HUFS-DILAB at WMT 2026 Automated Translation Quality Evaluation Task 1: Language-Pair Prompt Routing for Error Span Annotation

Jisoo Jang, Haewon Cho, and **Seungtaek Choi***
Division of Language and AI, Hankuk University of Foreign Studies
{jsjang0104,hcho12977,seungtaek.choi}@hufs.ac.kr

Abstract

This paper describes HUFS-DILAB's submission to Task 1, segment-level error detection and span annotation, of the WMT26 Automated Translation Quality Evaluation task. We investigate how far an off-the-shelf LLM can be taken as a judge without task-specific training: Gemma-4-31B-it is prompted to annotate error spans, omissions, and severities as constrained JSON given the source, the reference, and the translation, with each language pair routed to a prompt variant specialized for that pair. We calibrate these prompts by comparing the judge's severity, omission, and error-density distributions against human annotation, and correct a rule-based wrong-language filter whose false positives on closely related languages were silently overriding the judge. Our submission scores 0.5854 on the official metric, averaged over the eight prioritized language pairs.

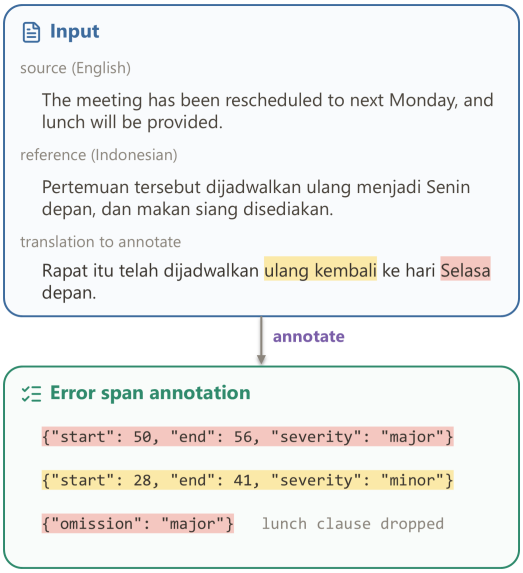


Figure 1: An example segment and its Task 1 annotation. Highlight color marks severity; the clause about lunch, dropped from the translation, has no span and is reported as a major omission.

