

STRA-MT: Deterministic Terminology Control for Document-Level Machine Translation

Vania Velda*
STRA, South Korea
vaniavelda88@gmail.com

Yigyu Hwang
Capable, South Korea
lethe.hwang@gmail.com

Yongho Kim
STRA, South Korea
yongho@stra.ai

Abstract

STRA-MT is a translation pipeline with deterministic terminology control built around small open weight language models. The pipeline (1) reuses exact repetitions and guides near-duplicates using translations from earlier occurrences, (2) maintains a growing terminology ledger initialized from either the provided glossary or a termbase mined from sample bitext, and (3) verifies every required term and, when one is missing, regenerates the paragraph once with the missing terms named. On the WMT25 zh-Hant→en development sets, STRA-MT improves mean terminology success over glossary prompting from 87.3 to 92.4 under the official WMT25 evaluation, while maintaining competitive BLEU, chrF++, and COMET scores. Official WMT26 test sets shows that term delivery on en→pl comes close to the reference ceiling while the translation quality depends on the backbones.

1 Introduction

Recent developments in large language models (LLMs) have significantly advanced the state of the art in machine translation (MT). In terms of fluency and naturalness, LLMs have rapidly narrowed the gap with dedicated MT systems and are increasingly used for general purpose translation. However, in specialist domains like medicine or finance, accurate and consistent terminology remains an open problem. WMT25 terminology task (Semenov et al., 2025) found that provided terminology helps, especially for systems that already translate well.

Previous work controls terminology through mechanisms such as constrained decoding (Hokamp and Liu, 2017; Post and Vilar, 2018) or by providing required terms as part of the input to instruction-following LLMs (Moslem et al., 2023). Recent LLM translation agents extend this

approach with document-level memory maintained in the prompt (Wang et al., 2025), building on mechanisms that page memory through the prompt (Packer et al., 2023). In these approaches, terminology and document context are provided to the model as part of translation.

STRA-MT keeps translation choices consistent through explicit steps in the translation process. It uses at most two small open weight models for each translation direction. Before translation, an offline planning step finds exact repeated paragraphs and reuses their earlier translations. For near-duplicate paragraphs, it provides the earlier translation as a guide while generating a new translation. During translation, a term ledger stores the provided glossary and the terms chosen by the system in earlier paragraphs. After each paragraph is generated, a verifier checks whether every required term appears in the output. If a required term is missing, the system translates the paragraph one more time with the missing terms provided in the prompt. A cached classifier checks whether a term can appear in an inflected form. Short notes record recurring entities and are included in later prompts while they remain relevant. Final checks look for untranslated source text, repeated or looping text, and malformed headings.

2 System Architecture and Pipeline

2.1 Overview

The WMT26 Terminology Translation Task consists of two tracks. In Track 1, every document comes with a terminology dictionary in two versions. The *proper* version lists the real domain terminology, and the *random* version is a control set of ordinary words with their reference translations. Each document is translated three times, once with no terminology, once given the proper version, and once given the random version. In Track 2 only sample bitext per do-

* Corresponding author.

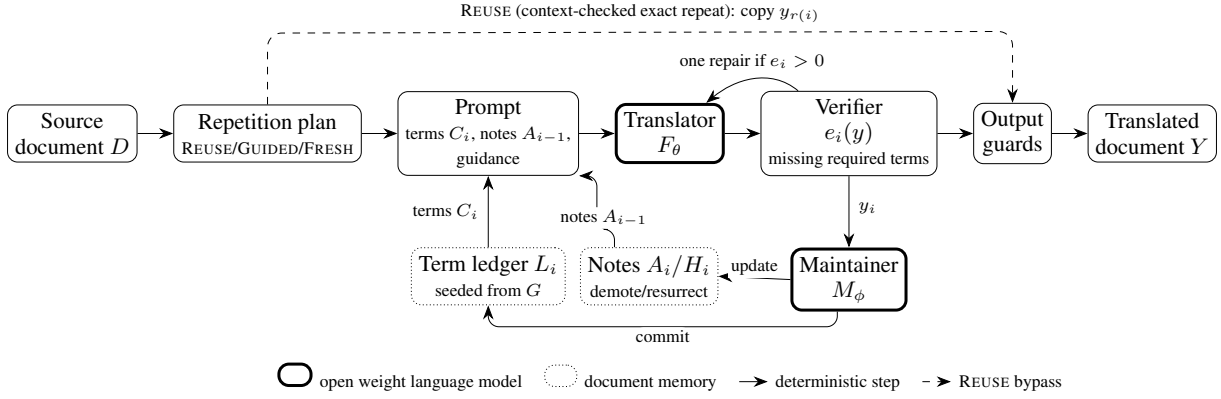


Figure 1: The STRA-MT pipeline. The plan routes each paragraph, translation is conditioned on the ledger and notes with at most one repair, and the maintainer updates both memories.

