%, and after 2023 the flagged movement is on the translation side: the MT-flagged share climbs from 0.30% to 3.48%, while the LLM-flagged share peaks and falls. These are flagged shares, not validated prevalence: on the only independent Bengali corpus available, which has no translated text, the detector scores **62.7%** against 97.3% in-benchmark. The benchmark, checkpoints and scripts will be released.

1 Introduction

Machine translation for Asian languages is built on monolingual web text. Back-translation converts target-side monolingual data into synthetic parallel data (Edunov et al., 2018); bitext mining scores candidate pairs drawn from crawls; and the pre-trained models that initialise low-resource systems are trained on Common Crawl derivatives such as CCNet (Wenzek et al., 2020), OSCAR (Abadji et al., 2022) and MADLAD-400 (Kudugunta et al., 2023). For Bengali and its Indic neighbours, where curated parallel data is scarce, the dependence is heavier still: Samanantar (Ramesh et al., 2022) and IndicTrans2 (Gala et al., 2023) both lean on web-derived material, as does NLLB-200 (NLLB Team et al., 2024).

That pipeline closes on itself if the web is already machine-translated. Thompson et al. (2024) show, using multi-way parallelism, that a large share of web text is MT output and that the effect is worst exactly where data is scarcest. Training on such data is a form of self-consumption (Shumailov et al., 2024), and the translationese it carries is known to distort both models and the test sets used to evaluate them (Riley et al., 2020; Zhang and Toral, 2019; Bogoychev and Sennrich, 2019). Audits of web-crawled multilingual corpora have documented serious quality problems in low-resource languages (Kreutzer et al., 2022; Caswell et al., 2020), but none measures the machine-translated share of a single Asian language across the period in which neural MT and instruction-tuned LLMs both reached scale.

Bengali is an instructive case. It has more than 270 million speakers and an active web presence, it is a target language in every major Indic MT effort, and yet no resource separates human-written, machine-translated and directly LLM-generated Bengali text. Without that separation one cannot tell whether a rise in machine-produced web text is an MT problem or a generation problem – and for a translation workshop those are different problems with different remedies. The gap is instrumentation: before proposing filters for MT training corpora, we need to be able to measure what is in them.

Research questions. **RQ1** (Sections 3–4): can human-written, machine-translated and LLM-generated Bengali be told apart when provenance is known? We introduce **BanglaGen-Bench**, a labelled three-class benchmark with a disclosed multi-system mix per synthetic class, to be released

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 104–118
November 9, 2026 ©2026 Association for Computational Linguistics

with checkpoints and code. **RQ2** (Section 5): does learned translatedness transfer across MT systems, and across one model's roles as generator and as translator? Here every label is known, and transfer fails twice. **RQ3** (Section 6): how do detector-flagged shares of Bengali Common Crawl change from 2018 to 2026, and do they move toward translation or generation? RQ3 concerns predicted labels, not verified provenance; Section 7 reports the external check that bounds it. None of these questions is new on its own (Section 2); what is new is asking them together, for one low-resource Asian language, with translation and generation kept apart.

What this paper measures, and what it does not. No oracle exists for whether a given web document was machine-produced. Our detector is trained on *constructed* classes – text we translated and text we generated – and applied to text whose provenance nobody knows, so everything downstream estimates the share of documents *a particular detector flags*, not prevalence; we use *detector-flagged share* in that literal sense. The gap is real: on the only independently labelled Bengali corpus we could obtain, the detector scores 62.7% against 97.3% in-benchmark, and that corpus has no translated text (Section 7), so we call the trend preliminary. The cross-system results of Section 5 do not share this limit, because there we produced every document.

2 Related Work

Translationese and its detection. That translated text is systematically distinguishable from originally authored text is a long-standing result (Baroni and Bernardini, 2006; Koppel and Ordan, 2011; Rabinovich and Wintner, 2015), established mostly on clean, parallel-aligned corpora in high-resource European languages. Our setting differs on three axes at once: noisy extracted web text, a low-resource Asian language with no comparable resources, and a three-way taxonomy that separates translation from direct generation. The last matters because a two-class human-versus-AI scheme cannot see the flagged composition shift we report in Section 6.2.

Machine-translated text in MT training data. Translationese in synthetic data measurably changes what a system learns (Edunov et al., 2018; Bogoychev and Sennrich, 2019) and what evaluation measures (Zhang and Toral, 2019); tagging it explicitly helps (Riley et al., 2020). Thompson et al. (2024) is the closest prior work to our measurement: they detect MT output on the web through multi-way parallelism across languages, and find low-resource languages worst affected. Our approach is complementary and independent in its failure modes – monolingual stylometry within one language, tracked over time, and separating MT from LLM generation, neither of which multi-way parallelism addresses. Marie et al. (2021) is the methodological precedent we try to follow in reporting what failed.

Crawl quality for low-resource languages. Kreutzer et al. (2022) audit web-crawled multilingual datasets and find substantial mislabelling and low-quality content, concentrated in exactly the languages that most need the data; Caswell et al. (2020) document the language-ID failures underneath. These audits are cross-sectional. He et al. (2026) find off-the-shelf detectors weak on LLM-dominant crawl and web data, which is one motivation for our multi-method audit; Longpre et al. (2024) study data age, coverage and quality; Cheng et al. (2025) survey contamination detection more broadly.

Indic and Bengali resources. IndicCorp (Kakwani et al., 2020), Samanantar (Ramesh et al., 2022), IndicTrans2 (Gala et al., 2023), MADLAD-400 (Kudugunta et al., 2023) and BanglaBERT (Bhattacharjee et al., 2022) filter for language, duplicates and quality, but none filters for, or reports, machine-produced source text. That is the practical gap this paper addresses.

Machine-generated text detection. DetectGPT (Mitchell et al., 2023) and Binoculars (Hans et al., 2024) use probability curvature and cross-model perplexity; both need a strong target-language reference model, which is harder to obtain for Bengali, whereas our stylometric method needs none. Wu et al. (2025) identify out-of-distribution generalisation and weak evaluation as the open problems in this area, and our Section 7 is an instance of both. MULTITuDE (Macko et al., 2023) and RAID (Dugan et al., 2024) cover multilingual and adversarial robustness but not Bengali. Ansary (2025), Islam et al. (2025) and Ammar et al. (2025) report 91–99% on curated two-class Bengali and Urdu detection; none separates translation from generation or tests noisy crawl text. Dycke et al. (2026)

warn that narrow AI classes fail on messier real AI text, a limitation our classes share and our external evaluation confirms.

3 BanglaGen-Bench

All three classes are built from, or matched to, the same pool of Bengali Wikipedia article titles, so that provenance rather than topic is the primary axis of variation. Table 1 summarises the composition.

Human-written class. Human text comes from two sources with strong prior odds against machine authorship: a stratified, fastText-filtered sample of `CC-MAIN-2018-17`, which predates instruction-tuned LLMs at web scale, and randomly drawn current Bengali Wikipedia articles, whose editorial norms make wholesale LLM authorship comparatively rare (576 and 324 documents respectively). Neither is assumed to be free of machine content, and Section 6.3 treats that assumption as the load-bearing one it is.

Machine-translated class. The mt class was produced by translating extracts of English Wikipedia articles into Bengali (en $\rightarrow$ bn), matched to the shared title pool through interlanguage links, initially with NLLB-200 distilled 600M (NLLB Team et al., 2024) alone. Each mt document is therefore a single translation step from an English source, the same kind of operation that puts machine-translated Bengali on the web; no round trip through a pivot language is involved. That configuration failed, in a way that turned out to be the most transferable result in the paper; Section 5 reports it. The final class mixes three systems, each translating English extracts from topic-disjoint title sets.

LLM-generated class. The llm class was directly prompted, which separates generation from translation at the point of construction. The disclosed mix uses Llama 3.1 8B (Llama Team, AI @ Meta et al., 2024) and Qwen2.5 7B (Qwen Team, 2024), yielding 308 Llama and 592 Qwen documents. Llama only partly fit the available 4GB GPU while Qwen2.5 fit and ran roughly 3.4$\times$ faster, which produced the uneven split; the constraint is disclosed rather than presented as a design choice.

Crawl samples and corpus integrity. Four snapshots run through an identical fetch, clean and dedup pipeline: `CC-MAIN-2018-17` as a pre-LLM baseline, `CC-MAIN-2023-06` as an early post-ChatGPT point, `CC-MAIN-2024-30` as an intermediate one and `CC-MAIN-2026-30` as the current one. Two integrity passes collapse within-snapshot near-duplicates (Jaccard 0.5; a government-portal boilerplate family disproportionately affected 2018) and exclude evaluation documents matching or near-duplicating benchmark construction data. Final sizes are 6,579 / 9,752 / 12,281 / 9,823 documents over 2,453 / 3,993 / 4,793 / 4,435 domains. Domain, not document, is the statistical unit throughout (Section 6.4).

Class	Source system / origin	n
human	`CC-MAIN-2018-17` sample	576
	Bengali Wikipedia	324
mt	NLLB-200 distilled 600M	626
	mBART-50	204
	Mistral 7B (training only)	70
llm	Qwen2.5 7B	592
	Llama 3.1 8B	308
Total		2,700

Table 1: Benchmark composition. All mt documents are extracts of English Wikipedia articles translated into Bengali in a single step (en $\rightarrow$ bn). The multi-system mix is a consequence of Section 5, not a design preference: a single-system mt class does not generalise. The 70 Mistral translations were added after the unseen-system test of Section 5.2 and are training-only, so no held-out-system claim is made for Mistral in the deployed detector.

4 Detector

Stylometric baseline. The deployed detector is a linear SVM over word-boundary-aware character n-grams ($n \in \{2, 3, 4\}$), TF-IDF weighted. It is deliberately language-agnostic: it requires no Bengali morphological tooling and no reference LLM at inference, which is what makes it portable to other low-resource Asian languages. It reaches 97.28% test accuracy on the 405-document test split and 95.5%$\pm$0.7% under five-fold cross-validation. Class balancing left 600 already-collected documents (200 per class) unused; the deployed model scores 96.67% on that disjoint holdout, with per-class recall within 1.5 points of the primary test, so the confusion matrix is not an artifact of one split.

A quality-estimation signal, explored and demoted. Following Wastl et al. (2025) and unsupervised quality estimation (Fomicheva et al., 2020), we also tried an MT-native feature: the NLLB-200 log-probability of back-translating each

document's leading span to English. It adds 0.99 test points, but McNemar's exact test finds the gain not significant ($p = 0.21875$), so it is used for no result here (Appendix A.1).

Calibration. Test accuracy is near 100% in the top nine decision-margin deciles and 78.0% (n=41) in the lowest; Platt-scaled Brier 0.022 against a constant baseline's 0.026.

Cross-architecture checks. Two transformer encoders were fine-tuned on the identical split and applied to the same snapshots under the same leakage guard. BanglaBERT (Bhattacharjee et al., 2022) is Bengali-pretrained and reaches 98.02% test accuracy. XLM-R-base (Conneau et al., 2020) is massively multilingual rather than Bengali-specialised, fine-tuned by LoRA (Hu et al., 2022) with rank-8 adapters (0.32% of parameters trainable), because full fine-tuning exceeded a 4GB GPU.

Why no off-the-shelf baseline. Zero-shot detectors such as DetectGPT and Binoculars need a strong Bengali reference model (Section 2), and published Bengali detectors are two-class, so none can score the mt class without retraining on our classes; the two encoders above are that retrained, like-for-like comparison.

5 Translatedness Did Not Transfer Across NMT Systems

This section reports the results whose provenance labels we control completely, and which therefore do not inherit the epistemic limit of Section 1. Their scope is narrow – two NMT systems, one source language, one domain – so non-invariance is an existence result: it shows that invariance cannot be assumed, not how often it fails.

5.1 Single-system training learns a system

We built the mt class with NLLB-200 alone, as most work of this kind would. Evaluated against held-out mBART-50 (Tang et al., 2021) translations of the same kind of source material, that detector recognised 5.67% of them. Trained the other way round – mBART-50 only, evaluated on NLLB-200 – it recognised none at all (Table 2). Adding 300 topic-disjoint mBART documents to training raised accuracy on 300 held-out mBART documents to 99.3% (298/300): held-out documents of a system seen in training, not an unseen system. (This experiment used a larger mBART-50 pool than the 204 documents in Table 1.)

Training configuration	Evaluated on	Detected
mt class, single system		
NLLB-200 only	mBART-50	5.67%
mBART-50 only	NLLB-200	0.00%
mt class, mixed systems		
NLLB + mBART	mBART-50†	99.3%
llm class, single system		
Llama only	Qwen2.5	97.97%
Qwen2.5 only	Llama	99.35%
a genuinely unseen system, two roles		
llm class, Mistral unseen	Mistral, generating	97/100
mt class, Mistral unseen	Mistral, translating	20/100‡

Table 2: Cross-system generalisation. †Held-out *documents* from a system present in training, not a held-out system; the distinction is load-bearing and we do not elide it. ‡Measured before the 70 Mistral translations of Table 1 entered training. The last two rows are the same unseen model in two roles: generation transfers, translation does not.

The magnitude matters less than the asymmetry with the llm class, where single-system training transfers almost perfectly (97.97% and 99.35% across systems). Whatever the detector learns from one NMT system's output is close to disjoint from what it needs for another's, while whatever it learns from one instruction-tuned LLM largely covers the next. The natural reading, for the two systems we tested, is that "machine-translated" as an empirically learnable category is, at single-system scale, a set of system-specific artifacts rather than a property of the translation process. Detectors, filters and quality-estimation heuristics built on one NMT system's output should be assumed not to hold for another until tested, and corpus-cleaning pipelines that report an MT-filtering step should report which systems that step was calibrated on.

5.2 Dedicated NMT and LLMs-as-translators are different problems

The last two rows of Table 2 are the result with the shortest shelf-life and the largest consequences. Mistral 7B, held out from the llm class entirely, is caught 97 times in 100 when it *generates* Bengali. The same model, prompted to translate fresh English Wikipedia extracts and absent from the mt class at the time, is caught 20 times in 100 when it *translates* into Bengali; half of its translations are labelled llm and 30 of 100 human. One unseen model, two roles, a nearly five-fold difference in detectability.

The mt-class repair of Section 5.1 holds for the

dedicated NMT systems it was trained on and does not extend to an LLM prompted to translate. We then added 70 topic-disjoint Mistral translations to the mt class (Table 1), which repairs this one model only. Mistral thus plays four roles, which we keep apart: unseen generator (97/100), unseen translator before the repair (20/100), source of 70 training translations after it, and one of the paraphrasers of Section 5.3. Only the first two are unseen-system results, and the deployed detector has not been tested on a dedicated NMT system absent from its training data. As prompting displaces dedicated NMT, this blind spot widens without anything in the detector changing, so any estimate of the machine-translated share of web text, ours included, understates LLM-produced translation; the TPR sweep of Section 6.3 treats it as a parameter (Appendix A.3).

5.3 Paraphrase robustness, and what one paraphraser revealed

The standard evasion against text-provenance detectors is paraphrase (Krishna et al., 2023; Hu et al., 2023; Zha et al., 2025). We paraphrased 80 held-out mt- and llm-class documents with three local models (Llama 3.1 8B, Qwen2.5 7B, Mistral 7B) and, on a 16-document subsample, with Claude Sonnet 5, then re-scored everything.

The result replicates the system-specificity of Section 5.1 on the attack side. **Mistral evaded the detector 63.75% of the time**; Llama and Qwen evaded it 0% and 1.3% (n=77–78), so a two-paraphraser evaluation would have concluded the detector was robust. Train-split augmentation with 60 extra paraphrases across the three local models closes Mistral evasion to 0.00%, and expanding the attack to n=247 and n=251 holds Llama and Qwen evasion at 0.40%, tightening one-sided Clopper–Pearson upper bounds to 1.91% and 1.88%. Claude Sonnet 5 was never in the augmentation and is the only paraphraser genuinely held out from selection: evasion falls from 18.75% to 12.5% (n=16), a one-document movement from which we take the direction and not the magnitude.

The repair has a cost, and it runs along the MT/LLM axis again. Generalisation to an unseen LLM generator regresses from 97% to 78%, while mt-side detection of Mistral-translated text moves from 20% to 96% – though Mistral is not fully unseen there, since 70 mt-class training documents are Mistral translations (Table 1), none overlapping this test. Both detector variants will be released: a robustness-for-generalisation trade, not a strict improvement.

Snap.	n	Raw	ACC	Cluster 95% CI
2018	6,579	2.13%	1.50%	[0.21%, 2.32%]
2023	9,752	5.65%	3.63%	[2.30%, 5.72%]
2024	12,281	5.14%	3.11%	[1.74%, 5.26%]
2026	9,823	6.20%	4.20%	[1.92%, 6.31%]

Table 3: Detector-flagged non-human share per snapshot. ACC is the 2018-anchored Adjusted Classify & Count estimate, whose estimand Section 6.3 qualifies; the rightmost column is its 95% interval under a bootstrap that resamples domains rather than documents.

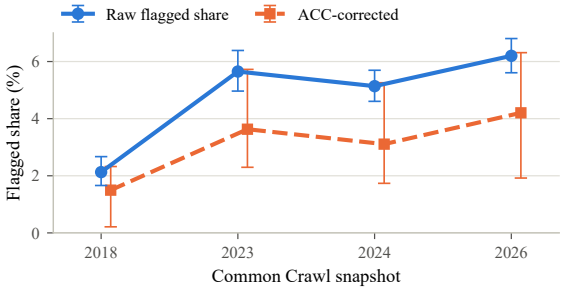


Figure 1: Detector-flagged share by snapshot with cluster-robust 95% intervals, plotting Table 3. The 2018-to-2023 step and the 2023–2026 plateau are the two features every check in this paper tests.

6 Detector-Flagged Shares in Bengali Common Crawl

We now apply the deployed detector to each snapshot and define the **detector-flagged share** as the predicted non-human (mt plus llm) proportion. Everything in this section is subject to the caveat of Section 1 and quantified in Section 7: it measures changes in predicted labels, not verified provenance, and a rising flagged share may reflect change on the web, detector errors that move with the snapshots, or both.

6.1 Rates

The raw rate rises 2.13% → 5.65% → 5.14% → 6.20%, a 2.9× relative increase from 2018 to 2026, or 2.8× under ACC. Every comparison against 2018 is significant (Holm-corrected two-proportion z, $p_{\text{holm}} \approx 0$). Neither 2023-vs-2024 nor 2023-vs-2026 reaches significance ($p_{\text{holm}} = 0.188$); 2024-vs-2026 does ($p_{\text{holm}} = 0.002$), a small naive-rate rise. Cohen's h against 2018 is small ($h \leq 0.21$), as it must be for low proportions: a large relative change in a small absolute share.

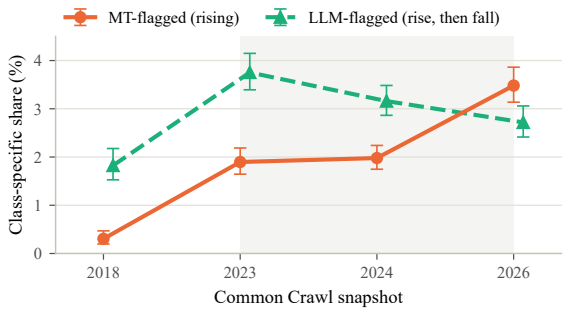


Figure 2: mt-flagged against llm-flagged share (Wilson 95% CI). The shaded 2023–2026 span is where the two move in opposite directions, which the combined rate of Figure 1 hides. Only the mt-side rise survives the matched-length retrain of Appendix A.7.

The plateau is not a power artifact. For the tightest post-2023 pair, 2024-vs-2026, a two-proportion power calculation puts the smallest step detectable at 80% power under the Holm worst-case threshold at 0.83 percentage points, and no observed post-2023 gap approaches it.

6.2 The composition shift the combined rate hides

Tested apart, the two flagged shares move in opposite directions, and for this venue that is the result.

The mt-flagged share rises monotonically (0.30% → 1.90% → 1.98% → 3.48%), is significant against 2018 throughout ($p_{\text{holm}} \approx 0$), is flat across 2023-vs-2024 ($p_{\text{holm}} = 0.663$), and rises in both 2023-vs-2026 and 2024-vs-2026 ($p_{\text{holm}} < 10^{-10}$). The llm-flagged share rises and then falls (1.82% → 3.75% → 3.16% → 2.72%), also significant against 2018, with declines across 2023-vs-2024 ($p_{\text{holm}} = 0.032$) and 2023-vs-2026 ($p_{\text{holm}} < 10^{-3}$) but not 2024-vs-2026 ($p_{\text{holm}} = 0.055$).

The 2024-to-2026 rise in the combined flagged share therefore comes from the mt side. A two-class human-versus-AI scheme would miss this inversion entirely. It also survives the strongest internal challenge we could mount: under the matched-length retrain that replaces the llm class entirely with length-matched documents (Appendix A.7), the mt-side rise persists across seeds while the llm-side movement does not. Read conservatively, we can defend the mt-side rise in what the detector flags, not the llm-side trajectory; whether the web itself holds more machine-translated Bengali, no external corpus available to us can say (Section 7).

6.3 Calibration, and an estimand we do not claim

Classify-and-count is biased under class imbalance whenever per-class error rates differ, which is our situation, so we apply Adjusted Classify & Count (Forman, 2008), inverting the confusion matrix and clipping infeasible negative solutions to zero.

The synthetic test-set false-positive rate is 4.48%. The 2018 snapshot predates both ChatGPT and GPT-2, so it is all but certainly free of LLM-generated text by temporal necessity; it nonetheless flags 2.13% of its documents as non-human (140 of 6,579; 95% CI [1.79%, 2.51%]). The correction uses the same quantity measured on a vintage-retrained detector (Appendix A.6), 2.10% (138 of 6,579); the FPR sweep (Figure 3) spans both values and no conclusion changes between them. The correction rescales the human row while preserving the llm:mt misclassification ratio, giving 1.50% / 3.63% / 3.11% / 4.20%; the synthetic-FPR alternative gives 1.13% / 3.02% / 2.43% / 1.99%. With the 2018 anchor every post-2018 year is significant against 2018 ($p_{\text{holm}} \leq 0.004$); with the synthetic FPR only 2023 is. The later years remain mutually indistinguishable ($p_{\text{holm}} \geq 0.09$, rising to ≥ 0.58 under cluster-robust testing). Across 31 swept FPR values from 2% to 5%, 2018 stays below all later years with a minimum 0.86-point gap (Figure 3), well powered throughout (Appendix A.2), though the later years' relative order flips across the sweep, which reinforces that they are not separable.