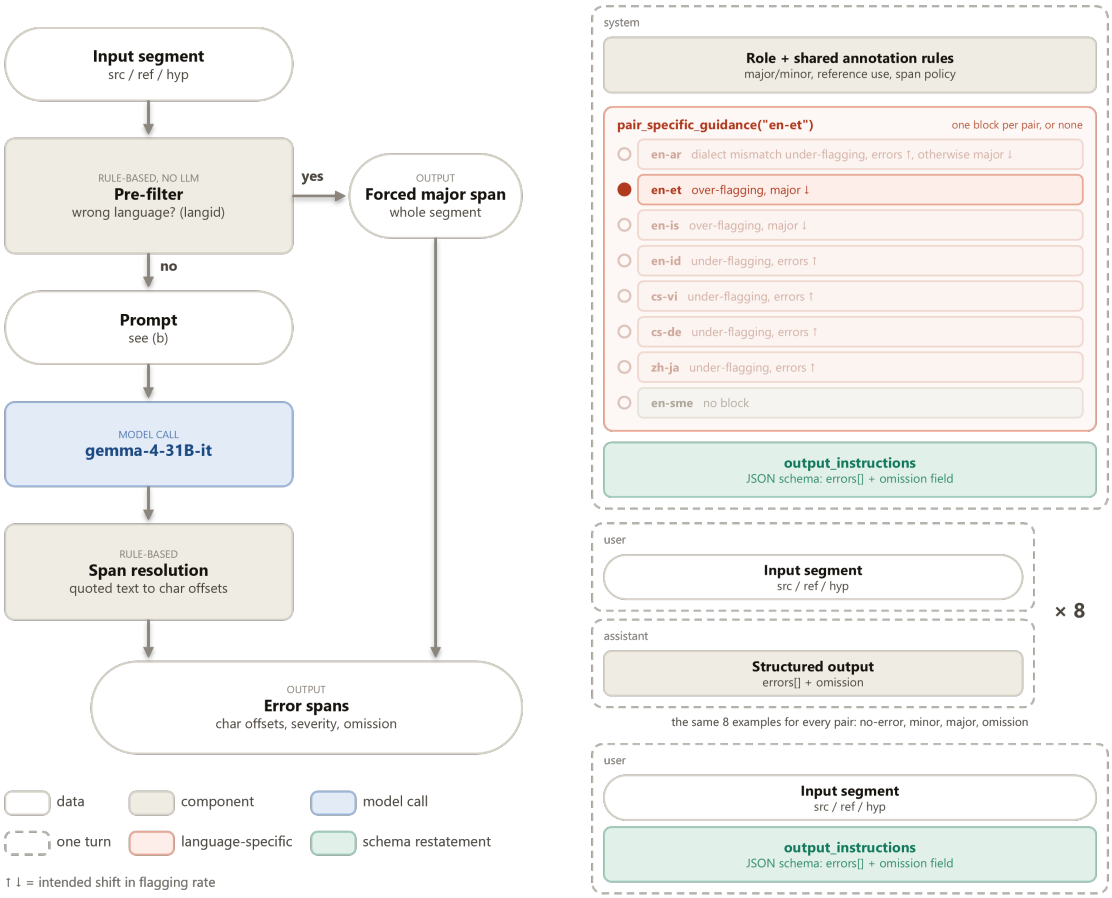
1 Introduction

Automated evaluation of machine translation (MT) has moved from a single quality score towards fine-grained error annotation. Task 1 of the WMT26 Automated Translation Quality Evaluation shared task[1] asks a system to read a source segment and its translation and mark the parts that are wrong: a set of character spans, each labelled *major* or *minor*, together with a segment-level judgement of whether anything was omitted (Figure 1). This paper describes HUFS-DILAB's submission, covering the eight pairs the organizers designate as the priority set: cs→de, cs→vi, zh→ja, en→ar, en→et, en→is, en→id, and en→sme.

Our system is an LLM judge used without any task-specific training (Zheng et al., 2023): one instruction-tuned model, prompted with eight fixed demonstrations and decoding the annotation as schema-constrained JSON. Such judges are usually driven by a single prompt applied uniformly to every language pair. However, the judge's annotation behavior deviates by pair, and the deviations are stable enough to measure. Across the evaluated MT systems, the uncalibrated judge marks *major* errors 19.6 points more often on zh→ja than its average on the other pairs, a gap that holds in 24 of the 26 systems that cover the priority set; on en→id, it answers that a segment has no errors 15.7 points more often, in 25 of 26. We read these gaps as a bias in the judge rather than a property of the data: the model is more generous on some language pairs than on others, whatever segments it is given.

Rather than retrain the judge, we counteract this bias in its prompt. Each pair is routed to its own short calibration block in the system message, derived from the distribution gaps and validated against human annotation where the previous year's

* Corresponding author.
[1] https://www2.statmt.org/wmt26/mteval-task.html



(a) Inference path.

(b) Prompt composition.

Figure 2: Overview of our system. (a) A segment either short-circuits through the wrong-language pre-filter, which emits one segment-wide *major* span without calling the model, or is annotated by the judge and passed to span resolution. (b) The prompt: shared annotation rules, a calibration block shown instantiated for en→et, eight fixed demonstrations, and the segment to annotate, with the JSON schema restated after it.

gold data covers the pair.

Before the model, a rule-based filter compares the detected language of the translation against the target, and when the two differ, the whole segment is annotated as a single *major* span. Language identification, however, makes plausible mistakes, such as fluent Indonesian as Malay, and each one marks a correct translation wrong from end to end. We thus ignore detections of languages known to be confused with the target and leave those segments to the model.

Our submission scores 0.5854 on the official metric, averaged over the eight prioritized pairs, and ranks between first and third on the seven pairs that the leaderboard covers.