main is supplied, and documents are translated without terminology and given the bitext. Both tracks cover es→eu (Spanish→Basque), en→pl (English→Polish). Track 2 additionally covers zh-Hant→en (Traditional Chinese→English). We submitted to every cell, that is every direction and domain, of both tracks.

STRA-MT is built around a translator and a maintainer model. Technical documents are usually repetitive. We therefore resolve repetition with a deterministic plan built before any translation begins (§2.3). Translation then proceeds paragraph by paragraph under a growing term ledger with a verify-and-repair loop (§2.4). While it runs, the maintainer writes short discourse notes that are paged in and out of the prompt (§2.5). Finished outputs pass three guards before staging (§2.6). The key prompts are given in Appendix A.

The components play different roles across the two tracks. The ledger, together with its verification and repair mechanisms, directly addresses the task. In Track 1, it is seeded from the provided dictionary, whereas in Track 2, it is seeded from a termbase mined from the sample bitext. For en→pl, the inflection check further allows the system to accept declined forms. The repetition plan, discourse notes, and output guards are intended more generally for long technical documents. On the development sets, these components do not affect term success (§3), while on this year’s blind documents, the repetition plan is triggered only rarely (Appendix B.2). We nevertheless retain them as inexpensive safeguards for repetitive or entity-dense documents.

Figure 1 summarizes the pipeline. Throughout, the source document is $D = (x_1, \dots, x_N)$ with

Direction	Backbone	BLEU	chrF++
zh-Hant→en	Gemma-4-E2B	25.8	53.7
	NLLB-1.3B	25.9	52.4
	TranslateGemma-4b	21.9	51.2
es→eu	Gemma-4-E2B	2.0	26.1
	NLLB-1.3B	12.8	46.9
	TranslateGemma-4b	4.2	35.2
en→pl	Gemma-4-E2B	20.7	48.2
	NLLB-1.3B	21.1	48.6
	TranslateGemma-4b	16.9	45.9

Table 1: Backbone screening on FLORES-200 devtest. Bold marks the selected backbone.

translation y_i for paragraph x_i , and $r(i)$ is the index of the earlier paragraph most similar to x_i . G is the provided glossary or the mined termbase, L_i the term ledger after paragraph i , C_i the set of ledger terms enforced on paragraph i , $e_i(y)$ the number of those terms missing from a candidate output y , A_i and H_i the active and archived discourse notes, F_θ the backbone translator, and M_ϕ the maintainer model. We call zh-Hant→en and en→pl the *instruct directions*. Their translator is an instruction-following model that reads the ledger and the notes from the prompt. On es→eu the translator is a specialist model, and the maintainer applies the ledger by post-editing (§2.2).

2.2 Backbones

We select one base translator per direction by evaluating each candidate on the FLORES-200 devtest (NLLB Team, 2022) with BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017) as the evaluation metrics. We choose instruction-following Gemma-4-E2B (Gemma Team, 2026), the multilingual specialist NLLB-1.3B (NLLB Team, 2022),

Direction	Translator	Maintainer	Ledger seed
zh-Hant→en	E2B	shared E2B	mined termbase
en→pl	E2B	E4B	dictionary / mined
es→eu	NLLB + E4B	E4B	dictionary / mined

Table 2: Submitted configuration per direction. On es→eu the E4B maintainer also post-edits the NLLB draft.