The anchor is class-asymmetric, and we do not pretend otherwise. Chronology rules out LLM generation in 2018. It does not rule out machine translation. Statistical MT, templated automation and text-spinning all predate 2018 and would have been on the Bengali web then. The two components of the 2.13% are therefore not equally well identified. Any 2018 document flagged `llm` must be a false positive, so that component is a direct measurement of the human-to-llm error rate. A 2018 document flagged `mt` may be a correct detection of pre-neural machine translation, so that component is only an upper bound on the human-to-mt error rate, and treating all of it as error subtracts real signal along with noise.

Two consequences follow. First, the combined ACC figure has a mixed estimand and should be read as a bound rather than a point estimate; using an FPR that is too high depresses every corrected rate, so the levels in Table 3 are conservative, and

the FPR sweep shows the ordering is insensitive to the choice anywhere in 2–5%. Second, the 2026 end of the mt-side trend is understated for an unrelated reason: LLM-produced translation was detected at 20 in 100 before the targeted Mistral repair (Section 5.2). A properly identified estimate would calibrate the two classes separately against Bengali text of known provenance; no such corpus was available to us, and we do not claim that number.

Bounding the sensitivity side. No real-text oracle grounds the true-positive rate either, so we bound rather than assume it. The llm-class synthetic TPR of 100% updates to the third-party estimate of 97% directly. The mt-class synthetic TPR of roughly 96% diverges sharply from the 20% measured on Mistral-translated text, but that 20% is the mechanism-specific worst case of Section 5.2 rather than a representative real-world rate, so we sweep the mixing share $w \in [0, 1]$ of real-world MT behaving like the worst case. The finding holds across the large majority of the range and crosses zero only near $w \approx 0.85$, which would require more than 85% of real Bengali web MT to come from prompting a general-purpose LLM rather than from cheaper, faster dedicated NMT (Appendix A.3).

6.4 What survives

Web documents from the same domain are not independent, so document-level intervals overstate precision. We resample each snapshot's *domain list* over 5,000 iterations with a shared per-iteration confusion-matrix draw (Appendix A.4). Naive intervals widen by 26–55%; ACC-corrected intervals by only 2–5%, because confusion-matrix resampling noise from the 405-document test set already dominates their width. Every significance conclusion survives: 2018 stays below all later years ($p_{\text{holm}} < 0.004$), and among the later years only 2024-vs-2026 separates, and only on naive rates: under cluster-robust ACC it does not ($p_{\text{holm}} = 0.58$).

Both alternative architectures place 2018 significantly below every later year and agree on the shape – flat 2023–2024, higher 2026 naive rates – while disagreeing substantially on level (BanglaBERT 1.08% / 4.40% / 4.01% / 4.90%; XLM-R-LoRA 7.39% / 11.25% / 10.56% / 12.65%). We therefore treat no architecture's absolute level as authoritative and read only the shape. Per-document agreement on the three-class label is only fair by Cohen's κ (0.20–0.28 against BanglaBERT, with 92–97% identical labels; 0.13–0.24 against XLM-R-LoRA, with 86–92%), which is what κ does at this prevalence: chance agreement alone is 82–97% and unequal marginals cap the attainable value well below unity (Appendix A.5). This is mutual consistency, not independent validation.

Two independent retrains on generators unseen during the original work – Claude Sonnet 5 (24 training, 16 held out) and Llama 3.2 3B (79 training, 60 held out) – reproduce the original rates to within 0.07 and 0.29 points with the identical significance pattern (Appendix A.6): the detector is not of the wrong vintage, though the classes remain constructed.

Corpus-quality metrics (type-token ratio, duplicate 5-grams, a compression proxy (Shaib et al., 2025)) move toward lower diversity, and a classifier two-sample test (Lopez-Paz and Oquab, 2017) separates our human class from the live snapshots increasingly well (AUC 0.68 → 0.93) – equally consistent with our training data drifting out of distribution. Movement is established; its cause is not.

No method applied to 2018-vs-2026 reverses its direction (Appendix A.8); only the llm-only per-class rate fails the matched-length retrain.

A regional split. By top-level domain, Bangladesh-hosted (`.bd`) content is flagged more often than India-hosted (`.in`) content in every testable year (11.11% against 4.19% in 2026; $p_{\text{holm}} < 0.001$ in 2023, 2024 and 2026). Our human class is drawn mostly from `.bd` sources, so we trust this only because calibration, leave-one-domain-out and organisation-level checks all hold (Appendix A.11); it covers only the 10.4% of documents with a country TLD (Limitations).

A finding that failed its own stress test. A flagged-share spike in the ChatGPT launch window survived class-composition and top-k domain checks but dissolved under inverse-propensity domain reweighting (Appendix A.12): single-crawl estimates can manufacture temporal spikes.

7 The External Check, and What It Costs the Trend

Every result in Section 6 tests internal consistency; none tests the classes against independently sourced labels. This section does, and its outcome is the most important caveat in the paper.

Genre	n/class	Human	AI
Newspaper Article	159	98.7%	38.4%
Scientific Abstract	81	18.5%	81.5%
Email	27	33.3%	63.0%
Poem	15	60.0%	100.0%
Short Story	15	60.0%	93.3%
Office Order	3	33.3%	100.0%

Table 4: Per-genre accuracy on the independently labelled dataset of Ansary (2025). *Human* is the share of human-labelled documents predicted human; *AI* the share of AI-labelled documents predicted mt or llm. Office Order, at three documents per class, is too small to weigh alone.

Among external Bengali provenance corpora we could locate, a 40,000-document Zenodo corpus is embargoed until December 2026 and the rest are paywalled or lack a peer-reviewed description. The dataset of Ansary (2025) is public: 600 documents, 300 human and 300 ChatGPT-generated, matched by genre, independently labelled and tied to a peer-reviewed paper. Its labels are binary where our detector predicts three classes and it contains no translation content, so it speaks only to the human-versus-llm axis. Taking the most permissive fair reading of correctness for an AI-labelled document – predicted mt *or* llm – human recall is 66.7% (200/300), AI recall 58.7% (176/300) and binary accuracy **62.7%** (376/600), against 97.28% in-benchmark.

The two failure modes are mechanistically distinct (Table 4). Human-authored Scientific Abstracts carry a dense register of English-loanword technical terms transliterated into Bengali script, entirely absent from our human class's Wikipedia and CC-2018 distribution, which plausibly explains why genuine human text in that genre is so often called non-human. For Newspaper Article – the genre closest both to our training register and to real web content – the low 38.4% AI-catch rate is not our length confound working in the generator's favour: the ChatGPT newspaper text is *shorter* on average than the paired human text (637 against 1,586 characters), which our documented length sensitivity should make easier to catch. It is a style-generalisation gap against an unseen generator. Of the AI-labelled documents caught, 161 of 176 (91.5%) are predicted llm rather than mt, as expected without translation.

What the check cannot reach. The corpus contains no machine-translated text, so it tests only the human-versus-non-human axis under distribution shift and gives no external support to the translation side of Section 6.2. We know of no independently labelled corpus of machine-translated Bengali web text; building one is the most useful next step.

What this permits and forbids. It forbids reading Table 3 as prevalence: internal agreement cannot repair a 62.7% transfer rate, since the internal checks share the construction bias that causes it. It permits the weaker claim we make: a fixed detector, held constant across four identically processed snapshots, flags a rising share over time, and the flagged rise decomposes toward the mt class. It does not establish a translation-driven shift in the true composition of the web. A detector can be badly calibrated in level and still informative in difference, provided its bias does not move with the snapshots – an assumption the C2ST result of Section 6.4 partially undermines. Hence we call the trend preliminary.

8 Implications for Asian-Language MT Resources

Three consequences follow for anyone building MT resources from Asian web text. **MT filters must disclose their calibration systems**: a detector trained on one NMT system's output recognised another's at 5.67% and 0.00% (Section 5.1), so pipelines that filter machine-translated text should name the systems the filter was calibrated on, as they already state language-ID thresholds. **LLM translation is the uncovered case**: filters calibrated on dedicated NMT miss it (Section 5.2). **The monolingual side may be moving**: if even part of the rise in the mt-flagged share, from 0.30% to 3.48%, is real, target-side data for Bengali back-translation holds a growing share of MT output – the self-consumption setting of Shumailov et al. (2024), to which synthetic-data quality is sensitive (Bogoychev and Sennrich, 2019) – and none of the corpus pipelines of Section 2 measures it.

Limitations

No independently sourced ground truth for real web text. This is the central epistemic limit, and it is stated in the abstract, in Section 1 and in Section 7 rather than here alone. All detectors in this paper train on constructed classes; the external check gives 62.7% against 97.28% in-benchmark, and it covers only the human-versus-non-human axis: with no translated text in it, the mt class and

the translation side of the trend have no external validation. Vintage retrains address the objection that the detector is out of date. They do not address the objection that the classes are constructed.

A held-out-system gap in both directions. Single-system MT training fails in both directions (0.00% and 5.67%), unlike single-system LLM training (97.97%, 99.35%), and that non-invariance rests on two NMT systems. The 99.3% mixed-training result is held-out documents, not a held-out system, and an unseen further NMT system may fail similarly. The mt-side repair holds for the dedicated NMT systems it was trained on, not for an LLM prompted to translate; the later Mistral repair covers one model only, and the deployed detector has not been tested on a dedicated NMT system absent from its training data.

What the mt class models, and what it may confound. Every mt document is a single en → bn translation of an English Wikipedia extract. Machine-translated Bengali on the web may also start from source languages other than English, from registers other than encyclopaedic prose, and from systems we did not use, commercial MT among them; the system-specificity of Section 5 is shown for one source language, one domain and two NMT systems. Title matching aligns topic across classes but not content origin: mt documents carry content first written in English, while human documents are native Bengali, so part of what separates mt from human may be English-origin content rather than translation artifacts.

Evaluations not reported here. We report the three-class model throughout; the external check of Section 7 reads it as two-class by collapsing its mt and llm predictions, and a directly trained human-versus-non-human detector is not reported. We also do not report how source titles are distributed across training, development and test splits; a title-grouped re-split would rule out topical leakage between them. Section 4 explains why no off-the-shelf detector is compared.

A length confound, partly resolved. The llm class has a fixed generation cap and is shorter than the others; a length-only classifier reaches 69.4%±1.8%, above chance, so length contributes without dominating. Full-class length parity was measured to be infeasible on the available hardware, so we regenerated 100 documents at a median 7.2× the original length; 97 of 100 are still classified llm, with no reciprocal false-positive rise among human and mt documents in that range (97.0% and 99.1% still correct). This supports inference-time robustness and does not remove the training confound. Crawl document length itself rises from 2018 to 2026, which should bias the measured trend downward (Wang et al., 2025).

The regional finding covers a minority of the corpus. Clear country TLDs cover 10.4% of the pooled corpus (8.5–15.9% per year). Targeted index-guided collection grew every `.in` cell (79→81; 432→448; 653→682; 406→423) while changing the rate by at most 0.068 points, and 2018 gained two documents from 25 independently confirmed segments, which indicates a structural ccTLD ceiling rather than a sampling failure (Appendix A.9). Proxy geolocation would add bias: hosting location is not authorship under a CDN, and language is not country. ccTLD documents are shorter (3,560 against 4,534 characters), carry lower human-classification margins (1.76 against 1.99) and are far more government-heavy (33.1% against 3.4%), as expected when `.gov.bd` and `.gov.in` structurally require a country TLD. The finding applies to that subpopulation.

Small and partly circular robustness samples. The paraphrase battery cannot exclude true evasion near 6% for models showing none at n=80, though the expansion to $n \approx 250$ tightens this to under 2%. Augmentation and attack data share the three local models; only Claude Sonnet 5 is genuinely held out from selection, at n=16. Extraction noise leaves human predictions nearly unchanged (0.0–0.2% flips at 5% character noise) but makes mt and llm predictions fragile (22.2% and 20.0%), which is consistent with the thinner minority-class decision margins found in Section 4.

Scale and sampling. Four time points give a coarse rather than continuous trend, and 900 benchmark documents per class from two or three systems each is small. Language ID is stable across snapshots (mean confidence 0.984–0.989, 2.4–3.4% borderline), and restricting the trend to the roughly 90 domains common to all four snapshots preserves the shape while losing power, which inverse-propensity weighting recovers at the same conclusion (Appendix A.10).

Ethics Statement

All text comes from public web sources (Common Crawl) and from Bengali and English Wikipedia, consistent with their terms of use for research. We do not target, collect or release personal or identifying information.

We are conscious of the dual-use potential of provenance detection for a low-resource language. A detector flagging likely machine-translated or LLM-generated Bengali could be repurposed to suppress or deprioritise non-native speakers, people using translation and writing tools as an accessibility aid, or Bengali communities whose style diverges from our training distribution. The external evaluation in Section 7 makes the risk concrete rather than theoretical: 81.5% of genuine human-authored Bengali scientific abstracts in an independent corpus were flagged as non-human by our detector, because their register is absent from our human class. We will release BanglaGen-Bench and the detector as research artifacts for corpus analysis and filtering research, and we caution explicitly against deployment in content moderation or individual-authorship judgements, given AI-detector false-positive risks and the specific risk of penalising legitimate non-native or assistive use of translation and writing tools.

Acknowledgments

We thank the three anonymous reviewers, whose comments shaped the research questions of Section 1, the scope statements of Sections 5 and 7, and the Limitations. We also thank Farig Yousuf Sadeque (BRAC University) for his contributions to this work.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.

Muhammad Ammar, Hadiya Murad Hadi, and Usman Majeed Butt. 2025. AI-generated text detection in low-resource languages: A case study on Urdu. *arXiv preprint arXiv:2510.16573*.

Md. Siam Ansary. 2025. A novel dataset for identifying AI generated Bengali texts. In *2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN)*.

Marco Baroni and Silvia Bernardini. 2006. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuxing Cheng, Yi Chang, and Yuan Wu. 2025. A survey on data contamination for large language models. *arXiv preprint arXiv:2502.14425*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. ArXiv:2405.07940.

Nils Dycke, Marina Sakharova, Nico Daheim, and Iryna Gurevych. 2026. "Your AI Text is not Mine": Redefining and evaluating AI-generated text detection under realistic assumptions. *arXiv preprint arXiv:2606.04906*.

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

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

George Forman. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.

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 languages of India. *Transactions on Machine Learning Research*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Rounak Saha, Yuxin Wen, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Sichang Steven He, Calvin Ardi, Ramesh Govindan, and Harsha V. Madhyastha. 2026. DeGenTWeb: A first look at LLM-dominant websites. *arXiv preprint arXiv:2605.00087*.

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

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-text detection via adversarial learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2307.03838.

Md. Rakibul Islam, Most. Sharmin Sultana Samu, Md. Zahid Hossain, Farhad Uz Zaman, and Md. Kamrozzaman Bhuiyan. 2025. Detecting AI-generated paraphrases in Bengali: A comparative study of zero-shot and fine-tuned transformers. *Preprint*, arXiv:2512.21709.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1318–1326, Portland, Oregon, USA. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, and 33 others. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv:2303.13408.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. MADLAD-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*.

Llama Team, AI @ Meta, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, and toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

David Lopez-Paz and Maxime Oquab. 2017. Revisiting classifier two-sample tests. In *International Conference on Learning Representations (ICLR)*.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. 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 7297–7306, Online. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht,

Jean Maillard, et al. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Qwen Team. 2024. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Ella Rabinovich and Shuly Wintner. 2015. Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432.

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.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7737–7746, Online. Association for Computational Linguistics.

Chantal Shaib, Tuhin Chakrabarty, Diego Garcia-Olano, and Byron C. Wallace. 2025. Measuring AI "Slop" in text. *arXiv preprint arXiv:2509.19163*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Brian Thompson, Mehak Dhaliwal, Peter Frisch, Tobias Domhan, and Marcello Federico. 2024. A shocking amount of the web is machine translated: Insights from multi-way parallelism. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1763–1775, Bangkok, Thailand. Association for Computational Linguistics.

Yitong Wang, Zhongping Zhang, Margherita Piana, Zheng Zhou, Peter Gerstoft, and Bryan A. Plummer. 2025. Real, fake, or manipulated? detecting machine-influenced text. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15022–15037, Suzhou, China. Association for Computational Linguistics.

Michelle Wastl, Jannis Vamvas, and Rico Sennrich. 2025. Machine translation models are zero-shot detectors of translation direction. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, Marseille, France.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia S. Chao, and Derek F. Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Yiwei Zha, Rui Min, and Shanu Sushmita. 2025. PADBen: A comprehensive benchmark for evaluating AI text detectors against paraphrase attacks. ArXiv:2511.00416.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

A Supplementary Material

Full derivations and setups for results summarised in the main text. Nothing here changes a conclusion stated in the body.

A.1 Hybrid-signal significance testing

The hybrid concatenates the mean per-token NLLB-200 back-translation log-probability to the n-gram matrix as a translation-confidence feature (Table 5). Its gain over the deployed stylometric model is four net documents on the 405-document test set, concentrated on predicted MT-versus-LLM confusion. McNemar's exact test on the MT-versus-LLM cell involves two discordant pairs ($p = 0.5$); on the full test set, six ($p = 0.21875$). Powering the class-specific test would require an order of magnitude more discordant pairs than this split yields. Rather than bill an unproven effect as a contribution, we report the signal as explored but not adopted.

Model	Dev acc.	Test acc.
Stylometric-only (deployed)	95.80%	97.28%
+ transl.-conf. (explored)	96.30%	98.27%

Table 5: Detector ablation on the 2,700-document benchmark. The stylometric-only model is the one deployed for every result in this paper; the hybrid is reported and demoted, not used.

A.2 Minimum detectable effect across the FPR sweep

The 0.86-point minimum gap of Section 6.3 describes 2018-vs-2026, whereas the 0.83-point floor of Section 6.1 describes 2024-vs-2026; the two are not interchangeable. Recomputing the minimum detectable effect specifically for 2018-vs-2026 at each swept FPR ($n = 6{,}579$ and $n = 9{,}823$) gives a floor of 0.58–0.69 points across the sweep, so the observed gap exceeds its own matched floor by a factor of 1.2–4.0 at every point. Script: `scripts/compute_minimum_detectable_effect.py`.

A.3 TPR-sensitivity sweep

The true-positive rate has no equivalent of the FPR's chronological anchor: it comes entirely from the synthetic test set, and no oracle exists for "this document is definitely machine-produced". We therefore bound it. The llm-class synthetic TPR of 100% is close to the third-party estimate of 97% on Mistral-generated text, so we apply that update directly. The mt-class synthetic TPR of roughly 96% diverges sharply from the 20% measured on Mistral-translated text, but that figure is a mechanism-specific worst case (an LLM prompted to translate) rather than a representative rate for dedicated NMT, on which the mt-class repair holds (99.3% on held-out mBART-50 documents). We therefore sweep $w \in [0, 1]$, the share of real-world mt content behaving like the worst case. The finding degrades gradually and crosses zero only near $w \approx 0.85$. This bounds an unmeasured quantity; it does not measure it.

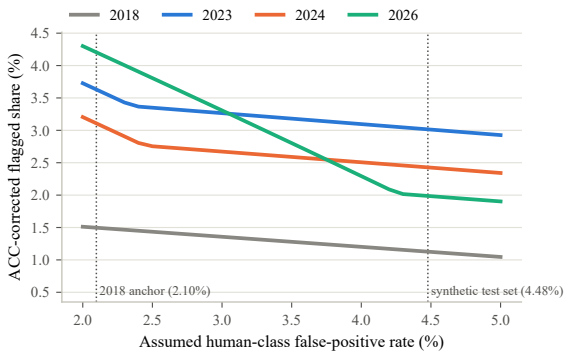


Figure 3: ACC-corrected rate as the assumed human-class FPR sweeps from 2% to 5%. Dotted lines mark the synthetic-test-set (4.48%) and 2018-anchor (2.10%) values. 2018 stays below all three later years throughout; their relative order does not.

A.4 Domain-cluster-robust bootstrap

Each snapshot's domain list is resampled with replacement over 5,000 iterations, routed through the same 2018-anchored ACC correction used for Table 3, with a shared per-iteration confusion-matrix draw across snapshots so paired tests remain valid. Naive-rate intervals widen by 26–55% relative to document-level Wilson intervals; ACC-corrected intervals by 2–5%, because confusion-matrix resampling noise from the 405-document test set already dominates.

A.5 Cross-architecture numbers

BanglaBERT gives 1.08% / 4.40% / 4.01% / 4.90% for 2018 / 2023 / 2024 / 2026 and, like the deployed model, finds 2023-vs-2024 not significant ($p_{\text{holm}} = 0.20$) with 2024-vs-2026 significant ($p_{\text{holm}} = 0.004$). XLM-R-LoRA gives 7.39% / 11.25% / 10.56% / 12.65%, substantially higher in level with the same shape (2023-vs-2024 $p_{\text{holm}} = 0.103$, 2024-vs-2026 $p_{\text{holm}} \approx 0$). Observed three-class κ against the deployed model is 0.20 / 0.26 / 0.28 / 0.28 for BanglaBERT (92.4–97.5% identical

labels) and 0.13 / 0.23 / 0.24 / 0.22 for XLM-R-LoRA (86.2–91.9%), for 2018 / 2023 / 2024 / 2026. Cohen's κ is depressed both by prevalence imbalance and by unequal rater marginals. Fixing each pair's marginals at the rates above, the largest attainable κ is 0.67–0.88 for the deployed model against BanglaBERT, 0.43–0.64 against XLM-R-LoRA and 0.24–0.53 for BanglaBERT against XLM-R-LoRA; we read the observed values against those ceilings rather than against unity.

A.6 Vintage retrains

A frontier model. 24 topic-disjoint training documents authored by Claude Sonnet 5 plus a disjoint 16-document held-out set, produced by direct prompting through the model's chat interface; the sampling configuration was therefore outside our control. *A small local model.* Qwen3 (8B and 4B) proved unusable on a 4GB GPU because its thinking mode consumed the token budget before any output; Llama 3.2 3B instead generated 79 training and 60 held-out documents directly. *Results.* Both retrained detectors classify their held-out documents with 100% accuracy and reproduce the deployed detector's per-snapshot rates to within 0.07 and 0.29 points, with the identical significance pattern (Section 6.4). The ACC correction of Section 6.3 takes its human-class FPR anchor from a vintage-retrained detector: 2.10% of 2018 documents flagged (138 of 6,579), against 2.13% (140) for the deployed detector; the FPR sweep of Figure 3 covers both values.

A.7 Matched-length retrain and rescore

We trained five detectors in which the llm class is replaced entirely by 100 matched-length documents, with human and mt classes drawn as a fresh 100/100 surplus sample per seed, disjoint from the primary benchmark and the calibration holdout. All four snapshots were rescored per seed under a combined leakage exclusion.

Comparison	mt-only sig.	llm-only sig.
2023 vs. 2024	1/5	1/5
2023 vs. 2026	5/5	1/5
2024 vs. 2026	3/5	1/5

Table 6: Seeds (of five) in which each Holm-corrected comparison reaches significance under the matched-length retrain.