2 Related Work

Prompted language models are now competitive evaluators of translation quality. Kocmi and Federmann (2023b) showed that a GPT model asked for a quality score correlates with human judgement at the system level, and GEMBA-MQM (Kocmi and Federmann, 2023a) extended the idea to error spans, prompting GPT-4 (OpenAI, 2023) for the span, category, and severity of each error. The annotation these systems imitate is MQM (Freitag et al., 2021), whose taxonomy is expensive to apply, and the shared task we enter uses its lighter successor, Error Span Annotation (Kocmi et al., 2024), which keeps target-side spans and the *major*/*minor* distinction and drops the taxonomy. Agreement at the system level, however, does not carry down to the span: predicted spans overlap human ones only loosely (Lu et al., 2025), and in last year's edition of this shared task no automatic system passed 35 micro-F1 on span-level error detection, far short of what a human annotator scored the same way (Lavie et al., 2025). That report also notes how unevenly the automatic systems were spread across language pairs, with none of them ahead every-

where, and it names the failure modes an LLM-era metric now has to survive rather than assume away: verbose output, and output in the wrong language, dialect, or script. Task 1 asks for exactly the annotation that gap is measured on.

What such a judge does also depends on the language pair. Hada et al. (2024) report that a GPT-4 evaluator agrees with human raters to different degrees across languages and scores some of them systematically higher, and recommend validating against native-speaker judgements language by language. Calibrating a judge from human labels is itself an established move: Liu et al. (2024) generate and refine the scoring criteria handed to an evaluator from labeled examples. We share that diagnosis and correct it more narrowly: we build our judge on an open-weight model and leave the model alone, so the correction is a few lines of instruction per language pair, derived from where that pair's output distribution sits relative to human annotation or to the other pairs.

3 Approach

The goal of our submission was to see how far an off-the-shelf model can be taken as a judge by prompting alone: one model, no fine-tuning, and everything else expressed as instructions. The system is a rule-based pre-filter, a single model call with a prompt assembled for the language pair, and a resolver that maps the strings the model quotes back to character offsets (Figure 2).

3.1 System overview

A segment first meets the **wrong-language pre-filter** (Figure 2a). A translation that is simply in the wrong language is not a span-level annotation problem, and asking the judge to mark it wastes a model call and usually produces a partial annotation, so we compare the detected language of the translation against the target with `langid.py` (Lui and Baldwin, 2012) and emit a mismatch as one segment-wide *major* span, with no model call. The first version of this filter was too eager — Indonesian and Malay are close enough that the classifier frequently misclassified correct Indonesian outputs as Malay, and such segments were excluded from annotation — so a detection counts only when the detected language is not a known confusion of the target: we exempt Malay for an Indonesian target, Finnish for Estonian, and Chinese for Japanese, where the writing system is largely shared. The filter is disabled altogether for en→sme, where `langid.py` has no support for Northern Sámi.

Every other segment goes into the prompt of Figure 2b, and `google/gemma-4-31B-it` (Gemma Team, 2026) returns the annotation as JSON.[2] The task expects the annotation as a set of spans with severities, so we constrain decoding to a schema (Geng et al., 2023; Willard and Louf, 2023) with two fields: a list of errors, each carrying the quoted error text and a severity in {*major*, *minor*}, and a segment-level omission field that is either null or one of the two severities (Appendix A.2). We ask for the erroneous text rather than the character offsets the task requires, since a quotation can be checked against the translation — it either occurs in it or it does not — while an index cannot; our own code locates each quoted string and drops the spans it cannot locate. Generation can still hit the length cap before the JSON is complete, which the schema cannot prevent; a separate parser repairs or discards those responses. The system finally produces a structured output: a list of spans with offsets and severities, plus the omission flag.

3.2 Prompt composition

The system message comes first, stating the judge's role, the shared annotation rules, the JSON schema, and, for some language pairs, a short calibration block; Appendix A reproduces it in full. The rules tell the judge to use the reference as evidence without requiring the translation to match it, and to report a span whenever it is more likely wrong than acceptable, without waiting for certainty.

We counteract the judge's pair-dependent bias with a few lines of extra instruction written for the pair being annotated, which we call a **calibration block** (Appendix A.3). It is the one part of the prompt that is not shared: exactly one block is inserted per request, and en→sme is the only pair that receives none. We derive the blocks from gaps in the judge's output distribution, and keep a block only when it improves every metric on a fixed validation set. A block states the direction in which the judge's output distribution deviates — an excess of *major*, or of segments left unannotated — and instructs it to move the opposite way. The red `pair_specific_guidance` block of Figure 2b

[2] We run the model with thinking disabled. It is served with vLLM (Kwon et al., 2023), whose structured-outputs interface enforces the schema during decoding, and decoded deterministically at temperature 0, with the context capped at 24,576 tokens and generation at 4,096 tokens.

gives the direction taken for each pair.