and TranslateGemma-4b (Finkelstein et al., 2026) as our candidates. We select Gemma-4-E2B for zh-Hant→en and en→pl (Table 1). It is competitive and follows an in-prompt glossary. On es→eu, Gemma-4-E2B collapses into pseudo-Basque with Spanish intrusions (for FLORES “...tenemos ratones de cuatro meses...” it produces “*Aktualean, urtekatarratarren ratonek dituztegi*” where NLLB gives “*lau hilabeteko saguak ditugu*”), therefore we use NLLB-1.3B as the translator. Because NLLB cannot utilize any prompt, the maintainer takes a second role as the *corrector*. After NLLB drafts a paragraph, the same loaded Gemma-4-E4B instance post-edits the draft so that the required terms appear. The edit is accepted only if it loses no term and does not drift toward the source language (Appendix A.3). E2B and E4B are the two smallest Gemma-4 instruction-tuned variants, with about 2B and 4B effective parameters. On zh-Hant→en the E2B translator serves as its own maintainer, which delivers more terms than a separate E4B (§3). On en→pl and es→eu we use a separate 4-bit Gemma-4-E4B maintainer. Table 2 lists the resulting configuration per direction. The pipeline is identical across directions. The plan thresholds, ledger policy, verification, note paging, guards, and prompts are shared. Only the models in the translator and maintainer slots differ.

2.3 Offline repetition plan

Before any translation, a deterministic pass scans the document for repetition. Paragraphs are compared after lowercasing and collapsing whitespace. For each paragraph x_i the pass finds the most similar earlier paragraph $x_{r(i)}$ and assigns one of three modes. Similarity is the `diff lib` Sequence-Matcher ratio over characters, $2M/T$, where M is the number of matched characters and T is the combined length. Pairs whose lengths differ by more than a factor of two are skipped. The pass runs on the CPU and makes no model call.

REUSE applies when the paragraph repeats an earlier one exactly and the paragraph immediately

before it also matches. We call this the *context check*. The earlier translation $y_{r(i)}$ is then copied with no model call. If the preceding paragraphs differ, the exact repeat is not copied and is handled as a near-duplicate instead.

GUIDED applies when the two paragraphs are similar but not identical, with a similarity ratio of at least 0.7. The paragraph is translated by the model, but the prompt shows the earlier paragraph and its translation as an example, together with an instruction derived from the difference (Appendix A.3). When only values differ, such as numbers or dates, the instruction is to keep the example’s wording and substitute the new values. When the wording itself differs, the instruction is to translate the changed part faithfully. This guidance requires a translator that follows prompts, so it applies on the instruct directions only. On es→eu, where NLLB translates the bare segment, a near-duplicate is translated as fresh and only the reuse of exact repeats operates.

FRESH applies to everything else, and the paragraph is translated normally.

2.4 Term ledger with verify and repair

Ledger The system maintains a growing term ledger as external memory for terminology. The ledger can be initialized from a provided glossary or from a termbase mined from sample bitext. Unlike the soft proper-noun records used by Delta (Wang et al., 2025), ledger entries are treated as fixed translation choices. After each paragraph, the translator identifies the proper nouns and fixed technical terms it used, and these source–target term pairs are added to the ledger under a first-commitment policy. Glossary entries always take priority, existing ledger entries remain unchanged, and only previously unseen terms receive a new translation.

When a ledger term appears again, its committed translation is provided to the translator as the required rendering. Without a glossary, the system’s first translation of a recurring term therefore becomes the translation used for later occurrences. The ledger does not assess whether a committed translation is correct. This means an incorrect entry is propagated just as consistently as a correct one.

Termbase mining In Track 2, we build a small termbase from the provided sample bitext using SimAlign (Jalili Sabet et al., 2020). We keep a source-target pair when it is aligned at least twice and the most frequent target accounts for at least 60

percent of that source term’s alignments. We then remove pairs where two nearby content words align to the same target token, as well as pairs containing a digit or function word.

Inflection-aware enforcement Enforcement applies to all terms in the ledger, whether they were seeded from a termbase or added by the system. For each paragraph x_i , the system first identifies the ledger terms that occur in that paragraph and stores them in an enforcement set C_i . Recognition is based on surface matching. Both the paragraph and the ledger keys are lowercased before matching. Keys containing Latin letters must occur as whole tokens, while keys in unspaced scripts, such as Chinese, are matched as substrings. Only the matched terms are included in the prompt, and the complete glossary is never provided. The system does not perform sense disambiguation (see Limitations).