The 2018 llm-only rate stays near zero (0.27–0.89%, mean 0.49%). Aggregate mt-only rates are 1.72% $\rightarrow$ 1.64% $\rightarrow$ 2.16% (2023/2024/2026), the original shape; llm-only rates are 0.81% $\rightarrow$ 0.71% $\rightarrow$ 0.81%, essentially flat. A length-binned check is inconclusive at $n \approx 25$–30 per seed, bin and year. Data and scripts: `results/matched_length_retrain_rescore.json`, `scripts/matched_length_retrain_rescore.py`.

A.8 Every method applied to 2018 versus 2026

Table 7 lists every distinct method applied to the 2018-vs-2026 comparison with its primary result.

Check	Method / model	p_{holm}
Raw rate, z-test	Linear (deployed)	≈ 0
Cluster bootstrap	Linear, naive	≈ 0
Cluster bootstrap	Linear, ACC-corr.	0.0032
ACC correction	Linear, 2018 FPR	0.004
Cross-architecture	BanglaBERT	≈ 0
Cross-architecture	XLM-R-LoRA	≈ 0
Per-class rate	mt-only	≈ 0
Per-class rate	llm-only*	6.5×10^{-4}
Vintage retrain	Claude Sonnet 5	≈ 0
Vintage retrain	Llama 3.2 3B	≈ 0
FPR sweep	2%–5%	holds

Table 7: Every distinct method applied to the 2018-vs-2026 comparison, using each method's primary result. These are not independent confirmations: all draw on the same snapshots and several share weights. *Significant in only two of five seeds under the matched-length retrain; every other row holds.

A.9 Targeted `.in` collection

Most of the web uses generic TLDs that do not encode country, so the ccTLD ceiling is largely structural. A Common-Crawl-index-guided fetch targeted WET segments pre-verified to contain `.in` pages (random sampling yields only about 1–4% incidentally). Every year's `.in` cell grew and the rate moved by at most 0.068 points; for 2018, 25 segments each confirmed to contain `.in` content added only two documents. Script: `scripts/tld_covariate_comparison.py`.

A.10 Upstream filtering and domain composition

Re-running the sampling-stage fastText language-ID model on the final corpus gives mean confidence 0.984–0.989 with 2.4–3.4% of documents borderline, stable across snapshots. Between 34.3% and 38.1% of documents have more than 10% non-Bengali characters in their first 2,000 characters, as expected for web text. Restricting the trend to the roughly 90 domains common to all four snapshots

(5.9–8.4% of each) loses significance on most comparisons but keeps the shape; an inverse-propensity-weighted alternative (Kish effective $n = 1{,}978$–$5{,}582$) reaches the same conclusion with substantially more power.

A.11 Regional split: full artifact checks

Cross-tabulating by top-level domain gives an axis native to corpus-building practice. Bangladesh-hosted (`.bd`) content is significantly more flagged than India-hosted (`.in`) content in every testable year: 9.06% against 1.85% in 2023, 10.27% against 3.37% in 2024, 11.11% against 4.19% in 2026, all at $p_{\text{holm}} < 0.001$; 2018 is excluded because its `.in` cell (n=79) cannot power the test. Because the human training class is drawn predominantly from `.bd` sources, three artifact checks preceded any trust in this. **Calibration**: using 2018 as ground truth, false-positive rates are indistinguishable across the two namespaces (1.16% `.bd` against 1.02% `.in`, $p = 0.90$). **Leave-one-domain-out**: dropping each contributing domain in turn preserves the gap across every removal in all three years (533/533, 437/437 and 393/393 iterations significant). **Cluster granularity**: a domain-level test cannot see one operator publishing under several domains, so we re-ran the bootstrap at document, domain and an organisation-proxy granularity that groups domains sharing a registrable base name across TLD suffixes; all nine year-by-granularity combinations remain significant ($p < 0.001$). Support rests on the three together. The scope caveat travels with the finding: documents with a clear country TLD are 10.4% of the pooled corpus, and that subpopulation is shorter and far more government-heavy than the rest (Limitations).

A.12 Onset dating: a reversal

We tried to date the onset by bracketing the ChatGPT and GPT-4 launches with four extra snapshots. The launch-window snapshot, `CC-MAIN-2022-49`, showed a naive spike of 6.24% (3.51% ACC), significantly above its neighbours, and a naive reading would report it as a dateable onset. Class composition ruled out direct ChatGPT authorship at that volume; removing the top three domains moved the rate only to 5.90%, so the spike survived a coarse check. Inverse-propensity domain reweighting (Horvitz–Thompson, weights capped at 10×, Kish effective $n = 2{,}122$–$4{,}856$) dissolved it: 6.24% fell to 4.46% and all pairwise comparisons became non-significant ($p_{\text{holm}} \geq 0.56$). The verdict is a domain-sampling artifact. The transferable point is that narrow-window, single-crawl estimates can manufacture temporal spikes that top-k domain checks miss and inverse-propensity reweighting catches. The four larger trend snapshots show no comparable artifact under identical reweighting.

UTSK26 at WAT2026 Patent Claims Translation/Evaluation Tasks

Yingyi Fu, Kosei Buma, Takehito Utsuro
Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba

Abstract

This paper describes the UTSK26 submission to the WAT2026 Patent Claims Translation/Evaluation Task for the Japanese-to-English (JA–EN) and English-to-Japanese (EN–JA) directions. The task requires participants to annotate translation errors in machine translation (MT) outputs of patent claims and to assign a quality score from 0 to 100 to each translation. For error annotation, we use GEMBA-Patent-Hybrid, which combines an LLM-as-a-Judge method for detecting error spans, categories, and severities using GPT-5.2 with another method for assigning categories and severities to encoder-detected error spans. For quality scoring, we use Patent-Rubric-Checklist, a rubric-based LLM-as-a-Judge method that instructs GPT-5.2 to answer patent-specific checklist items covering five evaluation dimensions and aggregates the answers into a score with predefined weights. On the training data, which consist of the human evaluation results of the WAT2025 task, Patent-Rubric-Checklist achieves the highest Pearson and Spearman correlations with human scores in both directions among the evaluated methods, including COMETKiwi, MetricX-24, GEMBA-DA, and GEMBA-MQM.

1 Introduction

Patent claims are typically written as a single long sentence with complex structures and highly specialized terminology, which makes them difficult to translate. Evaluating their translations is also challenging, because seemingly minor errors, such as the omission of a limitation or an incorrect relationship between claim elements, may substantially change the scope of the claimed invention. In the first Patent Claims Translation Task at WAT2025, Nakazawa et al. (2025) reported that automatic metrics such as COMETKiwi (Rei et al., 2023) and MetricX-24 (Juraska et al., 2024) showed low correlations with human evaluation.

The WAT2026 Patent Claims Translation/Evaluation Task focuses on this evaluation problem. Given patent claims and their MT outputs, participants annotate translation errors with their categories and severities, and assign a quality score to each translation. The ultimate goal of the task is to develop automatic evaluation methods that can accurately assess patent claim translations.

This paper describes the UTSK26 system for both the JA–EN and EN–JA directions. Our system consists of two components:

- **Error annotation:** GEMBA-Patent-Hybrid, which combines two methods: one instructs GPT-5.2 (OpenAI, 2025) to detect error spans, categories, and severities, while the other instructs GPT-5.2 to assign categories and severities to error spans detected by an encoder model.
- **Quality scoring:** Patent-Rubric-Checklist, which instructs GPT-5.2 to evaluate a translation with a rubric of five patent-claim-specific evaluation dimensions and computes the final score from the checklist answers outside the LLM.

2 Task Description

2.1 Task Definition

For each translated claim, participants submit the following evaluation results in the same JSON format as the training data:

- **Error annotation:** Participants mark error spans in the source text or the MT output, and assign each span the lowest-level category of a three-level error hierarchy and a severity of either major or minor. The hierarchy follows the MQM-based criteria (Freitag et al., 2021) adapted for patent translation in WAT2025 (Nakazawa et al., 2025), whose

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 119–133
November 9, 2026 ©2026 Association for Computational Linguistics

	Training		Test	
	JA–EN	EN–JA	JA–EN	EN–JA
Source documents	13	13	2	2
Source claims	28	28	2	3
MT outputs / claim	6	4	5	4
Translated claims	168	112	10	12
Error annotations	1,130	1,017	–	–
Avg. quality score	75.5	66.0	–	–

Table 1: Statistics of the training and test data.

top-level categories are accuracy, fluency, terminology, style, other, source error, and non-translation. For example, an incorrect article is annotated as *Article* under *Accuracy* > *Mistranslation*.

- **Quality scoring:** Participants assign a holistic quality score from 0 to 100 to each translated claim.

2.2 Data

Table 1 shows the statistics of the data. The training data consist of the human evaluation results for a subset of the WAT2025 task data (Nakazawa et al., 2025).[1] For each direction, 28 claims from 13 documents were translated by six (JA–EN) or four (EN–JA) MT systems, and each translation was annotated with error spans and a quality score. The test data consist of 10 JA–EN and 8 EN–JA files, each of which contains a source document and its MT output. The JA–EN files are derived from two single-claim source documents, each paired with five different MT outputs. The EN–JA files are derived from two source documents with three claims in total, each paired with four different MT outputs.

3 System Description

Our system evaluates each translated claim independently. Each line in a text file corresponds to one claim in the test set. Source and MT claims are aligned by line number. The original text is converted into the character-grid format. The grid text is used to construct input to GPT-5.2 for error annotation.

3.1 Error Annotation

We use GPT-5.2 and an encoder-based model for error annotation. For the encoder-based approach, we employ the token-level QE model introduced in our previous work (Buma et al., 2025), using mDeBERTa-v3-base[2] as the multilingual encoder. The model jointly encodes the source sentence and its machine translation, followed by a token-level classification layer that predicts an OK/BAD label for each target token. We then merge consecutive BAD labels into a single error span and record its start and end character offsets. We use three methods for error annotation.

GEMBA-Patent. We input the source and the MT claims to GPT-5.2. The model outputs the error annotations in JSON format, including error sides, tagged error spans, categories, and severities. In particular, for each error, it outputs the full sentence with the error span enclosed by `<ERROR_START>` and `<ERROR_END>` error tags. We then derive the corresponding `[start:end]` offsets from the tag positions. Moreover, we provide patent-specific definitions of the error categories and severities in the prompt. The prompt allows an empty annotation list when no translation error is detected. The detailed prompt for GEMBA-Patent is shown in Figures 1–4 in Appendix A.

Encoder-GEMBA-Patent. We convert the encoder's error spans into the same format as the error spans output by GEMBA-Patent, where for each error, the full sentence is output with the error span enclosed by the error tags. We then input the processed source and MT claims into GPT-5.2. GPT-5.2 assigns an error category and severity to each error. As in GEMBA-Patent, we provide patent-specific definitions of the error categories and severities in the prompt. The detailed prompt for Encoder-GEMBA-Patent is shown in Figures 6–9 in Appendix A.

GEMBA-Patent-Hybrid. For claims whose GEMBA-Patent outputs that do not satisfy our predefined format and validation requirements (invalid outputs), we use the annotations from Encoder-GEMBA-Patent and map the error categories and severities from English to the Japanese labels provided by the task organizers.

Error category. We instruct GPT-5.2 to output error category labels by joining the hierarchy levels with hyphens. We finally map each English error category label to the corresponding lowest-level Japanese category. For example, `"accuracy-mistranslation-numerals/symbols"` is mapped to `"数値・記号"`.

[1] https://github.com/Tzawa/pat-claim-trans

[2] https://huggingface.co/microsoft/mdeberta-v3-base

Labels containing an invalid subcategory combination, such as `"accuracy-mistranslation-inappropriate"`, are mapped to an empty label, `""`.

Severity. For both GEMBA-Patent and Encoder-GEMBA-Patent, we instruct GPT-5.2 to assign either major or minor severity to each error span. A major error changes or obscures the meaning, claim scope, an essential claim element, a technical relationship, or essential claim form, whereas a minor error does not materially affect the meaning or claim scope.

The prompts for these methods were designed mainly with the assistance of ChatGPT. Notably, we provided the patent-specific checklist explained in Section 3.2 to ChatGPT to help design the GEMBA-MQM-based prompts for GEMBA-Patent and Encoder-GEMBA-Patent, including patent-specific definitions of error categories and severities.

3.2 Quality Scoring

For quality scoring, we use Patent-Rubric-Checklist, the same method in our accepted work to appear at WAT2026 (Fu et al., 2026), which is a reference-free LLM-as-a-Judge method based on a rubric specific to patent claim translation evaluation. Inspired by LiTransProQA (Zhang et al., 2025), which evaluates literary translations by having an LLM answer evaluation questions, the method asks an LLM to answer checklist items rather than to predict a score directly.

Rubric. Table 2 summarizes the rubric, which consists of five evaluation dimensions with 16 checklist items in total. **[A] Content Fidelity** evaluates the preservation of the core inventive idea and technical functions or effects, as well as the absence of hallucinated technical content. **[B] Element Coverage** evaluates whether all claim elements are translated and whether element relationships, quantities, ranges, and conditions are preserved. **[C] Claim Scope & Logical Constraints** evaluates the preservation of the claiming scope, the correct attachment of constraint clauses, and the preservation of coordination and modification relations. **[D] Claim Form & Syntactic Requirements** evaluates whether the translation is written as a single-sentence claim, preserves the claim type, maintains a consistent antecedent basis, and avoids narrative or explanatory prose style. **[E] Fluency & Grammaticality** evaluates grammatical correctness, stylistic appropriateness, and readability. The checklist items were mainly developed with the assistance of Copilot, by providing general translation criteria such as semantic equivalence and fluency, as well as patent-specific constraints such as the number of claim elements and their logical relationships, the single-sentence structure of claims, and terminology consistency. The prompt was further refined with ChatGPT.

	Evaluation Dimension	Items	Weight
A	Content Fidelity	3	0.25
B	Element Coverage	3	0.25
C	Claim Scope & Logical Constraints	3	0.25
D	Claim Form & Syntactic Requirements	4	0.15
E	Fluency & Grammaticality	3	0.10

Table 2: Evaluation dimensions of Patent-Rubric-Checklist with the number of checklist items and the weight of each dimension.

Prompt. As shown in Figures 10–12 in Appendix B, the prompt describes the task setting and evaluation procedure, presents the rubric, and specifies the input and output formats. Given a source claim and its MT output, the LLM answers each checklist item with "yes" (no problem), "partial" (minor or weakening problem), or "no" (clear or serious problem), provides a short justification for each answer and an overall explanation, and returns them in JSON format.

Scoring. We map the answers "yes," "partial," and "no" to 1.0, 0.5, and 0.0, respectively. For each evaluation dimension, we average the mapped values of its checklist items to obtain a dimension score, denoted by A through E. The final quality score is then computed outside the LLM with the following weighted formula:

$$\begin{aligned}\text{score} = 100 \times \big(&0.25A + 0.25B \\ &+ 0.25C + 0.15D + 0.10E\big)\end{aligned} \tag{1}$$

The weights in Equation 1 were suggested by Copilot.

3.3 Implementation Details

For both quality scoring and error annotation, newline characters are removed from the model input. For error annotation, the predicted error span offsets are mapped back to the original grid text with newline characters restored.

4 Experiments

4.1 Setup

Since the test data do not include human evaluation results, we validate our system on the training data (Table 1).

Error annotation. We use Encoder-GEMBA-MQM, a GEMBA-MQM-based method, as the comparison method for error annotation. We use the same input format as Encoder-GEMBA-Patent for GPT-5.2. GPT-5.2 then assigns an error category and an error severity to each error. The error categories are modified to match the annotation criteria provided by the task organizers, whereas the error severities are limited to major or minor. The prompt for this method is also designed with the assistance of ChatGPT. The detailed prompt for Encoder-GEMBA-MQM is in Figure 5 in Appendix A.

We compute span-level precision, recall and F1 on the training data, as well as category and severity accuracy over matched spans. A predicted span is considered correct only when its block index, side, and start/end offsets exactly match those of a gold span. Matched spans are counted as true positives, unmatched predicted spans as false positives, and unmatched gold spans as false negatives. Category and severity accuracy are computed only over the matched spans. We compare GEMBA-Patent-Hybrid with Encoder-GEMBA-Patent and Encoder-GEMBA-MQM.

Quality scoring. We compute Pearson's correlation coefficient r and Spearman's rank correlation coefficient ρ between the automatic scores and the human scores of the 168 JA–EN and 112 EN–JA translated claims. We compare Patent-Rubric-Checklist with two reference-free neural metrics, COMETKiwi[3] (Rei et al., 2023) and MetricX-24[4] (Juraska et al., 2024), and two LLM-as-a-Judge methods, GEMBA-DA (Kocmi and Federmann, 2023b) and GEMBA-MQM (Kocmi and Federmann, 2023a). For MetricX-24, which outputs error scores, we negate the scores so that higher is better. GEMBA-DA directly assigns a score from 0 to 100. GEMBA-MQM identifies errors with MQM-style error categories and severities, and we convert its negative raw score s into $100 + 4s$. All LLM-as-a-Judge methods use GPT-5.2. We also conduct permutation tests to compare the correlations of Patent-Rubric-Checklist with those of GEMBA-DA and GEMBA-MQM. In particular, the submitted scores are rounded to the nearest integer, whereas the unrounded scores are used for the correlation analysis.

[3] https://huggingface.co/Unbabel/wmt23-cometkiwi-da-xl

[4] https://huggingface.co/google/metricx-24-hybrid-xl-v2p6

4.2 Results

4.2.1 Results on the Training Set

Error annotation. Table 3 shows the span-level precision, recall and F1 score, as well as category and severity accuracy over matched spans. Because Encoder-GEMBA-Patent and Encoder-GEMBA-MQM use the same error spans, their span-level precision, recall, and F1 are identical. GEMBA-Patent-Hybrid generally shows lower span-level performance than both encoder-based methods in both translation directions, except for precision in JA–EN. We notice that the error annotations of GEMBA-Patent-Hybrid contain a relatively large number of claims with empty annotations. An empty annotation list is allowed by the GEMBA-Patent prompt and indicates that no translation error is detected, as specified in Figure 4. When GEMBA-Patent returns an empty annotation list, the hybrid method treats the output as valid and therefore does not use the annotations assigned by Encoder-GEMBA-Patent. As a result, the hybrid output contains a relatively large proportion of claims with empty annotations. Overall, 33.93% of EN–JA claims and 37.50% of JA–EN claims have empty annotations in GEMBA-Patent-Hybrid, compared with 0.89% and 11.31%, respectively, in the encoder-based methods.

Moreover, in the JA–EN direction, Encoder-GEMBA-MQM achieves the highest category accuracy, while Encoder-GEMBA-Patent achieves the highest severity accuracy. In the EN–JA direction, Encoder-GEMBA-Patent achieves the highest category and severity accuracy.

We also conduct a two-shot experiment to further analyze the effects of adding human-annotated examples and removing the instruction to return empty annotations on GEMBA-Patent. Its setup and results are reported in Appendix C.

Quality scoring. Table 4 shows the correlations between the automatic and human quality scores. Patent-Rubric-Checklist achieves the highest Pearson and Spearman correlations in both directions. For JA–EN, the differences in Spearman correlation

Method	JA–EN Precision	Recall	F1	Category Acc.	Severity Acc.
Encoder-GEMBA-MQM (GPT-5.2)	0.040	**0.028**	**0.033**	**0.406**	0.688
Encoder-GEMBA-Patent (GPT-5.2)	0.040	**0.028**	**0.033**	0.250	**0.719**
GEMBA-Patent-Hybrid	**0.057**	0.011	0.018	0.250	0.667
Method	**EN–JA Precision**	**Recall**	**F1**	**Category Acc.**	**Severity Acc.**
Encoder-GEMBA-MQM (GPT-5.2)	**0.057**	**0.039**	**0.047**	0.375	0.675
Encoder-GEMBA-Patent (GPT-5.2)	**0.057**	**0.039**	**0.047**	**0.500**	**0.700**
GEMBA-Patent-Hybrid	0.108	0.017	0.029	0.353	0.588

Table 3: Error annotation results on the training data.

Method	JA–EN Pearson's r	Spearman's ρ	EN–JA Pearson's r	Spearman's ρ
COMETKiwi	0.2880^{**}	0.2703^{**}	0.1860^{*}	0.1810
MetricX-24 (reference-free)	0.2436^{**}	0.2945^{**}	0.1397	0.1770
GEMBA-DA (GPT-5.2)	0.4685^{**}	0.4210^{**}	0.1996^{*}	0.3017^{**}
GEMBA-MQM (GPT-5.2)	0.5533^{**}	0.4741^{**}	0.2461^{**}	0.3135^{**}
Patent-Rubric-Checklist (GPT-5.2)	$\mathbf{0.6214}^{**}$	$\mathbf{0.5931}^{**}$	$\mathbf{0.3767}^{**}$	$\mathbf{0.4417}^{**}$

$^{*}p < 0.05$, $^{**}p < 0.01$.

Table 4: Correlations between automatic quality scores and human quality scores on the training data.

between Patent-Rubric-Checklist and GEMBA-DA ($p < 0.01$) and between Patent-Rubric-Checklist and GEMBA-MQM ($p < 0.05$) are statistically significant, whereas the differences in Pearson correlation are not. For EN–JA, none of these differences are statistically significant. In addition, all methods show lower correlations for EN–JA than for JA–EN.

4.2.2 Results on the Test Set

The organizers evaluate our submitted system on the test set using gold annotations and gold MQM scores, which are GEMBA-MQM-style scores calculated from gold annotations. Our quality scoring method achieves the highest correlations with human evaluation among the evaluated methods, whereas our error span detection has low recall. Tables 6–7 in Appendix D show the evaluation results of our error annotation method and quality scoring method on the test set as reported by the organizers. For a more detailed evaluation of the submissions, see Nakazawa et al. (2026).

4.3 Submission

We submitted the outputs of the system described in Section 3 for both directions. Because the span-level precision, recall and F1 were calculated after submission, they were not used for system selection. Following GEMBA-MQM, major and minor errors were assigned weights of 5 and 1, respectively, and the negative raw score s was converted to $100 + 4s$. At most five errors per claim were counted, with major errors prioritized over minor errors. The scores used for system selection were computed from the intermediate error annotations before label mapping. We selected GEMBA-Patent-Hybrid because its scores achieved the highest correlation with human scores on the training set among the three evaluated systems. Notably, the choice of scoring process and severity weights was made with the assistance of ChatGPT. Table 8 in Appendix E shows Pearson and Spearman correlations between automatic scores obtained by the three evaluated systems and human scores on the training data.