Eight in-context demonstrations (Brown et al., 2020) follow the system message, one for each of the officially prioritized language pairs. All eight go into every request, whatever pair is being annotated. Their annotations are intended to cover the output space — a clean translation, single *minor* and *major* cases, an omission, a dialect mismatch, and a segment with several errors at once. Four of the eight are taken from the human-annotated data released with the WMT25 General Machine Translation shared task (Kocmi et al., 2025), and we wrote the other four ourselves, for the pairs that data does not include.

The input segment comes last, followed by a restatement of the JSON schema, so the format instruction is the final thing the judge reads. Stating it only once, in the system message, degraded format adherence in our runs as the prompt grew longer.

4 Experimental Setup

4.1 Datasets

For development, we use the human-annotated data released with the WMT25 General Machine Translation shared task (Kocmi et al., 2025), holding out 260 of its 2,840 segments as a validation set. They are drawn per language pair in proportion to that pair's size, from a document-level shuffle under a fixed seed, so that the split is reproducible and no document is split across the two halves. Four of the eight officially prioritized pairs (§1) — cs→de, en→ar, en→et, and en→is — also appear in that data. Those four are therefore the only pairs we can validate against human annotation.

4.2 Evaluation Metric

We use the official metric, Match with Partial Overlap and Partial Credit (MPP; Perrella et al., 2026). MPP computes micro-averaged span-level precision, recall, and F-score by matching predicted spans to gold spans one-to-one. It awards partial credit for overlapping spans and accounts for *major* and *minor* severities. We compare validation runs by MPP F-score. Our validation figures use the greedy matcher rather than the optimal one, consistently across every run, so they are comparable to each other; the official leaderboard scores in Table 1 come from the organizers' own scoring and are not on this scale.

Language pair	Score	Rank
zh→ja	0.6500	1
cs→de	0.5663	2
cs→vi	0.5765	2
en→et	0.6086	3
en→is	0.6650	3
en→ar	0.5322	3
en→id	0.5502	3
en→sme	0.5343	—
Average	0.5854	

Table 1: Official scores on the eight prioritized language pairs, from the scoring output of our final submission. Ranks count each team's best published submission; en→sme is scored but unranked, as the leaderboard never displayed it.

5 Results

Table 1 gives our official scores, which average 0.5854 over the eight pairs, or 0.5927 over the seven leaderboard ranks. We placed first on zh→ja, second on cs→de and cs→vi, and third on the four remaining ranked pairs; en→sme is scored but never displayed on the leaderboard, so it carries no rank. Ranks count each team's best submission among the results teams chose to publish, and publishing is optional, so they can only fall as more appear.[3] The scores span 0.5322 to 0.6650, though our highest and our lowest both come from pairs where we placed third. The rest of this section takes the system apart: what each design decision contributed on the validation set (§5.1), and then the pair-dependent bias and the calibration blocks written against it (§5.2).

5.1 Ablation

Table 2 tracks the four decisions that shaped the system, each measured on the same held-out segments. Constraining decoding to the schema recovers errors that a free-text judge finds but loses to parsing, and is worth roughly two F1 points. Giving the judge the reference translation, with annotation rules written around it, hands it evidence for dropping a span it cannot support, and adds two more. Moving to the 31B checkpoint adds a further two and a half. Pair calibration, the last row, adds one more point, but pairs carrying no block at all also move, one by as much as 6.4 F1, so that average is

[3] Read from the official leaderboard on 7 August 2026. Note that they are a snapshot of an open leaderboard rather than final standings.

Configuration	P	R	F1
Free-text output, 12B model	16.0	20.0	16.2
+ JSON schema	16.6	23.9	18.0
+ reference, annotation rules	20.6	23.2	20.0
+ 31B model	23.7	24.0	22.5
+ pair calibration	23.2	26.6	**23.5**

Table 2: Validation results, macro-averaged over the 13 language pairs of the held-out set that every configuration covers. Each row adds one design decision to the row above it; the first three run on `google/gemma-4-12B-it` and the last two on the `31B` checkpoint the submission uses.