The system then applies two cached checks provided by the maintainer M_ϕ . The proper-name check removes a term if its occurrence is part of a larger company, brand, or organization name. The term is still enforced when it occurs independently in another paragraph. The inflection check determines whether a term must be used in its committed form or can be inflected (Bergmanis and Pinnis, 2021). Fixed-form terms must be reproduced exactly. For example, the acronym *ERTMS* must appear verbatim. Inflecting terms can appear in different grammatical forms. For Polish *karcynogeneza*, for example, *karcynogenezy* is accepted because each word in the committed rendering matches an output token by stemmed prefix (Appendix B.4).

Both checks use only local models. The final set of requirements is then added to the translation prompt as instruct directions. For $es \rightarrow eu$, the E4B corrector also applies these requirements when post-editing the NLLB draft.

Verify and repair After each paragraph, a deterministic verifier counts the required terms missing from the output, $e_i(y)$. If any is missing, the system regenerates the paragraph once with the missing terms named (Appendix A.2) and keeps whichever of draft and repair misses fewer, with ties resolved for the repair. Regeneration is thus bounded to one extra model call and can never increase the number of violations.

2.5 Paged discourse notes

Notes While translating, the maintainer writes short notes about the entities in the current doc-

ument, recording who did what (Appendix A.3). Notes about recently mentioned entities are kept in the prompt so that later references stay coherent. Nothing is retrieved from outside the document, and the maintainer must use the same renderings the ledger enforces in every note and alias.

Demote and resurrect Because the context window is limited, note paging is deterministic. A note unmentioned for 8 paragraphs is demoted to the archive H_i , and at most 12 notes stay active in A_i . Before each paragraph is translated, every archived note is tested against the source paragraph. The test is the same surface match used for ledger terms (§2.4). It is applied to the note’s entity name, its object, and every alias the maintainer recorded, and a note with any hit is resurrected into A_i . It is tagged with the number of paragraphs since it was last seen, so the translator can tell old context from current context. Demotion archives rather than deletes, so a paged-out note can always return. The match is exact at the string level. It costs no model call and never resurrects a note for an unrelated entity. It misses a mention that appears only as an inflected form, a pronoun, or a paraphrase. The maintainer must also have recorded the alias for it to count. We did not measure the recall of this trigger against a gold standard (see Limitations).

2.6 Output guards

We add postprocessing guards over every finished output before it is staged as a submission file. The first two guards check every document and fire only on the failure they target. The third is switched on per cell. An LLM-based repair is accepted only when it improves the metric that flagged it. The guards are not part of the development harness, so all development results are measured without them. Appendix B.2 reports how often each fired on the blind test. The *target-language identity guard* flags output still in the source language via per-direction function-word and script rules and retranslates it with the direction’s repair prompt. The *repetition guard* flags a paragraph when any word trigram repeats four or more times and retranslates it, falling back to a deterministic collapse that keeps the first occurrence if the backbone keeps looping. The *heading guard* recases mangled all-caps headings through a minimal-edit pass. It runs only on the $en \rightarrow pl$ medical documents, a switch set by inspecting the released sources.

Year	Bare translator					+ glossary					STRA-MT				
	BLEU	chrF++	COMET	TS	TS ₂₆	BLEU	chrF++	COMET	TS	TS ₂₆	BLEU	chrF++	COMET	TS	TS ₂₆
2016	28.4	66.2	0.849	41.4	44.9	32.9	69.0	0.859	87.2	73.4	32.8	69.1	0.861	91.8	74.8
2018	27.5	65.6	0.835	44.4	39.7	34.8	69.8	0.849	88.3	69.9	35.0	70.0	0.856	92.6	71.7
2020	26.9	65.5	0.837	43.5	38.2	32.4	68.7	0.853	85.6	66.4	32.1	68.8	0.855	91.2	69.4
2022	26.8	65.7	0.848	39.9	39.3	32.7	69.6	0.865	86.6	69.5	32.8	69.7	0.865	92.7	73.0
2024	28.6	66.8	0.848	47.2	43.1	35.7	70.8	0.865	88.7	70.8	35.8	71.0	0.866	93.6	73.9
Avg	27.6	66.0	0.843	43.3	41.0	33.7	69.6	0.858	87.3	70.0	33.7	69.7	0.861	92.4	72.5

Table 3: zh-Hant→en WMT25 dev years. TS is WMT25 term success, TS₂₆ the WMT26 measure. Both glossary-fed configurations run the proper-name check, answered by E4B in the glossary column and by E2B in the STRA-MT column (see the proper-name check paragraph). Bold marks the best Avg.