5 Conclusion

We described the UTSK26 system for the WAT2026 Patent Claims Translation/Evaluation Task. For error annotation, we use GEMBA-Patent-Hybrid, which combines an LLM-as-a-Judge method that detects error spans and assigns categories and severities, with another method that assigns categories and severities to error spans detected by an encoder model. On the training data, it generally showed lower performance than Encoder-GEMBA-Patent and Encoder-GEMBA-MQM. For quality scoring, we used Patent-Rubric-Checklist, which instructs GPT-5.2 to answer patent-claim-specific checklist items and computes a weighted score from the answers. On the training data, it achieves higher correlations with human scores than COMETKiwi, MetricX-24, GEMBA-DA, and GEMBA-MQM in both directions. In future work, we will improve the quality of error annotation, the design of the rubric and checklist items, and optimize the dimension weights to better align with human evaluation.

Limitations

Limited Validation Data We validate our system on the training data, which contain only 28 claims per direction. Therefore, it remains uncertain whether the observed correlations generalize to other patent claims and MT systems.

Encoder-based QE Model The encoder-based QE model used for error annotation was trained on patent-domain data, but its training data did not include patent claims. This may affect the quality of the resulting error annotations.

Stability and Reproducibility We run each prompt only once; therefore, we cannot assess the stability of the LLM outputs. In addition, our quality scoring relies on a proprietary LLM, GPT-5.2, and its performance with open-weight LLMs remains unknown.

Rubric Design The rubric, checklist items, and dimension weights were designed with the assistance of LLM-based assistants and were not optimized on human evaluation data. The rubric is also designed for Japanese–English patent claims and may need to be adapted to other language pairs.

Error Annotation The LLM may output labels containing invalid subcategory combinations, which are mapped to empty labels and may result in missing error category annotations. The GEMBA-Patent-Hybrid method treats an empty annotation list from GEMBA-Patent as a valid output and therefore does not use Encoder-GEMBA-Patent in such cases. As a result, some claims remain without error annotations even when the encoder-based method detects errors. Moreover, because newline characters are removed from the model input (Session 3.3), a space within a predicted error span may be mapped back to a newline character when mapping the span offsets back to the original grid text.

Author Contributions

Yingyi Fu Conducted Patent-Rubric-Chesklist for quality scoring and implemented Encoder-GEMBA-MQM, Encoder-GEMBA-Patent, GEMBA-Patent, and GEMBA-Patent-Hybrid for error annotation.

Kosei Buma Conducted error annotation using an encoder-based model.

Takehito Utsuro Built and managed our team.

References

Kosei Buma, Takehito Utsuro, and Masaaki Nagata. 2025. Two step automatic post editing of patent machine translation based on pre-trained encoder models and LLMs. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 218–231, Mumbai, India. Association for Computational Linguistics.

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.

Yingyi Fu, Takehito Utsuro, and Masaaki Nagata. 2026. Towards a rubric-based llm-as-a-judge for patent claim translation evaluation. In *Proceedings of the Thirteenth Workshop on Asian Translation (WAT 2026)*. In press.

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 and Christian Federmann. 2023a. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023b. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Kazuhiro Kasada, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2025. Findings of the first patent claims translation task at WAT2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 1–15, Mumbai, India. Association for Computational Linguistics.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Taiki Ushimaru, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2026. Findings of the second patent claims translation/evaluation task at WAT2026. In *Proceedings of the Thirteenth Workshop on Asian Translation (WAT 2026)*, Hengqin, China. Association for Computational Linguistics.

OpenAI. 2025. Introducing GPT-5.2. `https://openai.com/index/introducing-gpt-5-2/`.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André F. T. 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.

Ran Zhang, Wei Zhao, Lieve Macken, and Steffen Eger. 2025. LiTransProQA: An LLM-based literary translation evaluation metric with professional question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29099–29121, Suzhou, China. Association for Computational Linguistics.

A Prompt for Error Annotation

Figures 1–9 show the prompts for error annotation.

B Prompt for Quality Scoring

Figures 10–12 show the prompt for Patent-Rubric-Checklist.

C Few-shot Experiment

Setup. We additionally conduct a two-shot experiment for error annotation using one human-annotated claim from each translation direction in the training set as examples in the prompt. For each translation direction, among documents containing only one claim, we select the document that has the largest number of different error categories in its human annotations. We convert the selected few-shot annotations to the same output format as GEMBA-Patent, excluding annotations with an empty error category or severity. We evaluate on the remaining 27 claims in each direction. We test two prompt versions. GEMBA-Patent-fewshot adds the two examples to the GEMBA-Patent prompt. GEMBA-Patent-fewshot-v2 additionally removes the instruction "If no error is detected, return {"annotations": []}." For both versions, we handle invalid outputs in the same way as GEMBA-Patent-Hybrid. Finally, we compare the results of GEMBA-Patent-fewshot-Hybrid and GEMBA-Patent-fewshot-v2-Hybrid with GEMBA-Patent-Hybrid, Encoder-GEMBA-Patent, and Encoder-GEMBA-MQM. For GEMBA-Patent-Hybrid, Encoder-GEMBA-Patent, and Encoder-GEMBA-MQM, we recompute the metrics after excluding the selected claims.

Results. Table 5 shows the span-level precision, recall and F1 score, as well as category and severity accuracy over matched spans, on the training data excluding the claims used as two-shot examples. Although GEMBA-Patent-fewshot-v2-Hybrid reduces the number of claims with empty annotations, both encoder-based methods achieve the highest recall and F1 in both translation directions. GEMBA-Patent-fewshot-Hybrid achieves the highest precision in the JA–EN direction, whereas GEMBA-Patent-fewshot-v2-Hybrid achieves the highest in the EN–JA direction. GEMBA-Patent-fewshot-Hybrid achieves the highest category and severity accuracies for the JA–EN direction, whereas Encoder-GEMBA-Patent achieves the highest values for the EN–JA direction.

D Evaluation Results on the Test Set

Tables 6–7 show the evaluation results on the test set as reported by the organizers.

E Correlations of Error Annotation Scores with Human Evaluation

Table 8 shows the Pearson and Spearman correlations between human scores and the automatic scores derived from Encoder-GEMBA-MQM, Encoder-GEMBA-Patent, and GEMBA-Patent-Hybrid on the training set.

```
You are an annotator for the quality of patent-claim machine translation. Your task is to identify errors and assess the quality of the translation.
```

Figure 1: System Prompt for GEMBA-Patent

```
Evaluate the following translation:

{source_lang} source:
```text
{source_seg}
```

{target_lang} translation:
```text
{target_seg}
```

Identify and classify translation errors using only the following hierarchy. Form each error label by joining its hierarchy levels with hyphens while preserving spaces and slashes within each category name.

```

Figure 2: User Prompt for GEMBA-Patent, Part 1: Input Format and Task Summary.

| | JA–EN | | | | |
|---|---|---|---|---|---|
| **Method** | **Precision** | **Recall** | **F1** | **Category Acc.** | **Severity Acc.** |
| Encoder-GEMBA-MQM (GPT-5.2) | 0.037 | **0.027** | **0.031** | 0.429 | 0.714 |
| Encoder-GEMBA-Patent (GPT-5.2) | 0.037 | **0.027** | **0.031** | 0.286 | 0.750 |
| GEMBA-Patent-Hybrid | 0.061 | 0.012 | 0.020 | 0.250 | 0.667 |
| GEMBA-Patent-fewshot-Hybrid | **0.067** | 0.015 | 0.024 | **0.467** | **0.933** |
| GEMBA-Patent-fewshot-v2-Hybrid | 0.064 | 0.018 | 0.029 | 0.368 | 0.737 |
| | **EN–JA** | | | | |
| **Method** | **Precision** | **Recall** | **F1** | **Category Acc.** | **Severity Acc.** |
| Encoder-GEMBA-MQM (GPT-5.2) | 0.059 | **0.043** | **0.050** | 0.368 | 0.684 |
| Encoder-GEMBA-Patent (GPT-5.2) | 0.059 | **0.043** | **0.050** | **0.500** | **0.711** |
| GEMBA-Patent-Hybrid | 0.107 | 0.018 | 0.031 | 0.313 | 0.563 |
| GEMBA-Patent-fewshot-Hybrid | 0.080 | 0.018 | 0.030 | 0.250 | 0.438 |
| GEMBA-Patent-fewshot-v2-Hybrid | **0.084** | 0.024 | 0.037 | 0.333 | 0.524 |

Table 5: Error annotation results on the training data, excluding the claims used as two-shot examples.

| **Direction** | **Annotation/claim** | **Detection F1** | **Span F1** | **Severity agreement** | **Pearson's $r$** | **Spearman's $rho$** |
|---|---|---|---|---|---|---|
| JA–EN | 3.1 | 0.111 | 0.097 | 0.36 | 0.65 | 0.73 |
| EN–JA | 4.2 | 0.139 | 0.155 | 0.70 | 0.38 | 0.32 |

Table 6: Evaluation results of our submitted system on the test set.

| **Direction** | $n$ | **Pearson's $r$** | **Spearman's $\rho$** | **MAE** | **Mean (Gold/Submitted)** |
|---|---|---|---|---|---|
| JA–EN | 10 | 0.654 (p=0.0401) | 0.727 (p=0.0172) | 43.3 | 34 / 76 |
| EN–JA | 12 | 0.383 (p=0.2193) | 0.320 (p=0.31) | 28.2 | 54 / 72 |

Table 7: Correlations, MAE, and mean scores for gold MQM scores and submitted scores on the test set.

| | JA–EN | | EN–JA | |
|---|---|---|---|---|
| **Method** | **Pearson's $r$** | **Spearman's $\rho$** | **Pearson's $r$** | **Spearman's $\rho$** |
| Encoder-GEMBA-MQM (GPT-5.2) | $0.2874^{**}$ | $0.3293^{**}$ | $0.2752^{**}$ | $0.3330^{**}$ |
| Encoder-GEMBA-Patent (GPT-5.2) | $0.2992^{**}$ | $0.3562^{**}$ | $0.2040^{*}$ | $0.2689^{**}$ |
| GEMBA-Patent-Hybrid (GPT-5.2) | $\mathbf{0.4927}^{**}$ | $\mathbf{0.4591}^{**}$ | $\mathbf{0.3458}^{**}$ | $\mathbf{0.3746}^{**}$ |

$^{*}p < 0.05$, $^{**}p < 0.01$.

Table 8: Correlations between scores derived from error annotation and human quality scores on the training data.

```
- accuracy (main category)
  - addition (sub category): the translation adds technical content or claim elements not present in the source
  - omission (sub category): the translation omits an essential claim element or other source content
  - untranslated text (sub category): source text has been left untranslated
  - mistranslation (sub category): the translation incorrectly represents the inventive idea, technical function or effect, element relationship, claim scope, claim type, or other source meaning
    - numerals/symbols (sub-sub category): a quantity, numerical range, condition, numeral, or symbol is translated incorrectly
    - article (sub-sub category): an article or antecedent basis, such as "a," "the," or "said," is used incorrectly
    - incorrect dependency (sub-sub category): an element relationship, constraint-clause attachment, coordination, or modification relationship is translated incorrectly
    - unknown dependency (sub-sub category): the dependency or attachment relationship expressed in the source is not maintained and cannot be determined from the translation
    - ambiguity (sub-sub category): the translation introduces ambiguity not present in the source, particularly ambiguity affecting element relationships or claim scope

- fluency (main category)
  - punctuation (sub category): punctuation is incorrect, including improper sentence division when the patent claim should be written as one sentence
  - spelling (sub category): spelling or capitalization is incorrect
  - grammar (sub category): the translation is grammatically incorrect in the target language
  - register (sub category): the grammatical register is inappropriate for a patent claim
  - inconsistency (sub category): the translation contains an internal inconsistency unrelated to terminology
  - character encoding (sub category): characters are garbled because of incorrect encoding

- terminology (main category)
  - inappropriate for context (sub category): a term is non-standard or inappropriate for the technical or patent context
  - inconsistent use (sub category): the same technical or claim term is translated inconsistently

- style (main category)
  - awkward (sub category): the translation is excessively awkward, difficult to read, or written in narrative or explanatory prose rather than patent-claim style

- other (main category): any other translation issue
- source error (main category): an error in the source text
- non-translation (main category): the output does not constitute a translation for which distinct errors can be reliably identified

```

Figure 3: User Prompt for GEMBA-Patent, Part 2: Error Categories.

```
Classify every error as:
- major: an error that changes or obscures meaning, claim scope, an essential claim element, a technical relationship, or essential claim form
- minor: a limited error that does not materially affect meaning or claim scope

Return JSON only in the following format:

{{
  "annotations": [
    {{
      "side": "input",
      "tagged_text": "preceding context <ERROR_START>exact text span<ERROR_END> following context",
      "text": "exact text span",
      "error": "accuracy-mistranslation-article",
      "severity": "major"
    }}
  ]
}}

For every annotation:
- "side" must be either "input" or "output".
- "tagged_text" must contain the complete source or translation claim with exactly one error span enclosed by "<ERROR_START>" and "<ERROR_END>".
- If "side" is "input", create "tagged_text" from the complete source_seg.
- If "side" is "output", create "tagged_text" from the complete target_seg.
- Do not alter any part of the claim except by inserting the error tags.
- "text" must exactly equal the text enclosed by "<ERROR_START>" and "<ERROR_END>".
- "severity" must be either "major" or "minor".
- Output one annotation object per error, with exactly one tagged error in each complete claim.

If no error is detected, return {{"annotations": []}}.
```

Figure 4: User Prompt for GEMBA-Patent, Part 3: Error Severities, Output Format, and Task Request.

```
(system) You are an annotator for the quality of machine translation. Your task is to identify errors and assess the quality of the translation.

(user) {source_lang} source:
```{source_seg}```
{target_lang} translation:
```{target_seg}```

The source segment and machine translation surrounded with triple backticks contain exactly one pair of <ERROR_START> and <ERROR_END> tags. The tags enclose a pre-identified error span. Classify only the error type and severity of this span. Do not identify any additional errors.

The categories of errors are: accuracy (addition, omission, untranslated text, mistranslation (numerals/symbols, article, incorrect dependency, unknown dependency, ambiguity)), fluency (punctuation, spelling, grammar, register, inconsistency, character encoding), terminology (inappropriate for context, inconsistent use), style (awkward), other, source error, or non-translation. Join the applicable category levels with hyphens in the "error" field.

Each error is classified as one of two categories: major and minor. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.


Output only the following JSON:
{{
  "annotations": [
    {{
      "text": "exact text enclosed by the tags, excluding the tags themselves",
      "error": "accuracy-mistranslation-article",
      "severity": "major"
    }}
  ]
}}
```

Figure 5: Prompt for Encoder-GEMBA-MQM

```
You are an annotator for the quality of patent-claim machine translation. Your task is to assign an error category and severity to each pre-identified error span.
```

Figure 6: System Prompt for Encoder-GEMBA-Patent

```
Evaluate the following translation:

{source_lang} source:
```{source_seg}```
{target_lang} translation:
```{target_seg}```

Text enclosed by <ERROR_START> and <ERROR_END> is a pre-identified error span. The tags are markers and are not part of the source or translation. Classify only the tagged error span. Do not identify additional errors or modify the span. A tagged source span does not necessarily indicate a source error.

Use only the following hierarchy. Form the error label by joining its hierarchy levels with hyphens while preserving spaces and slashes within each category name.

```

Figure 7: User Prompt for Encoder-GEMBA-Patent, Part 1: Input Format and Task Summary.

```
- accuracy (main category)
  - addition (sub category): the translation adds technical content or claim elements not present in the source
  - omission (sub category): the translation omits an essential claim element or other source content
  - untranslated text (sub category): source text has been left untranslated
  - mistranslation (sub category): the translation incorrectly represents the inventive idea, technical function or effect, element relationship, claim scope, claim type, or other source meaning
    - numerals/symbols (sub-sub category): a quantity, numerical range, condition, numeral, or symbol is translated incorrectly
    - article (sub-sub category): an article or antecedent basis, such as "a," "the," or "said," is used incorrectly
    - incorrect dependency (sub-sub category): an element relationship, constraint-clause attachment, coordination, or modification relationship is translated incorrectly
    - unknown dependency (sub-sub category): the dependency or attachment relationship expressed in the source is not maintained and cannot be determined from the translation
    - ambiguity (sub-sub category): the translation introduces ambiguity not present in the source, particularly ambiguity affecting element relationships or claim scope

- fluency (main category)
  - punctuation (sub category): punctuation is incorrect, including improper sentence division when the patent claim should be written as one sentence
  - spelling (sub category): spelling or capitalization is incorrect
  - grammar (sub category): the translation is grammatically incorrect in the target language
  - register (sub category): the grammatical register is inappropriate for a patent claim
  - inconsistency (sub category): the translation contains an internal inconsistency unrelated to terminology
  - character encoding (sub category): characters are garbled because of incorrect encoding

- terminology (main category)
  - inappropriate for context (sub category): a term is non-standard or inappropriate for the technical or patent context
  - inconsistent use (sub category): the same technical or claim term is translated inconsistently

- style (main category)
  - awkward (sub category): the translation is excessively awkward, difficult to read, or written in narrative or explanatory prose rather than patent-claim style

- other (main category): any other translation issue
- source error (main category): an error in the source text
- non-translation (main category): the output does not constitute a translation for which distinct errors can be reliably identified

```

Figure 8: User Prompt for Encoder-GEMBA-Patent, Part 2: Error Categories.

```
Classify every error as:
- major: an error that changes or obscures meaning, claim scope, an essential claim element, a technical relationship, or essential claim form
- minor: a limited error that does not materially affect meaning or claim scope

Return JSON only in the following format:

{{
  "annotations": [
    {{
      "text": "exact text enclosed by the tags, excluding the tags themselves",
      "error": "accuracy-mistranslation-article",
      "severity": "major"
    }}
  ]
}}

The "text" value must exactly match the text enclosed by <ERROR_START> and <ERROR_END>, excluding the tags themselves. The "error" value must be one of the defined error labels, and "severity" must be either "major" or "minor". Do not output any additional fields or text.
```

Figure 9: User Prompt for Encoder-GEMBA-Patent, Part 3: Error Severities and Output Format.

```
You are an expert evaluator of MACHINE TRANSLATION OF PATENT CLAIMS.

Your task is to evaluate the translation quality using a
rubric-based evaluation framework,
implemented explicitly as a CHECKLIST.

This is NOT general prose.
The input is a PATENT CLAIM, which has strict legal and structural constraints.

You must:
(1) Answer ALL checklist questions below using:
    - "yes" (no problem)
    - "partial" (minor or weakening problem)
    - "no" (clear or serious problem)
(2) For each question, provide a short justification.
(3) Provide a brief explanation of the evaluation result.
(4) Evaluate SOURCE vs HYPOTHESIS only; no reference is provided.

```

Figure 10: Prompt for Patent-Rubric-Checklist, Part 1: Task Setting and Evaluation Procedure.

```
--------------------
CHECKLIST (PATENT-CLAIM-SPECIFIC)
--------------------

[A] CONTENT FIDELITY (Meaning Preservation)

A1. Is the core inventive idea (what is being claimed) preserved?
A2. Are the technical functions or effects preserved without distortion?
A3. Is the translation free from hallucinated technical content not present in the
    source?

[B] ELEMENT COVERAGE (Completeness)

B1. Are all essential claim elements translated (no omissions)?
B2. Are relationships between elements (having, connected to, arranged on, etc.)
    preserved?
B3. Are quantities, ranges, or conditions (e.g. "at least one", numerical ranges)
    preserved?

[C] CLAIM SCOPE & LOGICAL CONSTRAINTS (CRITICAL)

C1. Is the claiming scope preserved?
    (e.g., "comprising" vs "consisting of", "characterized in that")
C2. Are constraint clauses correctly attached?
    (e.g., "wherein", "前記◯◯が〜")
C3. Are coordination vs modification relations preserved?
    (e.g., A and B that each have C, NOT A and (B that has C))

[D] CLAIM FORM & SYNTACTIC REQUIREMENTS

D1. Is the translation written as a SINGLE SENTENCE claim?
D2. Is the claim type preserved (apparatus / method / composition, etc.)?
D3. Is antecedent basis consistent?
    (a/the/said, 前記〜)
D4. Does the translation avoid narrative or explanatory prose style?

[E] FLUENCY & GRAMMATICALITY

E1. Is the translation grammatically correct in the target language?
E2. Is it stylistically appropriate for patent claims?
E3. Is the sentence readable and not excessively awkward or ambiguous?

```

Figure 11: Prompt for Patent-Rubric-Checklist, Part 2: Rubric and Checklist Items.

```
--------------------
INPUT FORMAT
--------------------
{source_lang} source patent claim:
"{source_seg}"

{target_lang} translated patent claim:
"{target_seg}"

--------------------
OUTPUT FORMAT (JSON ONLY)
--------------------

{{
  "checklist_answers": {{
    "A1": {{"answer": "yes", "justification": "Short justification."}},
    "A2": {{"answer": "yes", "justification": "Short justification."}},
    "A3": {{"answer": "yes", "justification": "Short justification."}},
    "B1": {{"answer": "yes", "justification": "Short justification."}},
    "B2": {{"answer": "yes", "justification": "Short justification."}},
    "B3": {{"answer": "yes", "justification": "Short justification."}},
    "C1": {{"answer": "yes", "justification": "Short justification."}},
    "C2": {{"answer": "yes", "justification": "Short justification."}},
    "C3": {{"answer": "yes", "justification": "Short justification."}},
    "D1": {{"answer": "yes", "justification": "Short justification."}},
    "D2": {{"answer": "yes", "justification": "Short justification."}},
    "D3": {{"answer": "yes", "justification": "Short justification."}},
    "D4": {{"answer": "yes", "justification": "Short justification."}},
    "E1": {{"answer": "yes", "justification": "Short justification."}},
    "E2": {{"answer": "yes", "justification": "Short justification."}},
    "E3": {{"answer": "yes", "justification": "Short justification."}}
  }},
  "overall_explanation": "Brief explanation of the evaluation result in 1-2 sentences."
}}
```

Figure 12: Prompt for Patent-Rubric-Checklist, Part 3: Input and Output Formats.

Ehime-U System for the WAT 2026 Patent Claims Evaluation Task: LLM-Based Translation Error Prediction Using Word Alignments

Wang Xu Yin **Isao Goto** **Takashi Ninomiya**
Ehime University
wang@ai.cs.ehime-u.ac.jp {goto.isao.fn,ninomiya.takashi.mk}@ehime-u.ac.jp

Abstract

This paper describes the Ehime-U system submitted to the Japanese–English Patent Claims Translation/Evaluation Task at WAT 2026. Our system predicts translation errors by providing an LLM with word-level correspondences between source and target claims. We compare six settings that differ in how these correspondences are obtained and whether they are used: four Unbalanced Optimal Transport (UOT)-based alignment methods (Forward, Reverse, AND, and OR), one alignment generated by LLM, and a baseline without alignment. We discuss how the choice of word correspondence affects error prediction. Against the gold reference annotations, under the hyperparameter settings used in this study, providing word correspondences improves error-detection F1 by up to 3.7 points over the no-alignment baseline (0.512 vs. 0.475). The improvement is category-specific: recall increases for alignment-dependent error categories but not for fluency errors. Comparing the two approaches to obtaining correspondences, we find that LLM-based alignment covers substantially more source tokens than the UOT alignments and achieves a higher F1 score than the best UOT setting under the hyperparameter settings used in this study (0.512 vs. 0.486).