Pair	Deviation	Correction	ΔF1
major share, vs. validation			
en→ar	+20.1	dialect, *major* ↓	+1.6
en→et	+31.4	*major* ↓	+1.8
en→is	+22.3	*major* ↓	−0.3
error-free share, vs. other pairs			
zh→ja	+11.4	errors ↑	—
en→id	+9.8	errors ↑	—
cs→vi	+9.7	errors ↑	—
cs→de	+7.1	errors ↑	—
en→sme	—	none	—

Table 3: The per-pair corrections in the submitted system. **Deviation** is measured before the correction was added. For the first group, it is the share of a pair's spans marked *major* against the validation set; for the second, the share of its segments left error-free against the other pairs. **ΔF1** is the change in validation F1 afterwards, available only for the pairs the validation set covers.

not the effect of the blocks alone.

5.2 Pair-Dependent Bias

The judge does not annotate every language pair the same way. We compare pairs within a single MT system, on two quantities: the share of a pair's spans that carry the *major* label, and the share of its segments left without any span. With the shared prompt alone, zh→ja takes *major* 19.6 points more often than the average across the other pairs, in 24 of the 26 systems. On en→id, the judge leaves 15.7 points more segments without a span, in 25 of 26. We take a deviation that survives nearly every system to be the judge's own, not the data's.

Table 3 gives the deviation behind each of the seven blocks. The axis a block adjusts follows from the signal that diagnosed it: pairs diagnosed on the validation set deviated on severity, while those diagnosed against the other pairs deviated on how often the judge reports anything.

Only three of the seven corrections can be checked on the validation set. The correction for en→is loses 0.3 F1 there; we kept it because the run that introduced it gained overall. The other four cannot: three target pairs the validation set does not cover, and cs→de shows no deviation there, only against the other pairs. For those four, the official score is the only check, and it separates one correction from everything else only where that correction was submitted on its own, as it was for cs→vi and zh→ja. On en→id, neither check is available, so all we can say is that its correction did what it was written to do: the share of segments left without a span fell from 59.4% to 52.1%. The largest severity gap of all belongs to zh→ja, yet the correction we kept there raises how often errors are reported and leaves severity alone. Tightening the severity definition was our first attempt, and it lowered the pair's score; the replacement took it from 0.6437 to 0.6500.

6 Conclusion

This paper presented HUFS-DILAB's system for WMT26 Task 1, an LLM judge that annotates error spans with no task-specific training. It scores 0.5854 across the eight prioritized language pairs, ranking between first and third on the seven the leaderboard covers. Its design follows from one observation: the judge's severity and error density deviate by language pair, and the deviation is stable enough across the MT systems being judged to be measured and corrected. We correct it in the prompt, routing each pair to a few lines of its own instruction while the model and the shared prompt stay fixed. Future work should look for a calibration signal on pairs with no human annotation, and test whether the same deviations appear in other judges.

7 Limitations

Our study has the following three limitations. First, our experiments used a single judge, `google/gemma-4-31B-it`. Though it performed best among the models we explored during development, we did not test the prompt routing and calibration on any other LLM, so how far the approach generalizes remains unclear. Second, our calibration is tuned to evaluation data that is assembled to be hard for MT, where errors are common. On translations that are mostly correct, the same in-

structions would cost precision. Lastly, we ask the judge for the erroneous text rather than character offsets. When that text occurs more than once in the translation, we take the first occurrence, which is not always the one the judge meant. Such segments are rare, but a span placed on the wrong occurrence both misses the real error and counts as a false positive.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Gemma Team. 2026. Gemma 4 technical report. *Preprint*, arXiv:2607.02770.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952.

Rishav Hada, Varun Gumma, Adrian De Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2024. Are large language model-based evaluators the solution to scaling up multilingual evaluation? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1051–1070.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, et al. 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.

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.

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.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Alon Lavie, Greg Hanneman, Sweta Agrawal, Diptesh Kanojia, Chi-Kiu Lo, Vilém Zouhar, Frederic Blain, Chrysoula Zerva, Eleftherios Avramidis, Sourabh Deoghare, Archchana Sindhujan, Jiayi Wang, David Ifeoluwa Adelani, Brian Thompson, Tom Kocmi, Markus Freitag, and Daniel Deutsch. 2025. Findings of the WMT25 shared task on automated translation evaluation systems: Linguistic diversity is challenging and references still help. In *Proceedings of the Tenth Conference on Machine Translation*, pages 414–461.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. Calibrating llm-based evaluator. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)*, pages 2638–2656.