3 Results

Evaluation setup WMT26 provides no development references for es→eu or en→pl, so reference-based validation is possible only for zh-Hant→en. We use the five zh-Hant→en years of the WMT25 Terminology Task development data (finance). We compare against the bare Gemma-4-E2B translator and the same translator with the glossary in the prompt (Appendix A.1). STRA-MT uses Gemma-4-E2B as translator and maintainer. The metrics are BLEU (Papineni et al., 2002), chrF++ (Popović, 2017), COMET (Rei et al., 2020) with the wmt22-comet-da checkpoint (Rei et al., 2022), and term success (TS), the share of required dictionary terms whose allowed rendering appears in the output. For zh-Hant→en we also report TS₂₆, the WMT26 lemmatised exclusive term success from the organisers’ scorer¹ (Charkiewicz et al., 2026). It is stricter than TS, because it requires a separate output span for every source occurrence of a term, so its ceiling lies below 100. The development data has no occurrence annotations, so we annotate each source paragraph with the dictionary entries it contains that pass the WMT25 reference filter, one per occurrence. BLEU and chrF++ score each document as one segment. COMET is averaged over aligned paragraphs. BLEU, chrF++, and TS use the official WMT25 implementation².

Findings Table 3 shows the zh-Hant→en results. Showing the model the glossary brings the largest improvement, from 43.3 to 87.3 term success, in line with the WMT25 finding that instruction-following models exploit provided terminology (Seimenov et al., 2025). STRA-MT improves term suc-

Configuration	BLEU	chrF++	COMET	TS	TS ₂₆
Bare translator	27.6	66.0	0.843	43.3	41.0
+ glossary in prompt	33.7	69.6	0.858	87.3	70.0
Ledger + verify/repair	33.6	69.7	0.857	92.4	72.6
+ plan (no notes)	33.7	69.7	0.861	92.4	72.6
+ notes (no plan)	33.7	69.7	0.857	92.4	72.6
Full pipeline (STRA-MT)	33.7	69.7	0.861	92.4	72.5
Full, mined termbase	26.2	65.1	0.845	40.6	38.8
Full, same-corpus mined	26.9	66.1	0.848	42.0	42.9

Table 4: Component ablation and termbase seeding, zh-Hant→en five-year means, same setting as Table 3. Each “+” row adds one component to the ledger row. Bold marks the best in the upper block.

cess by a further 5.1 points to 92.4 and is the best of the three configurations in every year. The proper-name check paragraph below gives the like-for-like comparison. The cost is about one BLEU point, because a repaired paragraph is regenerated whole and kept when it delivers more terms. Translation quality is inherited from the backbone. Qualitative examples are in Appendix B.

Ablation The upper block of Table 4 adds one component at a time. *Glossary in prompt* lists the terms found in the paragraph as mandatory, without verification. *Ledger + verify/repair* adds the ledger of §2.4, together with its two checks and single repair. The + *plan* and + *notes* variants do not further improve term success. Neither changes BLEU, chrF++, or COMET. The output guards are not part of these experiments.

Each development set contains only 5 to 17 exact repeats among 427 to 641 paragraphs. The context check rejects all of them, so REUSE never fires here. In 2016, 44 paragraphs receive near-duplicate guidance.

Cost Table 5 gives the time per step. Translation takes under a second per paragraph, and the repair

¹github.com/wmt-conference/wmt26-terminology

²github.com/wmt-conference/wmt25-terminology

Step	Calls	Seconds	s/paragraph
Translation	427	403	0.94
Repair	126	215	0.50
Name and inflection checks	cached	71	0.17
Ledger extraction	427	488	1.14
Note maintenance	427	962	2.25
Full pipeline		2,139	5.01
Without notes		1,177	2.76

Table 5: Time per step of the full pipeline, 2016 zh-Hant→en (427 paragraphs), Gemma-4-E2B on one RTX 4090.

adds half a second because it fires on 30 percent of paragraphs. The two maintainer calls dominate. Ledger extraction costs more than the translation, and note maintenance more than twice as much. The terminology steps together take 2.8 seconds per paragraph and give the whole gain, while the notes add 2.3 seconds for no measurable gain.