1 Introduction

The Ehime University team (Ehime-U) participated in the Japanese–English Patent Claims Translation/Evaluation Task at the 13th Workshop on Asian Translation (WAT 2026) (Nakazawa et al., 2026). This paper describes the system that produced our submitted results. Patent claim translation requires both terminological precision and structural fidelity (Buma et al., 2025; Nakazawa et al., 2025). Errors such as omissions, additions, and mistranslations can change the legal scope of a claim (Li et al., 2019), making accurate error annotation critical (Lommel et al., 2013). Most existing systems prompt an LLM to compare the source and target directly (Kocmi and Federmann, 2023; Lu et al., 2024). In contrast, our system provides the LLM with explicit word-level correspondences as additional input when predicting translation errors.

Our motivation is that many patent-claim errors are alignment-level phenomena (Li et al., 2019; Arase et al., 2023). A source content word with no target counterpart is likely an omission candidate; a target word with no source counterpart is likely an addition candidate; a source word aligned to the wrong target word is likely a mistranslation candidate. Explicit correspondences make these patterns visible to the model.

Our contributions are as follows:

- We empirically evaluate the effect of using estimated word correspondences in LLM-based translation error prediction. Against the gold reference annotations, under the hyperparameter settings used in this study, providing explicit word correspondences improves error-detection F1 over the no-alignment baseline, and the LLM-based alignment, which covers substantially more source tokens than the UOT alignments, achieves the best overall F1 (0.512 vs 0.475 for the baseline). Moreover, the improvement is selective: recall increases for alignment-dependent error categories (accuracy and terminology consistency), while recall for fluency errors remains unchanged. These results show that, in Japanese–English patent claim error prediction, the proposed method using word correspondences is effective for some error categories such as omissions and mistranslations.
- We compare two ways of estimating word correspondences: unbalanced optimal transport (Cuturi, 2013; Arase et al., 2023) using scores produced by an LLM and LLM-based alignment estimation. We find that

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 134–144
November 9, 2026 ©2026 Association for Computational Linguistics

LLM-based alignment achieves the highest F1 (0.512), followed by the best UOT setting, UOT-AND (0.486). These results indicate that, under the hyperparameter settings used in this study, the LLM we use estimates word correspondences more appropriately than UOT.

2 System Description

Our system consists of three stages: tokenization, word alignment, and error prediction.

2.1 Tokenization

- Japanese: MeCab in `-Owakati` mode (Kudo et al., 2004).
- English: Moses tokenizer (Koehn et al., 2007).

These token IDs are the atomic units for downstream alignment and error annotation.

2.2 Word Correspondence

For each source–target sentence pair, we compute a set of word links. A word link is a correspondence from a source token to one or more target tokens. We use word links as additional input to the error-prediction stage: they indicate which source and target tokens correspond to each other.

We obtain word links with two different methods. The first estimates a score for each token pair and solves an unbalanced optimal transport (UOT) problem on that score. The second asks an LLM to predicted the alignment directly. We describe the two methods below.

Unbalanced Optimal Transport-based Alignment

We estimate word correspondence by first computing a score for each source–target token pair, and then solving a UOT problem on that score. We use two alternative scoring methods: an LLM-based semantic score (the main setting) and a character-based lexical score (a lightweight fallback).

The semantic score is assigned by an LLM. We present the tokenized source and target as local token IDs and ask the LLM to return a sparse list of plausible token pairs with a score $s \in [0, 1]$ (1.0 = exact translation, 0.7 = strong candidate, 0.4 = weak candidate; unlisted pairs are treated as no correspondence). The full prompt is listed in Appendix A.1. We convert the scores into a dense ground cost with $C[i][j] = 1 - s$ for listed pairs and $C[i][j] = 1$ for unlisted pairs. Because the score is assigned by the LLM based on meaning rather than character overlap, it provides a meaningful signal for Japanese–English alignment.

As a lightweight fallback, we also compute a character-based lexical score

$$\begin{aligned}\mathrm{sim}_{\mathrm{lex}}(s_i, t_j) = \max(&0.5 \cdot \mathrm{Jacc}(s_i, t_j) \\ &+0.5 \cdot \mathrm{Lev}(s_i, t_j),\ \mathrm{Syn}(s_i, t_j))\,,\end{aligned} \tag{1}$$

where $\mathrm{Jacc}(s_i, t_j)$ is the Jaccard similarity of the character n-gram sets of the two tokens ($n = 3$),

$$\mathrm{Jacc}(s_i, t_j) = \frac{|G(s_i) \cap G(t_j)|}{|G(s_i) \cup G(t_j)|}$$

with $G(x)$ the set of character n-grams of token x; $\mathrm{Lev}(s_i, t_j)$ is the normalized Levenshtein similarity

$$\mathrm{Lev}(s_i, t_j) = 1 - \frac{\mathrm{dist}(s_i, t_j)}{\max(|s_i|, |t_j|)}$$

with $\mathrm{dist}(\cdot, \cdot)$ the Levenshtein edit distance (Levenshtein, 1966); and $\mathrm{Syn}(s_i, t_j) = 0.9$ if the pair appears in a small Japanese–English function-word table and 0 otherwise. Because Japanese and English use almost disjoint character sets, this score is only meaningful for numerals, Latin loanwords, and function words.

The semantic score is the main setting. When the LLM is unavailable or returns an empty list of token pairs, we fall back to the character-based lexical score, which is computed entirely offline. Unless otherwise noted, all experiments in this paper use the semantic score.

Whichever score is used, we convert it into a non-negative ground cost

$$C[i][j] = 1 - \mathrm{score}(s_i, t_j) \in [0, 1], \tag{2}$$

where $\mathrm{score}(s_i, t_j)$ denotes either the semantic score s or the lexical score $\mathrm{sim}_{\mathrm{lex}}(s_i, t_j)$. Both scores are in $[0, 1]$, so the resulting cost is also in $[0, 1]$. A smaller cost means a closer correspondence.

Following Cuturi (2013) for the entropy-regularized formulation, Liero et al. (2018) and Chizat et al. (2018) for the marginal relaxation with KL penalties, and Arase et al. (2023) for the unbalanced transport formulation for word alignment,

we then solve the entropy-regularized unbalanced transport problem

$$\min_{P}\Big\{\langle P, C\rangle - \varepsilon H(P) + \tau\big(\mathrm{KL}(P\mathbf{1} \,\|\, \mathbf{a}) + \mathrm{KL}(P^{\top}\mathbf{1} \,\|\, \mathbf{b})\big)\Big\}, \quad (3)$$

where $P \in \mathbb{R}^{n\times m}_{\geq 0}$ is the transport plan (n and m are the numbers of source and target tokens, respectively), C is the cost matrix defined in Equation 2, $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ are the uniform marginal distribution vectors for the source and target tokens, respectively. Specifically, their components are defined as $a_i = 1/n$ for $i = 1, \ldots, n$ and $b_j = 1/m$ for $j = 1, \ldots, m$, ε is the entropy regularization strength, τ is the marginal relaxation strength, $H(P)$ is the entropy of the plan, and $\mathrm{KL}(x \,\|\, y)$ is the Kullback–Leibler divergence. $P\mathbf{1} \in \mathbb{R}^n$ and $P^{\top}\mathbf{1} \in \mathbb{R}^m$ are the row and column sums of P. Their values are given in Section 3.

Each source row $P[i, :]$ is row-normalized into a distribution over target tokens. A target token j is linked to source token i when the normalized mass is at least `link_threshold`; at most `max_links_per_src` target tokens are kept per source token. A source row whose total mass is below `row_mass_floor` is treated as having no correspondence. The values of these three parameters are given in Section 3.

From the same cost matrix we derive four alignments. Forward (source→target) and Reverse (target→source) are obtained by solving UOT on C and on $C^{\top}$, respectively. AND is the strict intersection of the Forward and Reverse link sets, and OR is their union.

LLM-based Alignment

We use an LLM to estimate word correspondence. In this setting, the LLM is given the tokenized source and target claims and is asked to produce span boundaries and a token-level alignment in a single request. The prompt used for this alignment is listed in Appendix A.2. The specific model used is described in Section 3.2.

2.3 Translation Error Prediction Using Word Alignment

In the error prediction stage, we use an LLM to predict translation errors. Each input to the LLM consists of the tokenized source claim, the tokenized target claim, the word correspondence list, and the token IDs of both sides. Each claim is processed independently: the model receives only the current claim, and no information from other claims of the same patent is provided. The LLM then predicts translation errors, each containing five parts: the side (`input` or `output`), the span (source side uses `source_token_ids`, target side uses `target_token_ids`), the MQM type (e.g., 誤訳, 訳抜け, 文法) (Lommel et al., 2013), and the severity (小 or 大).

The full prompt is listed in Appendix A.3. It contains the word correspondence list and asks the model to examine the source–target correspondence item by item, using each aligned pair as reference and referring to the definitions of the MQM error types(e.g., 誤訳, 訳抜け, 文法), following the definitions provided in the task Web page[1], and finally to predict the errors present in the source and target. The prompt also contains the complete token ID lists for both sides, allowing the model to indicate the span of each error by token ID. We then map these IDs back to character offsets: that is, we convert the token IDs produced by the model into a character range in the corresponding text, taking the minimum start and the maximum end across the tokens in the group as the character span of the error.

The prompt additionally includes one few-shot example drawn from the training data, to convey the expected output format. The example is a gold-annotated block containing a source claim, a target claim, and a list of MQM errors, rendered in the same format as the expected output:

```
SOURCE: <source claim>
TARGET: <target claim>
ERRORS: [{"side": ..., "text": ...,
"error": ..., "severity": ...}, ...]
```

The LLM returns a single JSON object with a quality score and a list of errors. Each error is specified by side (`input` or `output`), token IDs, verbatim span text, MQM error type, and severity (小 or 大). We parse and normalize the JSON, and map the token IDs back to character offsets. The resulting annotations are written in the same format as the training data.

2.4 Baseline Translation Error Prediction without Using Word Alignment

In the no-alignment baseline, no word correspondence is provided. An LLM receives only the

[1] https://sites.google.com/view/pat-claims-trans-2026/

source claim, the target claim, and the token ID lists, and must infer errors from the raw source–target comparison. Apart from the instructions regarding the use of word correspondence, the baseline follows the same procedure as described in Section 2.3, using the same few-shot example and output format. The complete prompt used for the no-alignment baseline is provided in Appendix A.3.1.

3 Experimental Setup

3.1 Data

We use test-set files containing Japanese patent claims and their English translations, provided by the Japanese–English Patent Claims Translation Task at WAT 2026. The test set contains 10 claims for JA–EN and 12 claims for EN–JA.

In the LLM-based error prediction stage, the few-shot example is drawn from a single gold-annotated file in the training data:

```
ja-en_sys1_02_WO2024241416.json
```

This file contains gold MQM annotations for one Japanese – English patent claim. We use the first annotated block from this file as the few-shot example.

3.2 Models and Parameters

This section summarizes which model is used for which purpose and lists the parameters of each model.

For word alignment, the semantic score between source and target tokens is computed by an LLM. We use use OpenAI GPT-6 Astra via its API for this scoring step, with reasoning effort `low` and temperature 0; the full prompt is listed in Appendix A.1. Alternatively, the LLM can produce span boundaries and a token-level alignment directly in a single request, as described in Appendix A.2.

For error prediction, we use OpenAI GPT-6 Astra via its API, with reasoning effort `low` to reduce latency, and temperature 0 for deterministic output.

For UOT, we set the entropy regularization strength to $\varepsilon = 0.05$, which controls how smooth the transport plan is.[2] We set the marginal relaxation strength to $\tau = 0.05$, which controls how strongly the source and target marginals are enforced.[3] The link threshold is set to 0.3, the minimum normalized transport mass required to emit a word link. The row mass floor is set to 0.0, effectively disabling this filter. The maximum number of links per source token is set to 3.

[2] Smaller values make the plan closer to a hard assignment; larger values make it smoother and easier to solve.

[3] Smaller values relax the marginals more, so a source token with no counterpart can drop its mass; larger values approach balanced optimal transport.

System Name	Submitted Result Folder
LLM	`GPT6`
BASELINE	`GPT6-noAlignment`
UOT-AND	`AND`
UOT-OR	`OR`
UOT-FORWARD	`Forward`
UOT-REVERSE	`Reverse`

Table 1: Mapping between submitted result folder names and system names used in this paper.

3.3 System Names

We compare six settings.

LLM Uses word correspondences estimated by an LLM and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-AND Estimates word correspondences using the strict intersection (AND) of the Forward and Reverse UOT alignments and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-OR Estimates word correspondences using the union (OR) of the Forward and Reverse UOT alignments and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-FORWARD Estimates word correspondences using the Forward (source→target) UOT alignment and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

UOT-REVERSE Estimates word correspondences using the Reverse (target→source) UOT alignment and predicts translation errors using an LLM based on the source – target text and the estimated correspondences.

BASELINE Uses no word correspondences and predicts translation errors using an LLM based only on the source – target text.

See Section 2.2, Section 2.2, and Section 2.3 for details. The correspondence between the submitted result folder names and the system names used in this paper is shown in Table 1.

Setting	Detection F1			Recall (gold category)			Ann/claim	Sev. agr.	Pearson r	Spearman ρ
	all	output	span	Acc.	Term.	Flu.				
LLM	**0.512**	**0.555**	**0.283**	0.40	0.34	0.51	11.2	0.47	0.65	0.63
UOT-AND	0.486	0.528	0.254	0.36	0.17	0.48	11.2	0.46	0.64	0.59
UOT-REVERSE	0.485	0.523	0.276	0.35	0.17	0.51	10.3	0.47	0.60	0.65
UOT-OR	0.482	0.526	0.261	0.35	0.34	0.52	10.6	0.42	0.61	0.59
BASELINE	0.475	0.510	0.255	0.34	0.17	0.51	11.1	0.45	0.63	0.62
UOT-FORWARD	0.454	0.485	0.256	0.28	0.31	0.52	11.0	0.39	0.63	0.56

Table 2: Official results for the Ehime-U submissions to the WAT 2026 Japanese–English Patent Claims Translation/Evaluation Task. Detection F1 measures error-detection performance against the gold reference. The *all* setting counts errors on both the input and output sides, whereas *output* counts only errors on the output (target) side. The *span* score is span-level detection F1 at an intersection over union (IoU) threshold of 0.5. Recall is calculated over gold-reference characters belonging to each MQM category: Acc. denotes accuracy, Term. denotes terminology consistency, and Flu. denotes fluency. Ann/claim is the average number of submitted annotations per claim. Sev. agr. is the severity agreement rate over characters marked as errors by both the gold reference and the submission. Pearson r and Spearman ρ measure claim-level correlations between the submitted quality scores and the gold MQM scores, respectively. The gold annotation density is 9.3 annotations per claim for JA–EN. Correlations should be interpreted with caution because they are calculated over only 10 claims.

Setting	Aligned src	Null src	Aligned tgt	Null tgt
LLM	3113	1232	2667	399
UOT-AND	1346	2999	1172	1894
UOT-OR	1508	2837	1192	1874
UOT-FORWARD	1508	2837	1190	1876
UOT-REVERSE	1174	1892	1346	1720
BASELINE	—	—	—	—

Table 3: Correspondence coverage per setting. "Aligned" counts tokens with at least one link, and "Null" counts tokens with no link. For UOT-REVERSE, the source columns count English tokens and the target columns count Japanese tokens, so its totals are not directly comparable to the other rows.

4 Experimental Results

4.1 Effect of Word Correspondence on Error Prediction

We compare the six settings using the official scores provided in the task organizers' result report (Nakazawa et al., 2026).

Table 2 summarizes the official results for all six settings, including character-level and span-level error-detection F1, category-specific recall, annotation density, severity agreement, and claim-level score correlations. We count a reference error as detected if a predicted error span overlaps with it; a predicted error that overlaps with no reference error is counted as a false positive.

For character-level detection F1 on both the input and output sides (the *all* metric), the four UOT variants achieve an average score of 0.477. The best UOT setting, UOT-AND, reaches 0.486, outperforming BASELINE (0.475). LLM achieves the highest score (0.512), suggesting that the LLM-based alignment is the most effective of the six settings under this metric. Among the UOT variants, UOT-AND (0.486) and UOT-REVERSE (0.485) perform best, whereas UOT-FORWARD (0.454) performs worst. These results suggest that the choice of alignment direction and combination strategy affects error-prediction performance.

The category-specific recall results reveal differences across error categories. Compared with BASELINE, LLM improves recall for accuracy from 0.34 to 0.40 and for terminology consistency from 0.17 to 0.34, while fluency recall remains unchanged at 0.51. This pattern suggests that word-correspondence information may be particularly helpful for identifying accuracy- and terminology-related errors.

4.2 Correspondence Statistics

Table 3 reports correspondence coverage on both the source and target sides. "Aligned src" counts source tokens with at least one link, and "Null src" counts source tokens with no link; "Aligned tgt" and "Null tgt" are defined analogously for target tokens.

Reverse is shown from the target side, so its two "source" columns count English tokens and its two "target" columns count Japanese tokens; its totals are not directly comparable to the other rows.

The statistics show that, under the hyperparam-

Error label	Main category	Subcategory	Sub-subcategory
訳抜け (omission)	Accuracy	Omission	–
未翻訳 (untranslated text)	Accuracy	Untranslated text	–
湧き出し (addition)	Accuracy	Addition	–
誤訳 (mistranslation)	Accuracy	Mistranslation	–
(誤訳) 数値・記号 (numerals/symbols)	Accuracy	Mistranslation	Numerals/Symbols
(誤訳) 冠詞 (article)	Accuracy	Mistranslation	Article
(誤訳) 係り先誤り (incorrect dependency)	Accuracy	Mistranslation	Incorrect dependency
(誤訳) 係り先が不明 (unknown dependency)	Accuracy	Mistranslation	Unknown dependency
文脈上不適切 (訳語選択)	Terminology	Inappropriate for context	–
句読点 (punctuation)	Fluency	Punctuation	–
文法 (grammar)	Fluency	Grammar	–
ぎこちない (awkward)	Style	Awkward	–
用語の一貫性 (inconsistent terminology)	Terminology	Inconsistent use	–

Table 4: Mapping of the error labels used in our experiments to the MQM taxonomy.

Setting	Top error types
LLM	訳抜け (41), 誤訳 (25), 冠詞 (19)
UOT-AND	訳抜け (40), 誤訳 (26), 冠詞 (19)
UOT-OR	訳抜け (39), 誤訳 (25), 冠詞 (16)
UOT-FORWARD	訳抜け (39), 誤訳 (24), 冠詞 (16)
UOT-REVERSE	訳抜け (39), 誤訳 (24), 冠詞 (18)
BASELINE	訳抜け (39), 誤訳 (30), 冠詞 (18)

Table 5: Top three error types per setting.

Setting	Total	Major	Minor
GOLD	65	40	25
LLM	112	37	75
UOT-AND	112	41	71
UOT-OR	106	36	70
UOT-FORWARD	110	37	73
UOT-REVERSE	103	39	64
BASELINE	111	34	77

Table 6: Error counts per setting. The GOLD row reports the number of errors annotated in the gold reference.

eter settings used in our experiments, LLM predicted substantially more correspondences than any UOT-based setting. On the source side, LLM links 3,113 tokens, whereas the UOT variants link 1,174 – 1,508; on the target side, LLM links 2,667 tokens, compared with 1,172 – 1,346 for UOT variants. Under these experimental conditions, the LLM-based alignment is considerably denser than the UOT alignments.

4.3 Error Type Distribution

Table 4 presents the mapping between all error labels used in our experiments and the corresponding categories in the MQM taxonomy. Table 5 lists the three most frequent error labels for each setting. The same three labels were the most frequent across all settings: 訳抜け (omission), 誤訳 (mistranslation), and 冠詞 (article error). 訳抜け was the most frequent overall, appearing 39–41 times per setting, indicating that omission was the dominant error type in this dataset regardless of the alignment setting.

Table 6 compares the number and severity of errors in the gold reference with those predicted by the six settings. The gold reference contains 65 annotated errors, including 40 major and 25 minor errors. In contrast, the six settings predicted between 103 and 112 errors in total, substantially more than the number of errors annotated in the gold reference.

Among the six settings, UOT-AND and LLM predicted the most errors (112 each), whereas UOT-REVERSE predicted the fewest (103). The number of predicted major errors ranged from 34 to 41, with UOT-AND predicting the most (41) and BASELINE the fewest (34). The number of predicted minor errors ranged from 64 to 77, with BASELINE predicting the most (77) and UOT-REVERSE the fewest (64).

Overall, under the hyperparameter settings used in this study, all six settings predicted more errors than were annotated in the gold reference. The predicted error counts and severity distributions also varied across settings, although the three most frequent error labels remained the same. The differences between settings are small overall (Table 5).

4.4 Qualitative Observations

We observe that omission (訳抜け) and mistranslation (誤訳) errors dominate in every setting, and that the three most frequent error types are the same across all six settings. This suggests that the choice of word correspondence mainly affects the number of errors the model reports, not the set of dominant error types. The differences between settings are small: UOT-AND predicted the most major errors,

while BASELINE predicted the fewest major but the most minor errors. The latter is consistent with the no-alignment baseline flagging more low-severity issues when it cannot rely on correspondence information.

5 Conclusion

We presented a three-stage error prediction system for patent claim translations that used word-level correspondence as auxiliary information. The first stage tokenized the source and target claims using language-specific tokenizers. The second stage predicted word correspondences using five approaches: four unbalanced optimal transport (UOT) alignment strategies (Forward, Reverse, AND, and OR) and one LLM-based alignment method. Together with a no-alignment baseline, these approaches yielded six experimental settings. The third stage prompted an LLM to predict MQM errors using the word correspondences, or without them in the baseline setting, along with one few-shot example and a filtered correspondence list.

Among the six settings, the LLM-based alignment achieved the highest error-detection F1. It also outperformed the no-alignment baseline (0.512 vs. 0.475; Table 2). Compared with the gold reference annotations, the improvement varied across error categories: recall for accuracy and terminology consistency increased, whereas recall for fluency errors remained unchanged (Table 2). The three most frequent error types were the same across all settings (訳抜け, 誤訳, and 冠詞; Table 5), while the total number of predicted errors varied only slightly (103–112; Table 6). These results suggest that word correspondence can improve error-detection performance, particularly for accuracy- and terminology-related errors, although the observed performance differences were relatively small and depended on the experimental settings.