Qingyu Lu, Liang Ding, Kanjian Zhang, Jinxia Zhang, and Dacheng Tao. 2025. Mqm-ape: Toward high-quality error annotation predictors with automatic post-editing in llm translation evaluators. In *Proceedings of the 31st international conference on computational linguistics*, pages 5570–5587.

Marco Lui and Timothy Baldwin. 2012. `langid.py`: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Stefano Perrella, Eric Morales Agostinho, and Hugo Zaragoza. 2026. Span-level machine translation meta-evaluation. *arXiv preprint arXiv:2603.19921*.

Brandon T Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *arXiv preprint arXiv:2307.09702*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Prompt

The system message is assembled from the blocks below: the shared annotation rules, the calibration block for the pair being annotated (if any), and the schema instruction. The eight demonstrations and the segment to annotate follow as user and assistant turns (Figure 2b); the demonstrations are omitted here for space.

A.1 Shared annotation rules

```
DECISION STANDARD
- Report errors that are supported by sufficient linguistic evidence from the source, target, and
reference.
- Do not require absolute certainty. Report a candidate when the target is more plausibly erroneous
than an acceptable translation.
- Do not annotate a target span merely because another wording would also have been acceptable or
sounds more natural -- omit anything that is a valid paraphrase, alternative translation, or harmless
stylistic variation.
- Balance completeness and restraint: do not invent errors, and do not artificially cap how many you
report in either direction. Report every independently supported error regardless of its severity, but
do not assume a segment must or must not contain one.
- Judge every candidate independently from the actual source and target. A segment may validly contain
zero, one, or multiple errors.
- Return a no-error response only when no candidate is sufficiently supported after comparing the
complete source and target.

WHAT COUNTS AS AN ERROR
- A major error materially changes, contradicts, omits, or seriously obscures important meaning.
Examples include a clear mistranslation, unsupported addition, important omission represented by a
literal [MISSING] token, contradiction, wrong named entity, number, date, unit, relation, polarity,
modality, or terminology.
- A minor error is a clear localized defect whose impact is limited while the surrounding core message
remains recoverable. It may concern grammar, agreement, spelling, punctuation, formatting, fluency,
terminology, or localized wording.
- First decide whether a span is erroneous. Severity is a separate judgment about its impact.
- Do not omit a supported error merely because it is minor. If it is clearly erroneous, include it and
assign the appropriate severity.
- Do not label an uncertain candidate as minor merely to avoid making a firm decision. If the existence
 of the error itself is too uncertain, omit it.
- Do not annotate the source text or reference translation. Only target substrings may be returned.

REFERENCE USE
- Treat the source text as the primary authority for meaning.
- Use the reference translation as auxiliary evidence, not as the only correct translation.
- The reference may help identify omissions, additions, mistranslations, terminology, relations, or
natural target-language realizations.
- A target span must not be marked as an error solely because its wording, syntax, or lexical choice
differs from the reference -- a lexical difference from the reference alone is insufficient evidence.
- If the target accurately expresses the source through a valid alternative translation, do not
annotate it.
- If the source and reference appear to conflict, prioritize the source and avoid speculative
annotation.

SPAN SELECTION POLICY
- Annotate only text that already appears in the target translation.
- Each error span must be one exact continuous substring copied from the target. Preserve every
character exactly.
- Select the smallest complete lexical, grammatical, or semantic unit that clearly represents the error
, not the smallest possible character sequence.
- Prefer one complete word or phrase over a character fragment, subword, stem, affix, or inflectional
ending. A smaller span is allowed only when it independently and completely represents the error.
- Keep a necessary function word, inflection, punctuation mark, or adjacent element when excluding it
would make the span incomplete or misleading.
- Do not include surrounding correct text or select an entire sentence when a smaller complete span
identifies the error.
- Do not split one semantic error into artificial fragments. Do not output duplicate, overlapping, or
contained alternatives for the same error.
- Distinct errors may be reported separately only when each is independently justified. Do not merge
independent errors merely because they are adjacent.
- A missing concept may be annotated only when the target literally contains [MISSING]. Never invent it
 or use an empty span.

OUTPUT DISCIPLINE
- Return exactly one valid JSON object and nothing else.
- Do not output Markdown, code fences, prose, explanations, corrections, rationales, categories,
quality scores, character indices, reasoning, or an analysis trace.
- Use exactly the required field names and no additional fields.
- Severity must be exactly "major" or "minor". Preserve error order by first appearance in the target.
```