Random glossary condition The pipeline delivers a randomly selected glossary as faithfully as the proper one. On the 2016 year, term success on the random dictionary rises from 91.6 under prompting to 93.5 under enforcement. Term success on the proper dictionary falls from 41.4 to 36.9 at the same time, and BLEU by 6.3 against the bare translator.

Seeding from a mined termbase The WMT26 zh-Hant→en submission seeds the ledger from a mined termbase (§2.4). The lower block of Table 4 scores this configuration against the official dictionaries. Term success falls below the bare translator (40.6 vs. 43.3). Enforcement works, but the mined inventory covers less than 1% of the dictionary terms. The miner extracts word-level pairs, while the dictionaries hold multiword expressions and names, such as 金立群→“liqun” where the dictionary requires “President Jin Liqun”. Against its own inventory the same pipeline reaches 95.3 term success, against 86.8 for the bare translator, and same-corpus termbases reach 97.0 against 80.9.

Building the termbase with the maintainer The mined termbase can be replaced by a termbase built by the maintainer from the same sample bitext. The maintainer reads each sample sentence pair and identifies the term pairs it contains. A pair is kept only if both sides occur verbatim in that sentence pair, and disagreements across batches are resolved by majority vote. Terms that occur only in the test documents are translated afterwards, using the verified pairs as references. On mechanical

Track 2 cell	E2B		31B	
	none	mined*	none	built
mech. eng. en→pl	25.1	24.9	40.0	88.4
medical en→pl	53.2	52.6	63.2	87.0
machine-tool es→eu	25.6	25.6	28.5	67.4
railways es→eu	51.8	55.0	58.5	70.8
finance zh-Hant→en	68.6	64.9	80.4	80.6

Table 6: Term success (TS₂₆) on the Track 2 test cells with no termbase, the mined termbase, and the one the maintainer builds. Within each block only the termbase changes. *The submitted files.

engineering, this procedure produces 761 entries, compared with 108 from the aligner.

Table 6 shows the results on the official test sets. With the backbone held fixed, the model built termbase improves term success by 48.4 points on mechanical engineering, 38.9 on machine tools, 23.8 on medical, and 12.3 on railways. The mined termbase we submitted does not improve the same system as it scores at or below the no-termbase setting on four of the five cells.

The difference is important because the ledger commits each entry once and then requires the same rendering at every later occurrence. A misaligned pair therefore does not cause just one local error, since it can propagate the wrong rendering throughout the document. A termbase containing enough such errors can consequently perform worse than having no termbase at all.

The proper-name check The proper-name check utilizes the maintainer in deciding whether a matched term is part of a larger product or company name (§2.4). If it is, the term is removed from the prompt and is not verified. This prevents the system from forcing a translation inside a name that should remain unchanged.

Chinese is written without spaces, so a glossary term can match a substring even when it is part of a larger name, and the check is applied to nearly every term. How often it answers yes therefore changes how many terms the system must deliver, which matters for Tables 3 and 4. The glossary row was run with E4B answering the check, while the STRA-MT rows use E2B, and E4B identifies more terms as part of names. Thus, part of the 5.1-point difference between these rows comes from the check rather than from the pipeline. Using the same E2B check for both settings gives 91.6 for prompting and 92.4 for the pipeline. Turning the check off, as we did for the submitted zh-Hant→en

Track	Direction	chrF++	TS ₂₆	Ceiling	COMET
1	en→pl	72.9	91.7	93.6	87.0
1	es→eu	65.8	86.3	99.1	81.3
2	en→pl	67.3	38.7	94.9	84.0
2	es→eu	64.8	40.3	99.1	78.6
2	zh-Hant→en	65.5	64.9	86.3	80.2

Table 7: Official WMT26 scores of STRA-MT per direction, Track 1 in proper mode and Track 2 in sample mode (Charkiewicz et al., 2026). Ceiling is the reference scored against itself.

system, gives 95.3 for prompting and 96.4 for the pipeline, with BLEU at 36.6 and 35.6 and COMET equal at 0.866. The pipeline therefore adds about one term-success point under either setting.

We keep the check for en→pl and es→eu, where spaces make the matches unambiguous and the check fires mainly when a term occurs inside a larger name.