Future work includes tuning the unbalanced optimal transport parameters (ε, τ, link threshold, and row mass floor) on development data, and extending the evaluation to more translation directions.

Acknowledgments

This work was supported by joint research funding from Mirai Translate, Inc.

References

Yuki Arase et al. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kosei Buma, Takehito Utsuro, and Masaaki Nagata. 2025. Two step automatic post editing of patent machine translation based on pre-trained encoder models and LLMs. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 218–231, Mumbai, India. Association for Computational Linguistics.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2018. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, pages 2292–2300.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation (WMT)*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303.

Matthias Liero, Jürgen Jost, and Giandomenico Mielke. 2018. Optimal transport in competition with reaction: The hellinger–kantorovich distance. *SIAM Journal on Mathematical Analysis*, 50(1):1170–1204.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2013. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. pages 455–463.

Qingyu Lu et al. 2024. Error analysis prompting enables human-like translation evaluation in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Taiki Ushimaru, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2026. Findings of the 2nd patent claims translation/evaluation task. In *Proceedings of the 13th Workshop on Asian Translation*.

Toshiaki Nakazawa et al. 2025. Findings of the first patent claims translation task at WAT2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*.

A Prompts

This appendix lists the prompts used in the pipeline. All prompts were issued to GPT-6-Astra through an OpenAI API with reasoning effort `low` and temperature 0. Placeholders in curly braces are replaced at run time.

A.1 LLM-based Token-Pair Scoring

The UOT score stage asks the model to score how strongly each source token corresponds to each target token, so that a downstream UOT solver can find the globally optimal word alignment.

```
[System]
You are an expert bilingual patent linguist.

SOURCE LANGUAGE: {source_language}
TARGET LANGUAGE: {target_language}

You are given the tokenized SOURCE and TARGET
    as LOCAL token IDs.
Your job is NOT to produce the final alignment.
      Your job is to score how
strongly each SOURCE token corresponds to each
     TARGET token, so that a
downstream Unbalanced Optimal Transport (UOT)
    solver can find the globally
optimal word alignment.

For every SOURCE token that has a plausible
    translation in the TARGET, list
the candidate TARGET tokens and a score in [0,
     1]:

- 1.0  = this is the exact translation of the
    source token
- 0.7  = a strong candidate (synonym,
    inflection, or the same term)
- 0.4  = a weak candidate (related meaning,
    part of the same phrase)
- omit = not a plausible counterpart

Rules:
- Score by MEANING, not surface form.
    Japanese and English rarely share
  characters, so a lexical match is usually
    meaningless.
- Repeated terms (前記 / "the", 出力電圧 / "
    output voltage") should be scored
  for EVERY occurrence, by syntactic position.
- Function words (の, を, に, が, は, と)
    usually map to an English
  preposition/determiner or to nothing; score
    them 0.4-0.7 where a target
  counterpart exists.
- List at most 5 target candidates per source
    token.
- You do NOT need to cover every source token;
     the solver handles gaps.

Return JSON only:
{"scores": [{"source": 0, "target": 3, "score":
     0.9},
            {"source": 1, "target": 4, "score":
     0.6}]}

[User]
SOURCE TOKENS:
{src_listing}

TARGET TOKENS:
{tgt_listing}

Score the plausible source->target token
    correspondences.
- List only pairs that are plausible
    counterparts.
- Use a score in [0, 1].
- You do NOT need to cover every source token.

Return JSON only.
```

A.2 Word Alignment

The alignment stage asks the model to produce span boundaries and a token-level alignment in one JSON object.

```
[System]
You are an expert bilingual patent linguist.

SOURCE LANGUAGE: {source_language}
TARGET LANGUAGE: {target_language}

You are given the tokenized SOURCE and TARGET
    as LOCAL token IDs.
Do TWO things in ONE answer:

(A) SPAN CONSTRUCTION -- mark, for EACH side
    independently, where each
new span begins. Output "source_breaks" / "
    target_breaks": the token
index where each new span begins. Index 0
    always starts the first span.

(B) TOKEN-LEVEL ALIGNMENT -- state which
    SOURCE TOKENS correspond to
which TARGET TOKENS. Use the DIRECT token IDs
    you were shown
([S0], [S1], ... / [T0], [T1], ...), NOT span
```

```
    ordinals.

CRITICAL RULES FOR ALIGNMENT:
1. Each alignment entry has EXACTLY ONE source
     token in "source_tokens".
2. A source token may map to ZERO, ONE, or TWO
     target tokens.
3. ALIGN LEXICAL MEANING, not phrase meaning.
4. For noun phrases, align nouns to nouns and
    modifiers to modifiers.
5. Repeated words must be resolved
    independently.
6. Match repeated terms by SYNTACTIC POSITION,
     not surface proximity.
7. A target token should normally be the
    counterpart of AT MOST ONE
source token.
8. Word order may differ.
9. Leave a source token unaligned ([]) ONLY
    when it is a Japanese
grammatical particle.

Return JSON only:
{"source_breaks": [0, 2, 3], "target_breaks":
    [0, 3],
 "alignments": [{"source_tokens": [0], "
    target_tokens": [1]}, ...]}

[User]
SOURCE TOKENS:
{src_listing}

TARGET TOKENS:
{tgt_listing}

Mark the span boundaries for each side, then
    align each SOURCE token to
its direct lexical translation equivalent in
    the TARGET.
- ONE source token per alignment entry.
- ZERO, ONE, or TWO target tokens. Never more
    than TWO.
- If there is no direct lexical counterpart,
    return [].
- Search the entire target window; do not rely
     on token position.

Return JSON only.
```

A.3 Error Prediction

The error prediction stage asks the model to report translation errors in the MQM schema. The prompt contains the source and target claims, the word correspondence list, the token ID lists, the MQM error type definitions, and one few-shot example.

```
[System]
You are an expert MQM patent translation
    annotator. Output exactly one
JSON object, nothing else.

[User]
Report all translation errors you can identify.

Perform the analysis in multiple passes:
1. Check every source content word and short
    phrase against its
translation.
2. Check every target content word and short
    phrase for unsupported
additions.
3. Check numbers, symbols, terminology,
    articles, grammar, dependencies,
and punctuation.
4. Check whether any source content is missing
     or left untranslated.
5. Check consistency across the whole
    translation.
6. After completing all checks, output every
    independently identifiable
error.

Do not stop when you find a few errors.
Do not assume that a generally understandable
    translation is error-free.

SOURCE ({source_language}):
{source_text}

TARGET ({target_language}):
{target_text}

TOKEN IDS (use these IDs for every error span):

{token_listing}

{alignment_section}

Output shape (exactly this JSON, no prose, no
    markdown):
{"score": 85, "errors": [{"side": "output",
 "target_token_ids": ["T0"], "text": "<
    verbatim span>",
 "error": "<label>", "severity": "小"}]}

The "score" must be an integer from 0 to 100.
    Do NOT output
"start"/"end"; Python computes offsets. For
    output errors return
target_token_ids; for input errors return
    source_token_ids. Use the
smallest consecutive token ID span covering
    the error.

湧き出し = Translation includes information
    that is not present in the source and that
     is not supposed to be included
訳抜け = Translation is missing content from
    the source and the omission is
    inappropriate.
未翻訳 = Source text has been left untranslated
誤訳 = Translaiton does not accurately
    represent the source.
(誤訳)数値・記号 = Translation errors related
    to numerals and symbols.
(誤訳)冠詞 = Incorrect use of articles
(誤訳)係り先誤り = The adjective phrase or
    parallel structure has an incorrect
    dependency (please point out the correct
    dependency)
(誤訳)係り先が不明 = The dependency structure
    of the source is not maintained.
(誤訳)解釈の曖昧性 = 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).
句読点 = 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.
スペリング = Incorrect spelling or
    capitalization.
文法 = Problems with grammar, other than
    orthography.
言語使用域 = Wrong grammatical register (eg,
    inappropriately informal pronouns).
一貫性の欠如 = Internal inconsistency (not
    related to terminology)
文字エンコーディング = Characters are garbled
    due to incorrect encoding.
文脈上不適切 (訳語選択) = Terminology is non-
    standard or does not fit context.
用語の一貫性 = Terminology is used
    inconsistently.
ぎこちない = Translation has stylistic problems
    .
その他 = Any other issue.
原文エラー = An error in the source.
非翻訳 = Impossible to reliably characterize
    distinct errors.

SPAN RULE (very important): for each error,
    quote the MINIMAL sub-span
that IS the error -- ONE word or ONE short
    phrase. Do NOT quote the
surrounding clause.

Identify all errors you can find.
Example (format only):
{fewshot}

Return exactly one JSON object.
```

A.3.1 No-Alignment Baseline

The error prediction stage asks the model to report translation errors in the MQM schema. The prompt contains the source and target claims, the token ID lists, the MQM error type definitions, and one few-shot example.

```
\begin{lstlisting}
[System]
You are an expert MQM patent translation
    annotator. Output exactly one
JSON object, nothing else.

[User]
Report all translation errors you can identify.

Perform the analysis in multiple passes:
1. Check every source content word and short
    phrase against its
   translation.
2. Check every target content word and short
    phrase for unsupported
   additions.
3. Check numbers, symbols, terminology,
    articles, grammar, dependencies,
   and punctuation.
4. Check whether any source content is missing
     or left untranslated.
5. Check consistency across the whole
    translation.
6. After completing all checks, output every
    independently identifiable
   error.

Do not stop when you find a few errors.
Do not assume that a generally understandable
    translation is error-free.

SOURCE ({source_language}):
{source_text}

TARGET ({target_language}):
{target_text}

TOKEN IDS (use these IDs for every error span):

{token_listing}

Output shape (exactly this JSON, no prose, no
    markdown):
{"score": 85, "errors": [{"side": "output",
 "target_token_ids": ["T0"], "text": "<
    verbatim span>",
 "error": "<label>", "severity": "小"}]}

The "score" must be an integer from 0 to 100.
    Do NOT output
"start"/"end"; Python computes offsets. For
    output errors return
target_token_ids; for input errors return
    source_token_ids. Use the
smallest consecutive token ID span covering
    the error.

湧き出し = Translation includes information
    that is not present in the source and that
     is not supposed to be included
訳抜け = Translation is missing content from
    the source and the omission is
    inappropriate.
未翻訳 = Source text has been left untranslated
誤訳 = Translaiton does not accurately
    represent the source.
(誤訳)数値・記号 = Translation errors related
    to numerals and symbols.
(誤訳)冠詞 = Incorrect use of articles
(誤訳)係り先誤り = The adjective phrase or
    parallel structure has an incorrect
    dependency (please point out the correct
    dependency)
(誤訳)係り先が不明 = The dependency structure
    of the source is not maintained.
(誤訳)解釈の曖昧性 = 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).
句読点 = 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.
スペリング = Incorrect spelling or
    capitalization.
文法 = Problems with grammar, other than
    orthography.
言語使用域 = Wrong grammatical register (eg,
    inappropriately informal pronouns).
一貫性の欠如 = Internal inconsistency (not
    related to terminology)
文字エンコーディング = Characters are garbled
    due to incorrect encoding.
文脈上不適切 (訳語選択) = Terminology is non-
    standard or does not fit context.
用語の一貫性 = Terminology is used
    inconsistently.
ぎこちない = Translation has stylistic problems
    .
その他 = Any other issue.
原文エラー = An error in the source.
非翻訳 = Impossible to reliably characterize
    distinct errors.

SPAN RULE (very important): for each error,
    quote the MINIMAL sub-span
that IS the error -- ONE word or ONE short
    phrase. Do NOT quote the
surrounding clause.

Identify all errors you can find.
Example (format only):
{fewshot}

Return exactly one JSON object.
```

A.4 Few-Shot Example Format

The few-shot example is a gold-annotated block rendered in the same format as the expected output:

```
SOURCE: {truncated source claim}
TARGET: {truncated target claim}
ERRORS: [{"side": ..., "text": ..., "error":
    ..., "severity": ...}, ...]
```

GABU@WAT 2026: Zero Shot and QLoRA Adaptation of Gemma 4 E4B for Multimodal English to Indic Translation

Shiv Ram Saud
Kathmandu University
saudshivram373@gmail.com

Aashish Mahato
Kathmandu University
aashishmtho24@gmail.com

Prajwal Ghimire
Kathmandu University
prajwalghimire22@gmail.com

Sundeep Dawadi
Kathmandu University
sundeepdwd@gmail.com

Ramesh Saud
rajusaud245@gmail.com

Sunil Regmi
Kathmandu University
sunil.regmi@ku.edu.np

Abstract

We describe *GABU*, our WAT2026 English-to-Indic multimodal translation system (WAT2026 Organizers, 2026), which translates image grounded English captions into Hindi, Bengali, Malayalam, and Odia. The task follows the WAT English-to-Indic multimodal track setup with text only, image captioning, and multimodal tracks (Parida and Bojar, 2025; Parida et al., 2024). Our system builds on the pretrained, instruction-tuned vision-language model `google/gemma-4-E4B-it`. Each input combines a cropped visual region with the English caption, rendered through a chat template with a script constrained translation instruction. We explore two approaches which includes a zero shot baseline and a fine tuned approach of the model using QLORA (rank 64, alpha 128, 4-bit NF4, double quantization), trained for 3,000 steps due to computational resource constraints, reflecting our aim to adapt the model's existing instruction-following ability rather than train it from scratch. We evaluate both systems using BLEU and CHRF3 under greedy decoding and report macro averaged scores across the target languages. The fine-tuned model substantially outperforms the zero shot baseline and achieves a BLEU score of 43.97 and 69.99 CHRF3 on the test split, and 40.37 BLEU and 68.25 CHRF3 on the challenge split. These results correspond to improvements of 24.39 BLEU and 18.72 CHRF3 points on the test split, and 16.70 BLEU and 11.90 CHRF3 points on the challenge split.

1 Introduction

The English-to-Indic multimodal machine translation (MMT) aims to improve translation of image-grounded English captions into low resource Indic languages by using paired visual context alongside source text (Parida and Bojar, 2025; Parida et al., 2024). The underlying task builds on the Visual Genome-derived Hindi and Bengali datasets (Parida et al., 2019; Sen et al., 2022) and the broader Multi30K paradigm of image-grounded translation (Elliott et al., 2016). Visual context is most valuable where the source caption is ambiguous: classic examples include polysemous words such as "court" or "dish", whose correct Indic rendering can depend on the depicted scene. Whether models genuinely exploit images rather than improving through regularization or strong textual priors remains actively debated at the same time (Wu et al., 2021; Caglayan et al., 2019; Li et al., 2021). Our work targets the WAT2026 English-to-Indic multimodal task. The system described here is implemented and evaluated for four languages: Hindi (`hi`), Bengali (`bn`), Malayalam (`ml`), and Odia (`or`). We investigate how far a modern pretrained vision-language model can be taken in this setting rather than engineering a bespoke fusion architecture: first as a zero shot image grounded translator, then adapted with parameter efficient fine-tuning under tight compute constraints. Our contributions are:

- A parameter efficient fine-tuning approach using QLoRA joint training and over the four target languages.
- A comparison between zero shot baseline and finetuning approach.
- An evaluation on BLEU and CHRF3 with greedy based decoding.

The adapted system substantially outperforms the zero shot baseline on every language

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 145–154
November 9, 2026 ©2026 Association for Computational Linguistics

and split (Section 6.3), with the largest gains observed where the base model could barely produce the target script.

2 Related Work

Multimodal translation architectures. Early neural MMT introduced a doubly attentive decoder that attends to source words and spatial image features independently (Calixto et al., 2017). Subsequent Transformer based MMT proposed multimodal self-attention that grounds image representations in the source text to avoid encoding irrelevant visual information (Yao and Wan, 2020), building on the Transformer backbone (Vaswani et al., 2017). The recent WAT/WMT Indic systems explore gated fusion of textual and visual encoders (Hatami et al., 2024), object tags verbalized as text for a pretrained multilingual backbone (Gupta et al., 2021), ViT visual tokens projected into a multilingual LLM with autoregressive decoding (Khan et al., 2024), and InternVL2-derived image context with knowledge distillation (Rajpoot et al., 2024).

English-to-Indic and low resource MMT. The WAT2025 findings define three tracks (text only, captioning, multimodal) across Hindi, Bengali, Malayalam, and Odia (Parida and Bojar, 2025); the WMT2024 low resource MMT task covers a related English-to-low resource setting including Indic languages (Parida et al., 2024). The strong text only multilingual backbones are central: IndicTrans2 covers all 22 scheduled Indian languages (Gala et al., 2023), mBART provides multilingual denoising pretraining transferable even to unseen pairs (Liu et al., 2020), and web scale NLLB-style models demonstrate scaling to 200 languages (NLLB Team, 2024). Task specific WAT2025 systems build directly on IndicTrans2 with parameter efficient adaptation and data correction (text only track) (Betala et al., 2025), or probe selective CLIP based visual grounding on top of it (Kumar et al., 2025).

Datasets. Hindi Visual Genome provides English–Hindi image-caption pairs created by post editing automatic translations with the image in view (Parida et al., 2019). Bengali Visual Genome extends the same methodology to English–Bengali for both translation and captioning (Sen et al., 2022). These follow the Multi30K precedent of pairing images with multilingual descriptions (Elliott et al., 2016).

When does vision help? Whether visual context genuinely improves MMT is actively debated. Controlled studies show models can exploit images under degraded textual context (Caglayan et al., 2019), while other analyses find gains may stem from regularization rather than visual grounding (Wu et al., 2021), and that Multi30K-style references translated without seeing images weaken visual reliance (Li et al., 2021). Stronger vision encoders help when probed carefully (Li et al., 2022), with contrastive vision-language models such as CLIP (Radford et al., 2021) widely used for selective grounding (Kumar et al., 2025). For language pairs lacking multimodal parallel data, visual prediction networks that supply pseudo-visual features enable zero shot cross-modal transfer (Hirasawa et al., 2023), which is relevant for very low resource targets.

Parameter efficient adaptation and evaluation. Low rank adaptation freezes pretrained weights and learns rank decomposed residual updates that can be merged at deployment without inference latency (Hu et al., 2022); its quantized variant combines a frozen 4-bit NormalFloat base with double quantization and trainable adapters (Dettmers et al., 2023). For evaluation, BLEU computed follows the standardized detokenized reporting practice (Post, 2018), and chrF is the character n-gram F-score whose beta parameter balances precision and recall (Popović, 2017). Concretely, BLEU combines a brevity penalty BP with the geometric mean of n-gram precisions p_n (Papineni et al., 2002),

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{4} \tfrac{1}{4} \log p_n\right),$$

computed here with 13a tokenization, case sensitively, against a single reference. chrF is the F_β score over character n-grams (here $\beta = 3$, character order 6 with word-bigram statistics),

$$\mathrm{chrF}_\beta = \frac{(1+\beta^2)\, P\, R}{\beta^2 P + R},$$

where P and R are the averaged n-gram precision and recall.

3 Dataset and Preprocessing

3.1 WAT Task Setup

The WAT English-to-Indic multimodal track setup comprises text only, image captioning, and multimodal tracks (Parida and Bojar, 2025; Parida et al., 2024). The WAT2025 instantiation covered English into Hindi, Bengali, Malayalam, and Odia (Parida and Bojar, 2025). The related WMT2024 low resource MMT task covers a comparable English-to-low resource setting including Indic languages (Parida et al., 2024). A distinguishing property of this task family is that captions are attached to rectangular image regions rather than whole images, so localized visual context is the relevant signal (Parida and Bojar, 2025). Our submission is a multimodal, multilingual system that uses the image, the annotated region, and the English caption jointly in one model covering all four languages.

3.2 Languages and Data

We work with the Visual Genome derived corpora for our four implemented languages. The hindi Visual Genome provides English–Hindi image-caption pairs created by post editing automatic translations with the image in view, together with a challenge subset concentrating ambiguous words (Parida et al., 2019). The bengali Visual Genome extends the same methodology to English–Bengali for both translation and captioning (Sen et al., 2022). Malayalam and Odia are covered by the Malayalam Visual Genome 1.0 and Odia Visual Genome releases (Parida and Bojar, 2021; Parida et al., 2025).

These follow the Multi30K precedent of pairing images with multilingual descriptions (Elliott et al., 2016). Our training uses the train and development splits only whereas test and challenge material is held out for evaluation (Table 1). For Odia, no challenge reference text was released (images only). Therefore, challenge evaluation covers Hindi, Bengali, and Malayalam.

Language	Train	Dev	Test	Challenge
Hindi	28,930	998	1,595	1,400
Bengali	28,930	998	1,595	1,400
Malayalam	28,930	998	1,595	1,400
Odia	28,930	998	1,595	–

Table 1: Dataset statistics.

3.3 Image Processing

Each annotated bounding box is cropped with 15% contextual padding (clamped to image bounds; degenerate boxes fall back to the full image) and converted to an RGB thumbnail capped at 448×448 preserving aspect ratio. Only this crop is fed to the model, with text preprocessing limited strictly to whitespace stripping.

3.4 Region Coordinate Representation

The padded box is additionally expressed on a 0–1000 scale and inserted into the prompt as `Region coordinates [x1, y1, x2, y2] on a 0-1000 scale`, so the model receives both the pixels of the region and its normalized location. We use annotated region cropping and coordinates only. There is no object detector and no learned grounding module beyond the vision-language model itself.

3.5 Prompt Construction

We render each instance as a single turn chat message: the system instruction, the region image, and the user text. The system instruction states that the model is given a cropped image region with its location and a short English caption describing that region, and must output exactly one line. The caption in the target language and script only, with no explanation, quotes, English, or romanization. Target scripts are named explicitly (Devanagari, Bengali, Malayalam, Odia). The user text gives the 0–1000 coordinates, the English caption, and requests the target language caption. The reference translation during finetuning is appended as the assistant turn; at inference the template ends with the generation prompt. Formally, with target language L_t, cropped region image C, normalized box B, and English caption x, the message sequence is

$$M = \big[\, s(L_t),\ C,\ u(B, x, L_t) \,\big],$$

where s is the system instruction, u the user text, and the rendered prompt is $P = \mathrm{ChatTemplate}(M)$ with the generation prompt appended; training appends the reference y as the assistant turn. The identical template is used in both approaches, so all measured differences are weight effects rather than prompt

effects. Table 2 summarizes the prompting and generation settings.

Setting	Value
System instruction	expert-translator framing + one-line target-script-only contract
User text	0–1000 box + English caption + target-script cue
Image turn	cropped 448-px region
Few-shot examples	none
Chat template	processor `apply_chat_template`, generation prompt appended
Training prompt	same template + reference as assistant turn
Decoding	greedy, max. 32 new tokens
Postprocessing	special tokens skipped, whitespace collapsed single line
Seed	3407 (training only)

Table 2: Prompting and generation settings.