A.2 Schema instruction

```
Return only a valid JSON object matching this schema:
{"errors": [{"text": "exact substring from the translation", "severity": "minor" or "major"}], "
omission": null}
Copy each span text exactly from the translation. Use an empty errors array when there are no errors.
No markdown, no comments, no offsets.
If important source content is entirely missing from the translation (an omission, with nothing to
quote in the translation), set the top-level "omission" field to "minor" or "major" for the whole
segment; otherwise set it to null. omission is a single value for the segment, never an entry inside
the errors array.
The response must start with '{' and end with '}'.
```

A.3 Pair-specific calibration blocks

```
[en-ar]
PAIR-SPECIFIC NOTE (Egyptian Arabic): scan carefully for register or dialect mismatches -- if the
target uses Modern Standard Arabic or a different dialect where the required Egyptian Arabic form was
expected, this is a real error even if the translation otherwise reads fluently. Most defects found are
 minor; reserve major for genuine dialect mismatches or mistranslations that change meaning.

[en-et]
PAIR-SPECIFIC NOTE (Estonian): most defects in this language pair are minor grammatical, agreement, or
fluency issues in Estonian morphology. Reserve major strictly for cases where the meaning is actually
changed, contradicted, or lost.

[en-is]
PAIR-SPECIFIC NOTE (Icelandic): most defects in this language pair are minor grammatical or fluency
issues. Reserve major strictly for cases where the meaning is actually changed, contradicted, or lost.

[en-id]
PAIR-SPECIFIC CALIBRATION (Indonesian): this language pair is consistently under-flagged. Most segments
 in this pair do contain at least one issue, and a fluent-looking translation typically still has one
to three of them (grammar, affixation, agreement, word-choice, register, omitted nuance, or slightly
shifted meaning); an entirely error-free translation is uncommon here. Treat a no-error answer as the
exception that needs justifying: before returning one, re-verify the target against the source and the
reference phrase by phrase (names, numbers, polarity, modality included), and only return no-error when
 that full check still finds nothing. At the same time, report each independent issue only once, do not
 split one problem into fragments, and do not invent errors to fill a quota -- a padded list is worse
than a short one. When you mark an error, cover the complete problematic phrase, not just the single
most suspicious word.

[cs-vi]
PAIR-SPECIFIC CALIBRATION (Vietnamese): this language pair is consistently under-flagged -- Czech-to-
Vietnamese is a demanding direction, and translations that read fluently at a glance still typically
contain one to three subtle issues (grammar, classifier or measure-word choice, word order, register,
omitted particles, or slightly shifted meaning). Before returning a no-error response, re-verify the
target against the source and the reference phrase by phrase (names, numbers, polarity, modality
included); only return no-error when that full check still finds nothing. Report each independent issue
 only once and do not invent errors to fill a quota. When you mark an error, cover the complete
problematic phrase, not just the single most suspicious word.

[cs-de]
PAIR-SPECIFIC CALIBRATION (German): this language pair is under-flagged -- translations that read
fluently at a glance still typically contain one to three subtle issues (case or gender agreement,
article choice, compound formation, word order, register, or slightly shifted meaning). Before
returning a no-error response, re-verify the target against the source and the reference phrase by
phrase (names, numbers, polarity, modality included); only return no-error when that full check still
finds nothing. Report each independent issue only once and do not invent errors to fill a quota. When
you mark an error, cover the complete problematic phrase, not just the single most suspicious word.

[zh-ja]
PAIR-SPECIFIC CALIBRATION (Japanese): this language pair is under-flagged -- a translation that reads
naturally at a glance still typically contains one to three subtle issues (particle or conjugation
choice, honorific or register level, word order, script choice, omitted nuance, or slightly shifted
meaning). Before returning a no-error response, re-verify the target against the source and the
reference phrase by phrase (names, numbers, polarity, modality included); only return no-error when
that full check still finds nothing. Report each independent issue only once and do not invent errors
to fill a quota. When you mark an error, cover the complete problematic phrase, not just the single
most suspicious word.
```