Official results Table 7 gives the official scores from the organisers. In Track 1, STRA-MT achieves 91.7 TS₂₆ on en→pl, compared with a reference ceiling of 93.6. On es→eu, it achieves 86.3 against a ceiling of 99.1. Thus, term success is high on both directions, although the gap to the ceiling is larger for es→eu. Across the two Track 1 directions, the average TS₂₆ is 89.0. Track 2 was not the main focus of this system. We submitted a termbase mined from the sample bitext. On the three Track 2 directions, STRA-MT achieves TS₂₆ of 38.7 on en→pl, 40.3 on es→eu, and 64.9 on zh-Hant→en. The corresponding chrF++ scores are 67.3, 64.8, and 65.5, and the COMET scores are 84.0, 78.6, and 80.2.

Ablation on the test set After the gold data were released, we reran the three main configurations on all four Track 1 test cells and evaluated them using the organisers’ scoring code.

For en→pl, the ledger provides a small improvement over prompting alone. It increases term success by 0.2 and 1.7 points, while improving chrF++ by only 0.1 to 0.2 points. Adding the notes removes 0.3 and 1.3 of these term-success points. The submitted files perform better than these reruns, with term-success scores that are 1.1 and 2.0 points higher. These files use an earlier system version followed by output guards.

The results for es→eu show a different pattern. Here, a specialist translator first produces the translation, and the maintainer then post-edits it. The

full pipeline improves term success over prompting by 0.9 points on energy and 1.1 points on automation. The ledger-only configuration gives intermediate results but reduces chrF++ by about one point. The components are therefore most useful when the corrector is responsible for applying the required terminology. When the translator can directly use the glossary, the components provide little benefit and may even reduce performance.

We also evaluated a larger 31B backbone using the ledger alone (Appendix B.3) and the model-built termbase (Table 6). Using the organisers’ released scorer and gold annotations,³ this system reaches 93.1 lemmatised term success and 72.0 chrF++ on Track 1, against 89.0 and 69.3 for the submitted files.

4 Conclusion

This paper presents our submission to both tracks of the WMT26 Terminology Translation Task. Our approach uses deterministic control procedures around small language models rather than relying on the model’s memory or its ability to maintain terminology consistency on its own. The main idea is that terminology and repeated content can be checked directly. String matching can verify whether a required term appears in the output, while a fixed procedure can reuse previous translations for repeated or related content. We therefore enforce these properties with explicit rules and leave the language model to focus on translation.

Our results show that this approach improves terminology consistency while keeping translation quality close to that of the underlying backbone model. The control procedure provides three main benefits. First, it verifies that required terms are actually delivered. Second, it ensures that the first committed translation of a term is used consistently throughout the document. Third, it enforces an arbitrary or mined inventory as faithfully as a reference one.

For Track 2, building the termbase with the maintainer recovers most of the performance gap caused by using a mined inventory (§3). In contrast, checking terminology at the occurrence level does not improve performance. On the en→pl test cells, it reduces term success by about one point. The main reason is that occurrence-level checking requires regenerating individual sentences, which gives the

³github.com/wmt-conference/wmt26-terminology

model less context than regenerating the full paragraph.

Overall, these results suggest that simple deterministic controls can provide reliable terminology enforcement without requiring a larger or more capable translation model.

Limitations

Translation quality still depends on the backbone model. Errors that are not detected by the guards remain in the output. The heading guard is also enabled separately for each domain by inspecting the released source texts. This manual step does not transfer automatically to new domains.

The system can only enforce the inventory it receives. Therefore, an incorrect first commitment can be repeated throughout the document. For example, if 金管局 is first translated as “Monetary Authority of Taiwan”, this translation is reused in later occurrences. Term recognition is based only on surface matching and assumes that each source string is the term and has a single meaning. Both assumptions fail for a form such as English *control*. It is a verb as well as a noun, so the system enforces the glossary rendering on occurrences that are not the term at all. As a noun it names different concepts, so one committed rendering is applied to all of them. The proper-name check removes only the first kind of false match, and only when the term sits inside a larger name. The system performs no part-of-speech filtering and no word-sense disambiguation, so a glossary that lists sense-specific entries for one string cannot be honoured.

The same surface matching procedure is used to restore notes, but we did not measure its recall. A note is not restored when the relevant content appears only as an inflected form, a pronoun, or a paraphrase.

Reference-based development evaluation is available only for zh-Hant→en because WMT26 does not provide development references for es→eu or en→pl. We therefore validated these two directions only with the official test-set scores, which became available after the system was frozen.