4 Methodology

4.1 System Architecture

Figure 1 shows the implemented pipeline. Shared input processing turns each Visual Genome sample (image, bounding box, English caption) into a padded region crop plus a chat template prompt carrying the 0–1000 region coordinates. The both approaches feed this identical input to Gemma 4 E4B-it (4-bit NF4), which generates the Indic caption.

4.2 Zero Shot Baseline

The baseline loads `google/gemma-4-E4B-it` with its auto processor under 4-bit NF4 quantization with double quantization and FP16 computation, and scores it without any adapter. Evaluation covers the test split in all four languages and the challenge split where reference text is available (Hindi, Bengali, Malayalam); the Odia challenge text was not released, so Odia challenge scoring is skipped. Two isolated worker processes (one per GPU) generate per-language outputs with resumable, Hub synced prediction files.

4.3 Parameter Efficient Fine-Tuning

The fine-tuning adapts the same base model with LoRA (Hu et al., 2022) under 4-bit quantization (Dettmers et al., 2023). Training rows from all four languages are flattened into a single dataset with a single adapter (joint multilingual training); only train and development splits are touched. The collator masks the rendered prompt and supervises only the reference translation. The base weights stay frozen via the PEFT adapter mechanism. The adapter contributes 139,526,144 trainable parameters, i.e. 1.73% of the 8,080,626,976 total parameters, using `peft` 0.19.1 with QLoRA because `google/gemma-4-E4B-it` is already instruction tuned. Our objective was to adapt the model's existing multimodal instruction-following ability to English-to-Indic translation rather than perform extensive task-specific pretraining. The adapter was trained for 3,000 steps Section 5.

4.4 QLoRA Configuration

The base model is loaded with 4-bit NormalFloat quantization, double quantization, FP16 computation dtype, and scaled dot-product attention.

LoRA uses rank 64, alpha 128, dropout 0.05, no bias, causal LM task type, and targets the query, key, value, and output projections as well as the gate, up, and down projections of the feed-forward blocks. Each adapted weight matrix takes the form

$$W' = W_0 + \frac{\alpha}{r} BA,$$

where W_0 is the frozen base weight, A and B are the trainable low-rank factors ($r = 64$, $\alpha = 128$), A is randomly initialized, B starts at zero so that $W' = W_0$ at the beginning of training, and dropout is applied to the LoRA input (Hu et al., 2022).

4.5 Final Model for Evaluation

Training was run for 3,000 steps (Section 5), and the resulting `best_adapter` checkpoint is merged into the base model (`merge_and_unload`) and re-quantized to 4-bit NF4 at scoring time with the identical worker configuration, so the final numbers are produced by the same code path as the baseline. The evaluation script pulls the adapter from the Hub head without a pinned revision. The stored `best.json` identifies step 3000 (macro CHRF3 71.59) as the best and final checkpoint, so the evaluated adapter corresponds to that recorded selection artifact rather than to a pinned Hub revision.

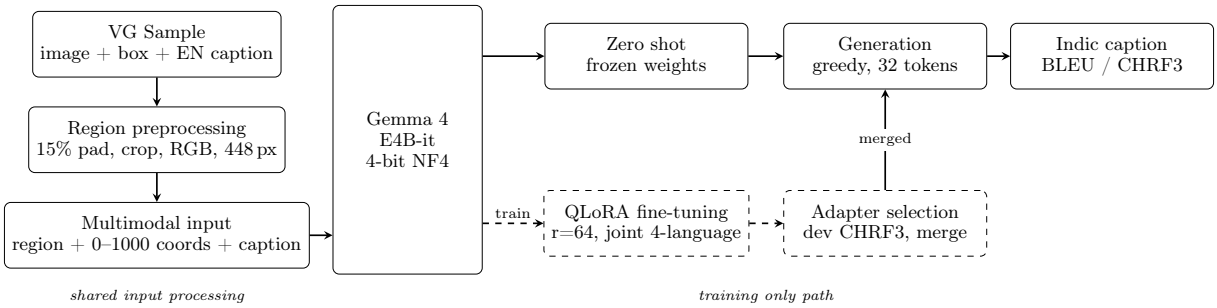


Figure 1: System architecture.

5 Training and Inference

Table 3 lists the training configuration. Optimization uses 8-bit AdamW with learning rate 10^{-4}, a cosine schedule, and a 0.05 warmup ratio. Each worker uses per device batch size 1 with gradient accumulation 8, giving an effective batch size of 16 over two-way data-parallel training; gradient checkpointing is enabled and the random seed is 3407. Maximum sequence length is 1024, with early stopping (patience 3, threshold 0.001); checkpoints are written every 500 steps (three retained) and pushed, rank-0 only, to a dedicated Hub repository together with training progress. Training is resumable from the latest local or Hub checkpoint. Training ran for 3,000 steps; the step-3,000 checkpoint is the one evaluated.

The training objective is causal language model cross-entropy restricted to the reference answer tokens. With rendered prompt P and reference token sequence $y = (y_1, \ldots, y_T)$, let $\mathcal{A}$ be the positions of the answer tokens (prompt prefix and padding positions are masked to -100 by the collator). The loss over the trainable LoRA parameters θ is

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{A}|} \sum_{t \in \mathcal{A}} \log P_\theta(y_t \mid y_{<t}, P),$$

with no label smoothing or token reweighting. The learning rate follows linear warmup for the first $s_w = 5\%$ of steps and cosine decay afterwards:

$$\mathrm{lr}(s) = \begin{cases} \mathrm{lr}_{\max} \cdot s / s_w, & s < s_w, \\ \frac{\mathrm{lr}_{\max}}{2} \left(1 + \cos \pi \frac{s - s_w}{S - s_w}\right), & s \geq s_w, \end{cases}$$

with $\mathrm{lr}_{\max} = 10^{-4}$ and $S = 3{,}000$ training steps.

Parameter	Value
Base model	`google/gemma-4-E4B-it`
Quantization	4-bit NF4, double quantization
LoRA rank / α	64 / 128
LoRA Dropout	0.05
Target modules	q, k, v, o, gate, up, down
Optimizer	AdamW (8-bit)
Learning rate	10^{-4}
Scheduler	cosine
Warmup	0.05
Batch size	$2 \times 1 \times 8 = 16$ effective
Hardware	2× T4
Gradient checkpoint	enabled
Max. sequence length	1024
Training steps	3,000 (compute-resource constraint)
Early stopping	patience 3, threshold 0.001
Checkpoint selection	best eval. loss + best macro CHRF3 adapter (step 3,000)
Generation	greedy, max. 32 new tokens

Table 3: Fine-tuning configuration as implemented.

Development monitoring combines forward loss scoring on a 400 row subset (100 per language) with generation scoring on a 200 row subset (50 per language) at every save. Checkpoint selection is dual: the trainer keeps the best model by development loss, while a separate `best_adapter` directory (with `best.json` and metric history) is refreshed whenever macro CHRF3 improves. Inference in both approaches is greedy (`do_sample=False`, at most 32 new tokens),

$$\hat{y}_t = \arg\max_w P_\theta(w \mid \hat{y}_{<t}, P),$$

with special tokens skipped and outputs whitespace normalized.

Setting	Value
Splits scored	test n=1,595 × 4; challenge n=1,400 × 3
References	dataset target captions, single reference per row
Predictions	per row JSONL (`idx`, `ref`, `decoded`) + plain text files
Metric code	`corpus_bleu` defaults + `corpus_chrf` (β=3)
Rounding / aggregation	2 dp; unweighted macro mean over evaluated languages
Dev role (fine-tune)	loss subset 400 + generation subset 200 for selection only

Table 4: Evaluation configuration.

6 Evaluation

6.1 Metrics

We report BLEU computed following the standardized detokenized reporting practice (Post, 2018), and chrF with beta 3 (CHRF3), the character n-gram F-score variant (Popović, 2017). Scores are computed per language and aggregated as unweighted macro means across the evaluated languages.

6.2 Experimental Setup

The zero-shot baseline is evaluated on the held-out test split (all four languages) and the challenge split (Hindi, Bengali, Malayalam). The fine-tuned adapter was monitored on development generation subsets during training and then evaluated on the same held-out splits. Decoding is greedy, so reruns are comparable without sampling variance. Final fine-tuned numbers come from scoring the merged best adapter on the full test (1,595 rows per language) and challenge (1,400 rows per language) splits, using the same prompt code, generation settings, row order, and metric code as the baseline. Table 4 summarizes the evaluation configuration, and Table 5 compares the two approach.

6.3 Results

On the development trajectory Table 6 where macro CHRF3 peaks at step 3,000 (71.59), which is the last checkpoint reached and therefore also the submitted adapter. Final held-out results are in Table 7.

7 Error Analysis

The both approach uses the same input construction, prompt template, decoding procedure, and evaluation code, the comparison isolates the effect of model adaptation under this experimental setup. However, without text only, image mismatch, or region/full image ablations, the contribution of visual information itself cannot be isolated. The test gains are uneven across languages (Table 7): Odia shows the largest absolute BLEU gain (+44.65, from 5.84 to 50.49) starting from the weakest zero shot baseline (CHRF3 37.25 to 72.82), while Hindi shows the smallest test BLEU gain (+10.10) starting from the strongest zero shot BLEU (31.99). Bengali (+18.93) and Malayalam (+23.85) fall in between. On CHRF3 the same ordering holds (Odia +35.57, Malayalam +16.26, Bengali +15.22, Hindi +7.85). These are observed patterns, not proof of why the model behaves this way. On challenge, Hindi reaches the best metric in our evaluation (50.58 BLEU / 69.68 CHRF3), while Bengali and Malayalam gains (+20.47/+14.10), (+15.55/+13.03) respectively.

8 Limitations

Our system adaptation uses a single rank-64 adapter, which bounds expressivity relative to full fine-tuning. Training was limited to 3,000 steps on two T4 GPUs (effective batch size 16, sequence length 1024) due to compute constraints; we do not claim this is an optimal budget or that longer training would not help. There is no dedicated fusion architecture, so architectural ablations are limited. The evaluated adapter is the recorded step-3,000 checkpoint, not a pinned Hub revision. Optimizer hyperparameters beyond learning rate, scheduler, and warmup use library defaults. Checkpoint selection reuses one 200-row development subset, and the same image pool underlies all splits by dataset design. Ablations, multiple seeds, and human evaluation remain future work.

9 Conclusion

We presented a Gemma 4 E4B-it based English-to-Indic multimodal translation system with region aware visual input, a zero shot baseline, and a quantized LoRA fine-tuning under joint multilingual training, evaluated with BLEU and CHRF3 under greedy based decoding because the base model is already instruction tuned. We adapted it for 3,000 steps, a prac-

Property	Zero shot baseline	Fine-tuned system
Model	Gemma 4 E4B-it, NF4, no adapter	same base + LoRA r64/α128 (139.5M, 1.73%)
Training data	none (eval only)	train 115,715 + dev 3,992, joint 4-language
Training	N/A	3,000 steps, eff. batch 16, 10^{-4} cosine
Prompt	Sec. 5 template, greedy 32 tokens	byte-identical template and decoding
Test macro	19.58 BLEU / 51.27 CHRF3	43.97 BLEU / 69.99 CHRF3
Challenge macro	23.67 BLEU / 56.35 CHRF3	40.37 BLEU / 68.25 CHRF3

Table 5: Baseline versus fine-tuned approach.

Step	Macro BLEU	Macro CHRF3	hi	bn	ml	or
500	43.70	67.20	36.01/59.23	36.81/66.88	49.04/73.38	52.95/69.29
1000	45.82	69.31	34.14/59.31	43.84/70.65	48.04/74.41	57.25/72.87
1500	41.70	65.43	33.60/58.14	45.02/68.44	32.73/63.70	55.47/71.44
2000	45.74	69.09	34.49/58.93	43.33/72.99	50.99/70.60	54.16/73.86
2500	41.67	67.03	39.29/61.54	42.39/74.40	32.09/60.29	52.90/71.88
3000	**49.76**	**71.59**	37.78/59.79	49.04/75.36	48.75/72.73	63.47/78.49

Table 6: Development trajectory up to 3,000 steps.

tical training budget under the computational resources available to us. The adapted system improves over the zero shot baseline on every language and split, with test macro gains of +24.39 BLEU / +18.72 CHRF3 and challenge macro gains of +16.70 / +11.90, confirming that parameter efficient adaptation of a frozen vision language backbone is effective for this task within this constrained training budget. Whether further training would yield additional gains remains an open question.

Acknowledgments

We thank the creators and annotators of the Visual Genome derived Hindi, Bengali, Malayalam, and Odia datasets (Parida et al., 2019; Sen et al., 2022; Parida and Bojar, 2021; Parida et al., 2025), and the WAT2026 organizers (WAT2026 Organizers, 2026) multimodal translation shared tasks for making the task data and evaluation protocols available.

References

Siddharth Betala, Kushan Raj, Vipul Betala, and Rohan Saswade. 2025. A picture is worth a thousand (correct) captions: A vision-guided judge-corrector system for multimodal machine translation. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 124–137, Mumbai, India. Association for Computational Linguistics.

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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

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.

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*. ArXiv:2305.16307.

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.

Ali Hatami, Shubhanker Banerjee, Mihael Arcan, Paul Buitelaar, and John Philip McCrae. 2024. English-to-low-resource translation: A multimodal approach for Hindi, Malayalam, Bengali, and Hausa. In *Proceedings of the Ninth Conference on Machine Translation*, pages 815–822, Miami, Florida, USA. Association for Computational Linguistics.

Tosho Hirasawa, Emanuele Bugliarello, Desmond Elliott, and Mamoru Komachi. 2023. Visual prediction improves zero-shot cross-modal machine translation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 522–535, Singapore. Association for Computational Linguistics.

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 *Proceedings of the Tenth International Conference on Learning Representations (ICLR 2022)*. OpenReview nZeVKeeFYf9.

Shaharukh Khan, Ayush Tarun, Ali Faraz, Palash Kamble, Vivek Dahiya, Praveen Pokala, Ashish Kulkarni, Chandra Khatri, Abhinav Ravi, and Shubham Agarwal. 2024. Chitranuvad: Adapting multi-lingual LLMs for multimodal translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 839–851, Miami, Florida, USA. Association for Computational Linguistics.

Deepak Kumar, Baban Gain, Kshetrimayum Boynao Singh, and Asif Ekbal. 2025. Does vision still help? multimodal translation with CLIP-based image selection. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 115–123, Mumbai, India. 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.

NLLB Team. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630:841–846.

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. Association for Computational Linguistics.

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 and Ondřej Bojar. 2025. Findings of WAT2025 English-to-Indic multimodal translation task. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 103–108, Mumbai, India. Association for Computational Linguistics.

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, 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. ArXiv:1907.08948. Presented at CICLing 2019.

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).

Maja Popović. 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.

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.

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. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR. ArXiv:2103.00020.

Pawan Rajpoot, Nagaraj Bhat, and Ashish Shrivastava. 2024. Multimodal machine translation for low-resource Indic languages: A chain-of-thought approach using large language models. In *Proceedings of the Ninth Conference on Machine Translation*, pages 833–838, Miami, Florida, USA. Association for Computational Linguistics.

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. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. ArXiv:1706.03762.

WAT2026 Organizers. 2026. WAT2026 English-to-Indic Multimodal Translation Task. https://ufal.mff.cuni.cz/wat2026english-indicmultimodaltranslation. Workshop on Asian Translation.

Zhiyong Wu, Lingpeng Kong, Wei Bi, Xiang Li, and Ben Kao. 2021. Good for misconceived reasons: An empirical revisiting on the need for visual context in multimodal 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 6153–6166, Online. Association for Computational Linguistics.

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.

Appendix

	Zero Shot		Fine-Tuned		Improvement (Δ)	
Language	BLEU	CHRF3	BLEU	CHRF3	BLEU	CHRF3
Test Split						
Hindi	31.99	57.44	42.09	65.29	+10.10	+7.85
Bengali	22.67	53.82	41.60	69.04	+18.93	+15.22
Malayalam	17.83	56.55	41.68	72.81	+23.85	+16.26
Odia	5.84	37.25	50.49	72.82	+44.65	+35.57
Macro Avg	**19.58**	**51.27**	**43.97**	**69.99**	**+24.39**	**+18.72**
Challenge Split						
Hindi	36.51	61.10	50.58	69.68	+14.07	+8.58
Bengali	17.61	54.36	38.08	68.46	+20.47	+14.10
Malayalam	16.89	53.59	32.44	66.62	+15.55	+13.03
Macro Avg	**23.67**	**56.35**	**40.37**	**68.25**	**+16.70**	**+11.90**

Table 7: Test and Challenge results.

Findings of the 2nd Patent Claims Translation/Evaluation Tasks at WAT2026

Toshiaki Nakazawa
National Institute of Informatics (NII)
nakazawa@nii.ac.jp

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

Isao Goto
Ehime University
goto.isao.fn@ehime-u.ac.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
ArtIntell Corp.
takashi.ieda@artintell.tech

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

Abstract

This paper presents the results and findings of the 2nd Patent Claims Translation/Evaluation Task at WAT2026. Building on the first task, which collected human error annotations for machine-translated patent claims, this year we introduce an *evaluation* task: given the source claims and the machine translations gathered last year, participants automatically annotate translation errors (spans, categories, and severities) and assign a holistic quality score to each translation. We received submissions from three teams covering the Japanese–English (JA–EN) and English–Japanese (EN–JA) directions. We evaluated the submissions against the organizers' reference annotations in terms of error-span detection, severity, and quality-score correlation and correlated the results with the design of each participating system. Our analysis highlights two observations. First, giving an LLM explicit word alignments mainly helps it detect errors that hinge on the source–target correspondence (mistranslation and terminology), while leaving target-only fluency errors unaffected. Second, locating errors and producing a good overall quality score behave as largely independent abilities: the system whose score correlated best with the reference detected the fewest errors, because it scored quality with a module separate from its error annotator.

1 Introduction

Machine translation (MT) based on neural models and large language models (LLMs) has reached a level where, in some domains and language pairs, its output rivals human translation. However, there is still no universally reliable way to *evaluate* MT quality. Even widely used neural metrics such as COMET (Rei et al., 2020) have been reported to behave unstably outside their training domains (Kocmi et al., 2025). This problem is especially acute for patent claims, which are long, single-sentence, structurally complex, and terminologically demanding, so that a seemingly small error—an omitted limitation or a wrong dependency between claim elements—can change the legal scope of the claimed invention.

The first Patent Claims Translation Task at WAT2025 (Nakazawa et al., 2025) addressed this difficulty by collecting *human* error annotations and holistic quality scores for machine-translated Japanese–English patent claims. That study confirmed that reference-free automatic metrics correlate poorly with human judgments in this domain, and it produced a set of human-annotated translation errors intended as training and development material for automatic evaluation.

Building directly on that effort, the 2nd task at WAT2026 shifts the focus from *translation* to *evaluation*. The machine translations collected in the first task serve as the material to be evaluated: given a source claim and its machine translation, a participating *evaluation* system must (i) annotate translation errors with their spans, MQM-style categories, and severities, and (ii) assign a holistic

Proceedings of the 13th Workshop on Asian Translation (WAT 2026), pages 155–163
November 9, 2026 ©2026 Association for Computational Linguistics

quality score from 0 to 100. The human annotations from WAT2025 are released as training data, and the organizers prepare new reference annotations on held-out translations as the test set.

This paper reports the task setup and the results of the three participating teams. Our contributions are: (1) we define and run the first automatic patent-claim evaluation task; (2) we propose an evaluation protocol that compares heterogeneous submissions against gold reference annotations at the character, span, severity, and score levels; and (3) we analyze how the design of each system—in particular, word-alignment conditioning and the separation of error annotation from quality scoring—manifests in the results.

2 Task Definition

The 2nd task is an *evaluation* task in two translation directions, Japanese-to-English (JA–EN) and English-to-Japanese (EN–JA). For each translated claim, a participating system is given the source claim and its machine translation (the output of an MT system collected in the first task) and must produce, in the same JSON format as the released training data, the following:

- **Error annotation.** The system marks error spans on either the source side (`input`) or the MT-output side (`output`), and assigns to each span a category from a three-level MQM-based hierarchy and a severity. The hierarchy follows the MQM framework (Lommel et al., 2014; Freitag et al., 2021) as adapted to patent translation in WAT2025 (Nakazawa et al., 2025); its top-level categories are *accuracy*, *fluency*, *terminology*, *style*, *other*, *source error*, and *non-translation*. For example, an incorrect article is annotated as *article* under *accuracy* > *mistranslation*.
- **Quality scoring.** The system assigns a holistic quality score from 0 to 100 to each translated claim.

The reference (gold) annotations are produced by the organizers in the same format. In the gold annotations, error spans are marked directly on a character grid aligned with the source and target text, categories are drawn from the MQM-based hierarchy, and severities take three levels (small, medium, large) including intermediate ranges.

The ultimate goal of the task is to develop automatic evaluation methods that assess patent-claim translations as reliably as expert human evaluators, both by localizing concrete errors and by producing calibrated quality scores.

	Training		Test	
	JA–EN	EN–JA	JA–EN	EN–JA
Source documents	13	13	2	2
Source claims	28	28	2	3
MT outputs / claim	6	4	5	4
Translated claims	168	112	10	12
Error annotations	1,130	1,017	93	66
Avg. quality score	75.5	66.0	34.4†	53.7†

Table 1: Statistics of the training and test data. The training data are the WAT2025 human-evaluation results; the test data are newly annotated by the organizers. †The test average is the reference MQM score derived from the gold annotations (Section 5), not a human score.

3 Data

Table 1 summarizes the data. The **training data** are the human-evaluation results of the first task (Nakazawa et al., 2025): for each direction, 28 claims from 13 documents were translated by six (JA–EN) or four (EN–JA) MT systems, and every translation was annotated with error spans and a holistic quality score, yielding 168 JA–EN and 112 EN–JA annotated translations. These annotations are released so that participants can develop and tune their evaluation systems.[1]

The **test data** consist of machine translations held out from the first task, re-annotated by the organizers to serve as the gold reference for this year's evaluation. The JA–EN test set is derived from two single-claim source documents, each paired with five different MT outputs (10 translated claims); the EN–JA test set is derived from two source documents containing three claims in total, each paired with four MT outputs (12 translated claims). Because the gold annotations are produced on the character grid, every error span can be mapped to a character range in the source or target text, which we use for the evaluation described in Section 5. In total the organizers annotated 93 (JA–EN) and 66 (EN–JA) errors on the test translations; the corresponding average reference MQM scores (Section 5) are 34.4 and 53.7 (Table 1).

4 Participants

Three teams submitted evaluation results (Table 2). We summarize each system below based on the

[1] https://github.com/Tzawa/pat-claim-trans

Team	Organization	Directions
Ehime-U	Ehime University	JA–EN
UTSK26	University of Tsukuba	JA–EN, EN–JA
Anonymous	(participant)	JA–EN, EN–JA

Table 2: Participating teams and the directions they submitted.

teams' system-description papers; Ehime-U and UTSK26 are analyzed in detail in Section 7.

Ehime-U Wang et al. (2026) predicts translation errors with an LLM (GPT-6-Astra) that is additionally given explicit word-level correspondences between the source and target claims. The intuition is that many patent-claim errors are alignment-level phenomena: a source word with no target counterpart is an omission candidate, a target word with no source counterpart an addition candidate, and a source word linked to the wrong target word a mistranslation candidate. The team compares six settings that differ in how the correspondence is obtained: four alignments from Unbalanced Optimal Transport (UOT; Arase et al. 2023)—Forward, Reverse, their intersection (AND) and union (OR)—one alignment produced directly by the LLM (GPT6), and a no-alignment baseline (GPT6-noAlignment). The LLM outputs the error list and the quality score in a single response. The team submitted all six settings for JA–EN.

UTSK26 Fu et al. (2026) separates error annotation from quality scoring. Error annotation uses *GEMBA-Patent-Hybrid*, which combines a GEMBA-MQM-style prompt (Kocmi and Federmann, 2023) to GPT-5.2 that detects spans, categories, and severities, with a fallback that labels encoder-detected spans (Buma et al., 2025) when the LLM output is invalid. Quality scoring uses *Patent-Rubric-Checklist*, a rubric of five patent-claim-specific dimensions with sixteen checklist items, whose yes/partial/no answers are aggregated into a 0–100 score outside the LLM. Crucially, the quality score is computed independently of the error annotation.

Anonymous submitted error annotations and quality scores for both directions but did not submit a system-description paper. We therefore report its results without a detailed method analysis, and the team is anonymized here.

5 Evaluation Methodology

Submissions and gold annotations are compared at the claim level. Because submissions differ from the gold in coordinate system (raw-text offsets vs. character-grid offsets), in whether the claim number is included, and in whitespace handling, raw `start`/`end` offsets are not directly comparable. We therefore **re-anchor** every annotation to a non-whitespace character position in the gold text: using each annotation's verbatim span text together with its reported offsets, we align the submission's non-whitespace character sequence to the gold's with a standard sequence-alignment procedure and map the span onto gold coordinates. As a sanity check, evaluating the gold against itself yields $F_1 = 1.0$ on all metrics.

We report the following metrics.

Error detection. Treating each character as "marked as an error or not," we compute character-level precision, recall, and F_1 (on the `input` side, the `output` side, and combined). Precision is the fraction of the system's marked characters that are also marked in the gold (fewer false alarms); recall is the fraction of gold-marked characters recovered (fewer misses). We additionally report a span-level detection F_1 in which a predicted and a gold span are matched when their intersection-over-union is at least 0.5.

Category and severity. Because the gold and the submissions use different category inventories, we do not score category identity directly; instead we report per-gold-category recall (how well each gold category is *located*, regardless of the predicted label) and, where useful, a confusion mapping from gold to predicted categories. For severity, we report the agreement rate over characters marked by both sides.

Quality score. From the gold annotations we derive a reference MQM score in the GEMBA-MQM style (Kocmi and Federmann, 2023): each error contributes a weight (small $=1$, medium $=3$, large $=5$; intermediate ranges take the midpoint; spans left unlabeled count as minor), the per-claim penalty is the sum of weights, and the score is $100 + 4 \cdot \max(-25, -\text{penalty})$, so that scores lie in $[0, 100]$. We correlate this reference score with each team's submitted score using Pearson's r and Spearman's ρ at the claim level ($n = 10$ for JA–EN, $n = 12$ for EN–JA). We emphasize that the reference MQM score is a construct of this formula

and weighting, not a human score, and that the small sample size makes the correlations indicative rather than definitive.

6 Results

Table 3 reports the official results. For **error detection**, the Ehime-U settings lead in JA–EN, with character-level F_1 between 0.454 and 0.512; the best setting is GPT6 (the dense LLM alignment). Anonymous achieves high recall but low precision (0.302), indicating substantial over-annotation, whereas UTSK26 annotates far fewer errors than the gold (3.1 vs. 9.3 annotations per claim) and consequently has the lowest recall and F_1. Span-level F_1 (IoU$\geq$0.5) is uniformly lower than character-level F_1 (0.07–0.28), showing that systems localize the approximate region of an error more reliably than its exact boundaries.

For **score correlation**, the picture is markedly different from detection. In JA–EN, UTSK26 attains the highest Spearman correlation (0.727) despite having the weakest error detection, and Ehime-U/GPT6 achieves the best overall balance ($r = 0.649$, $\rho = 0.633$). We analyze these two contrasting outcomes—the effect of alignment on detection, and the detection–scoring split—in the next section.

The gold MQM scores derived for the test set average 34.4 (JA–EN) and 53.7 (EN–JA); submitted scores are substantially higher on average (around 75 in JA–EN), so absolute score agreement (mean absolute error) is poor even where the rank correlation is moderate-to-high. This systematic leniency is discussed in Section 7.

7 Analysis

7.1 Ehime-U: alignment conditioning selectively helps alignment-level errors

Table 4 breaks the Ehime-U settings down by gold category. Although the combined F_1 varies only slightly across settings—consistent with the team's own observation that the choice of correspondence mainly changes *how many* errors are reported rather than the dominant error types—the per-category view reveals a structured effect. Moving from the no-alignment baseline to the dense LLM alignment (GPT6) raises Accuracy recall from 0.34 to 0.40 and Terminology recall from 0.17 to 0.34 (roughly doubling), while Fluency recall is essentially unchanged (0.51).

This pattern matches the intuition behind the system: alignment information helps errors that depend on the source–target correspondence—mistranslation (Accuracy) and inconsistent terminology—but offers no help for errors that can be judged monolingually on the target, such as fluency. The result thus provides gold-grounded support for Ehime-U's central hypothesis that a subset of patent-claim errors are alignment-level phenomena. Among the UOT variants, the sparse-but-reliable intersection (AND) is strongest, whereas the single-direction Forward alignment falls below the no-alignment baseline, suggesting that dense, bidirectional, high-precision correspondence is the most useful.

7.2 Error localization and holistic scoring behave as separate abilities

In WMT metric evaluation, error-span detection and segment-level quality scoring are treated as distinct subtasks (Freitag et al., 2024; Rei et al., 2023), and a system can be strong on one without the other. Our submissions provide a concrete illustration of the same decoupling in the patent-claim domain. UTSK26 has the lowest detection F_1 in both directions, yet its submitted quality score attains the best Spearman correlation with the gold MQM score in JA–EN (0.727). The explanation lies in its design: UTSK26 computes the quality score with *Patent-Rubric-Checklist*, a rubric evaluated independently of the error annotation, so a sparse or incomplete error list does not prevent the score from tracking overall quality. This is consistent with the UTSK26 paper's own report that the rubric-based score correlates best with human scores on the training data. By contrast, Ehime-U emits errors and score from a single prompt, which yields balanced but not dominant performance on both axes.

Two implications follow. First, a reliable holistic quality score does not require exhaustive error localization; the two capabilities can and perhaps should be optimized separately. Second, if the task goal includes explainability—pointing to the concrete errors behind a score—then UTSK26's sparse annotation is a practical weakness. Indeed, its low annotation density on the test set (Table 3) mirrors the "empty annotation" behavior the team identifies in its own paper, where an empty error list from the LLM is accepted as valid and the encoder-based fallback is not invoked.

Dir.	Team	Ann./ claim	Char-level detection F_1	P	R	Span F_1 (IoU≥.5)	Score corr. Pearson	Spearman
JA–EN	Ehime-U/GPT6	11.2	**0.512**	0.606	0.443	0.283	0.649	0.633
JA–EN	Ehime-U/AND	11.2	0.486	0.588	0.414	0.254	0.635	0.590
JA–EN	Ehime-U/Reverse	10.3	0.485	0.600	0.407	0.276	0.601	0.652
JA–EN	Ehime-U/OR	10.6	0.482	0.584	0.410	0.261	0.613	0.590
JA–EN	Ehime-U/GPT6-noAlignment	11.1	0.475	0.580	0.402	0.255	0.634	0.623
JA–EN	Ehime-U/Forward	11.0	0.454	0.565	0.379	0.256	0.633	0.561
JA–EN	Anonymous	15.8	0.375	0.302	0.494	0.104	0.674	0.364
JA–EN	UTSK26	3.1	0.111	0.121	0.102	0.097	0.654	**0.727**
EN–JA	Anonymous	22.5	0.154	0.089	0.562	0.071	0.604	0.600
EN–JA	UTSK26	4.2	0.139	0.183	0.112	0.155	0.383	0.320

Table 3: Official results per team and direction. "Ann./claim" is the average number of annotations per claim (for reference, the gold density is 9.3 for JA–EN and 5.5 for EN–JA). Detection is character-level against the gold; score correlation is between the gold MQM score and the submitted score ($n = 10$ for JA–EN, $n = 12$ for EN–JA). Best values in each column group are in bold.

Setting	Det F_1	Det F_1^{out}	Recall by gold category Acc.	Term.	Flu.
GPT6 (dense)	**0.512**	**0.555**	**0.40**	**0.34**	0.51
UOT-AND	0.486	0.528	0.36	0.17	0.48
UOT-Reverse	0.485	0.523	0.35	0.17	0.51
UOT-OR	0.482	0.526	0.35	0.34	0.52
no-align baseline	0.475	0.510	0.34	0.17	0.51
UOT-Forward	0.454	0.485	0.28	0.31	0.52

Table 4: Ehime-U six settings (JA–EN). "Det F_1^{out}" is the output-side character F_1; "Recall by gold category" is how well the system locates gold errors of each top-level category (Accuracy, Terminology, Fluency).

7.3 Systematic leniency of submitted scores

Across teams, submitted quality scores are considerably higher than the gold MQM scores (e.g., a JA–EN gold mean of 34.4 against submitted means around 75). Because Pearson and Spearman correlations are invariant to such an offset, the moderate correlations coexist with large mean absolute errors. This indicates that current systems rank translations by quality more reliably than they calibrate the absolute score, and it highlights the sensitivity of the reference MQM score to the severity weighting and the per-segment clipping at -25 (which places several heavily penalized claims at 0).

7.4 Reference-score gap between the two directions

The reference MQM scores differ markedly between directions (JA–EN mean 34.4, EN–JA 53.7). The gap is mechanical: JA–EN test translations carry about 70% more error annotations per claim than EN–JA (9.3 vs. 5.5) and a larger share of high-weight errors (14 large and 5 medium–large, vs. 7 and 8), so the per-claim penalty is larger and more JA–EN claims hit the -25 floor (three of ten, vs. one of twelve), pulling the mean down. We caution against reading this as a general quality gap between directions: each direction has only two source documents and 10–12 translations, so a few heavily penalized claims dominate the average, and the reference score reflects *how many* errors the organizers marked rather than a human holistic judgment. Indeed, it runs opposite to the WAT2025 training human scores, where JA–EN averaged higher than EN–JA (75.5 vs. 66.0; Table 1).

8 Discussion

Heterogeneity of submission formats. The submissions varied in coordinate system (raw-text vs. character-grid offsets), in whether the leading claim number was retained, and in whitespace, so that a naive comparison of `start`/`end` offsets was impossible. We absorbed these differences by re-anchoring every span to the gold text (Section 5), but for a shared task a stricter output specification—fixed coordinate system, a shared category vocabulary, and a fixed set of severity levels—would make submissions directly comparable and reduce the risk of scoring artifacts.

Mismatch of category inventories. The gold annotations use abstract MQM top-level categories (accuracy, fluency, terminology), whereas the submissions use finer labels (omission, mistranslation, article, incorrect dependency, etc.). We therefore evaluated *where* errors were found rather than *whether the exact label matched*. A confusion mapping from gold to predicted categories (produced

as part of the evaluation) shows, for example, that the gold *accuracy* class maps mostly to predicted *mistranslation* and *dependency* errors, and *fluency* to *article* and *grammar*. For future editions, agreeing on a single label set—or an explicit crosswalk between label sets—would allow category-level scoring.

Severity granularity. The gold uses three severity levels (small, medium, large) plus intermediate ranges, while the submissions use two (major, minor). The treatment of "medium" in the MQM score (we used a midpoint weight of 3) directly affects the reference scores and hence the correlations; aligning the severity scales across gold and submissions is desirable.

Detection vs. scoring. Error-span detection and segment-level scoring are already separate subtasks in WMT metric evaluation (Freitag et al., 2024). The observation in Section 7.2 suggests that the patent task should likewise report the two capabilities on separate leaderboards: error localization (span/character F_1, severity) and holistic scoring (correlation with the reference). A system can be strong on one and weak on the other, and conflating them would obscure where the state of the art actually stands.

Small test set. The current test set is small (10 JA–EN and 12 EN–JA claims). While it is sufficient to expose qualitative effects—such as the selective benefit of alignment and the detection/scoring dissociation—larger and more diverse test data are needed to draw robust quantitative conclusions and to compute stable correlations.

9 Conclusion and Future Perspective

We described the 2nd Patent Claims Translation/Evaluation Task at WAT2026, which reframes the first task's translation effort as an *evaluation* task: participating systems annotate translation errors and assign quality scores to machine-translated patent claims, and are compared against the organizers' reference annotations. Three teams participated, covering both translation directions.

Our analysis yields two main findings. First, conditioning an LLM on explicit word alignments (Ehime-U) selectively improves the detection of alignment-dependent errors—accuracy and terminology consistency—while leaving monolingual fluency errors unchanged, giving gold-grounded support to the hypothesis that some patent-claim errors are alignment-level phenomena. Second, consistent with the separate error-detection and scoring subtasks in WMT metric evaluation, error localization and holistic scoring behaved as largely independent abilities here: a system with a separate rubric-based scorer (UTSK26) ranked translations best despite detecting the fewest errors.

In future editions we plan to enlarge and diversify the test set, standardize the submission format and the category/severity inventories so that submissions are directly comparable, and report error localization and holistic scoring as separate tracks. We also intend to use the growing body of human and reference annotations as training data for more accurate and better-calibrated automatic evaluation of patent-claim translation.

Limitations

The findings reported here are subject to several limitations. The test set is small (10 JA–EN and 12 EN–JA claims), so detection scores and especially correlations are indicative rather than definitive and are sensitive to a few rank swaps. The reference MQM quality score is a construct of this paper (severity weights small/medium/large $= 1/3/5$, ranges at their midpoint, unlabeled spans counted as minor, and per-segment clipping at -25); the choice of the "medium" weight and of clipping affects the reported correlations, and the score is not a human judgment. Because the gold and the submissions use different category inventories, we evaluate error *location* rather than exact category identity. Input-side (source) gold annotations are very sparse (only two in JA–EN), so the input-side metrics are low-signal and our evaluation is effectively output-centric. Span-level F_1 requires boundary agreement at IoU ≥ 0.5 and is therefore conservative; the character-level metric better reflects whether an error location was found. Finally, each system's LLM output is essentially a single run, so output stability is not assessed.

Acknowledgments

We thank the participating teams for their submissions and system descriptions. The human annotation of the reference data is supported by the Asia-Pacific Association for Machine Translation (AAMT) and the AAMT/Japio[2] Special Interest Group on Patent Translation.

[2]Japan Patent Information Organization

References

Yuki Arase, Han Bao, and Sho Yokoi. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3966–3986, Toronto, Canada. Association for Computational Linguistics.

Kosei Buma, Takehito Utsuro, and Masaaki Nagata. 2025. Two step automatic post editing of patent machine translation based on pre-trained encoder models and LLMs. In *The 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 218–231, Mumbai, India. Association for Computational Linguistics.

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.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81, Miami, Florida, USA. Association for Computational Linguistics.

Yingyi Fu, Kosei Buma, and Takehito Utsuro. 2026. UTSK26 at WAT2026 patent claims translation/evaluation tasks. In *Proceedings of the 13th Workshop on Asian Translation (WAT 2026)*. System description paper.

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.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):455–463.

Toshiaki Nakazawa, Takashi Tsunakawa, Isao Goto, Kazuhiro Kasada, Katsuhito Sudoh, Shoichi Okuyama, Takashi Ieda, and Masaaki Nagata. 2025. Findings of the first patent claims translation task at WAT2025. In *Proceedings of the Twelfth Workshop on Asian Translation (WAT 2025)*, pages 1–15, Mumbai, India. Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, José 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.

Xu Yin Wang, Isao Goto, and Takashi Ninomiya. 2026. Ehime-U system for the WAT 2026 patent claims evaluation task: LLM-based translation error prediction using word alignments. In *Proceedings of the 13th Workshop on Asian Translation (WAT 2026)*. System description paper.

A Re-anchoring Procedure

Submissions differ from the gold in coordinate system (raw-text vs. character-grid offsets), in whether the leading claim number is kept, and in whitespace, so a span's reported `start`/`end` cannot be compared to the gold directly (Section 5). We map every annotation onto a common coordinate—the index of a non-whitespace character in the gold text—as follows.

1. For an annotation, pick the reference string whose `[start:end]` slice equals its verbatim `text`: the character grid if present, otherwise the raw text. This identifies which coordinate system the submission used.

2. Convert the span's endpoints to *non-whitespace* offsets by counting non-whitespace characters before each endpoint. This removes all whitespace differences.

3. Build the non-whitespace character sequences of the submission and the gold for that side and align them with a standard sequence

Dir.	Claim	Penalty	Score
JA–EN	01	19	24
JA–EN	02	9	64
JA–EN	03	19	24
JA–EN	04	50	0
JA–EN	05	17	32
JA–EN	06	6	76
JA–EN	07	0	100
JA–EN	08	25	0
JA–EN	09	42	0
JA–EN	10	19	24
EN–JA	01-1	16	36
EN–JA	01-2	0	100
EN–JA	02-1	20	20
EN–JA	02-2	6	76
EN–JA	03-1	10	60
EN–JA	03-2	14	44
EN–JA	04-1	11	56
EN–JA	04-2	0	100
EN–JA	05	11	56
EN–JA	06	47	0
EN–JA	07	15	40
EN–JA	08	11	56

Table 5: Reference MQM penalty and score per test claim (Eq. 1). Means: 34.4 (JA–EN), 53.7 (EN–JA).

matcher (Python's `difflib`); matched positions give a map from submission non-whitespace indices to gold non-whitespace indices. This absorbs the claim-number and minor text differences.

4. Map the span's non-whitespace endpoints through this map to obtain its position in the gold coordinate.

Gold annotations are expressed in the gold coordinate directly (step 3 is the identity). As a sanity check, evaluating the gold against itself with this procedure yields $F_1 = 1.0$ on every metric, confirming that the re-anchoring is lossless.

B Reference MQM Score

The reference quality score follows the GEMBA-MQM convention. Each error contributes a severity weight (small $= 1$, medium $= 3$, large $= 5$; a severity range, e.g. medium-to-large, takes the midpoint of its endpoint weights; spans left unlabeled count as minor). The per-claim penalty is the sum of weights, and the score is

$$\text{score} = 100 + 4 \cdot \max(-25, -\text{penalty}), \quad (1)$$

so that a clipped score lies in $[0, 100]$. Table 5 lists the penalty and score for every test claim; EN–JA files with two claims are shown as `file-block`.

Dir.	Team	Acc.	Term.	Flu.	Unlab.
JA–EN	Ehime-U GPT6	0.40	0.34	0.51	0.51
JA–EN	Ehime-U AND	0.36	0.17	0.48	0.53
JA–EN	Ehime-U Reverse	0.35	0.17	0.51	0.51
JA–EN	Ehime-U OR	0.35	0.34	0.52	0.48
JA–EN	Ehime-U GPT6-noAlignment	0.34	0.17	0.51	0.51
JA–EN	Ehime-U Forward	0.28	0.31	0.52	0.51
JA–EN	Anonymous	0.30	0.14	0.74	0.83
JA–EN	UTSK26	0.15	0.17	0.06	0.01
EN–JA	Anonymous	0.42	0.86	0.96	0.61
EN–JA	UTSK26	0.08	0.03	0.17	0.18

Table 6: Per-gold-category recall (Accuracy, Terminology, Fluency, Unlabeled).

Gold category	Predicted categories (char counts)
Ehime-U/GPT6	
Accuracy	mistranslation (190), dependency (157), numerals/symbols (25), article (21), grammar (17)
Fluency	article (128), grammar (21), punctuation (3)
Terminology	mistranslation (17), terminology (17)
Unlabeled	omission (99), dependency (78), over-translation (41), mistranslation (30), article (10)
UTSK26	
Accuracy	numerals/symbols (73), omission (38), context/word choice (38), grammar (10)
Fluency	omission (12), article (6)
Terminology	context/word choice (17)
Unlabeled	mistranslation (7)

Table 7: Gold-to-submission category confusion (overlapping characters) for two JA–EN systems. Submission labels are glossed in English (e.g. *dependency*, *over-translation*, *omission*).

C Per-Category Recall

Table 6 gives, for every team, the recall for each top-level gold category—how well the system *locates* gold errors of that category, regardless of the predicted label (Section 5). "Unlab." denotes gold spans highlighted without a category label.

D Category Confusion (Gold → Submission)

Because the gold and the submissions use different category inventories, we do not score label identity; instead, for characters marked by both sides, we count which predicted category a gold category maps to. Table 7 shows this mapping (character counts) for two representative JA–EN systems. The gold's abstract *accuracy* class maps mostly to the

submissions' concrete *mistranslation* / *dependency* labels, and *fluency* maps to *article* and *grammar*, confirming that the two inventories are compatible once the abstraction gap is accounted for.

Author Index

Abedin, Mantaqa, 104
Amimoto, Yuri, 17
Azeemi, Abdul Hameed, 1

Badzohreh, Mohammad, 46
Barati, Iman, 46
Basit, Abdul, 1
Buma, Kosei, 119

Choudhary, Ritvik, 17

Dawadi, Sundeep, 145

Fu, Yingyi, 84, 119

Gautam, Abhishek, 9
Ghimire, Prajwal, 145
Goto, Isao, 134, 155

Hida, Rem, 17

Ieda, Takashi, 155

Khan, Jeba Afia, 104

Mahato, Aashish, 145
Mitani, Ryosuke, 17

Nagata, Masaaki, 84, 155
Nakazawa, Toshiaki, 155
Ninomiya, Takashi, 134

Okuyama, Shoichi, 155

Regmi, Sunil, 145
Rostamkhani, Mohammadmostafa, 46

Saud, Ramesh, 145
Saud, Shiv Ram, 145
Sekiya, Toshiyuki, 17
Sudoh, Katsuhito, 155

Tsunakawa, Takashi, 155

Utsuro, Takehito, 84, 119

Yin, Wang Xu, 134

Zhang, Yiran, 71
Zhang, Ziqi, 71