All submitted models have at most 4B effective parameters. These smaller models may have weaker capabilities than models with 20B or more parameters and may perform worse on lower-resource languages. Our screening of es→eu backbones shows this effect (§2.2). Appendix B.3 fur-

ther examines the behaviour of the pipeline with larger models from the same Gemma-4 family.

All reported results are single greedy-decoded runs. The variation across development years is the only variability estimate provided.

References

- Toms Bergmanis and Mārcis Pinnis. 2021. [Facilitating terminology translation with target lemma annotations](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.
- Adrian Charkiewicz, Pinzhen Chen, Thierry Etchegoyhen, Harritxu Gete Ugarte, Kamil Guttman, Xu Huang, David Ponce, Artur Nowakowski, Frédéric Odermatt, Arturo Oncevay, Kirill Semenov, Dawei Zhu, and Vilém Zouhar. 2026. Findings of the WMT26 terminology translation task. In *To appear in Proceedings of the Eleventh Conference on Machine Translation*. Association for Computational Linguistics.
- Mara Finkelstein, Isaac Caswell, Tobias Domhan, Jan-Thorsten Peter, Juraj Juraska, Parker Riley, Daniel Deutsch, Geza Kovacs, Cole Dilanni, Colin Cherry, Eleftheria Briakou, Elizabeth Nielsen, Jiaming Luo, Kat Black, Ryan Mullins, Sweta Agrawal, Wenda Xu, Erin Kats, Stephane Jaskiewicz, Markus Freitag, and David Vilar. 2026. [TranslateGemma technical report](#). *Preprint*, arXiv:2601.09012.
- Gemma Team. 2026. [Gemma 4 technical report](#). *Preprint*, arXiv:2607.02770.
- Chris Hokamp and Qun Liu. 2017. [Lexically constrained decoding for sequence generation using grid beam search](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.
- Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. [SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.
- Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. [Adaptive machine translation with large language models](#). In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 227–237, Tampere, Finland.
- NLLB Team. 2022. [No language left behind: Scaling human-centered machine translation](#). *Preprint*, arXiv:2207.04672.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. **Memgpt: Towards llms as operating systems**. *Preprint*, arXiv:2310.08560.

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. 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.

Matt Post and David Vilar. 2018. **Fast lexically constrained decoding with dynamic beam allocation for neural machine translation**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

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

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.

Kirill Semenov, Xu Huang, Vilém Zouhar, Nathaniel Berger, Dawei Zhu, Arturo Oncevay, and Pinzhen Chen. 2025. **Findings of the WMT25 terminology translation task: Terminology is useful especially for good MTs**. In *Proceedings of the Tenth Conference on Machine Translation*, pages 554–576, Suzhou, China. Association for Computational Linguistics.

Yutong Wang, Jiali Zeng, Xuebo Liu, Derek F. Wong, Fandong Meng, Jie Zhou, and Min Zhang. 2025. **DelTA: An online document-level translation agent based on multi-level memory**. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

A Prompts

The base translation prompt, without its optional notes and fuzzy-match blocks, and the repair prompt are given in full. The other prompts are summarised.

A.1 Base translation prompt (instruct translator)

Translate the following {src} paragraph into {tgt}. This is a professional document; keep terminology precise and style consistent.

Use EXACTLY these {tgt} terms for the following {src} terms (mandatory terminology):
- "{source term}" -> "{committed term}"

{src} paragraph:
{paragraph}

Output ONLY the {tgt} translation, nothing else.

A.2 Repair prompt

Your previous translation:
{draft}

It is missing these mandatory terms:
"{term}", ...
Rewrite the {tgt} translation using ALL mandatory terms exactly as given.
Output only the translation.

A.3 Other prompts (summarized)

The *fuzzy-match block* for GUIDED paragraphs shows the earlier source and its translation and asks the model to reuse that phrasing, changing only what differs. The *es→eu corrector* prompt, answered by the maintainer, receives the source, NLLB’s draft, the mandatory terms, and the active notes, and rewrites the draft so that every mandatory term appears. It runs only when the draft misses a term or notes are active. The rewrite is accepted only if it misses no more terms than the draft, does not raise the overlap with the source, and copies no source span. The *note maintainer* receives the current notes, the term renderings, and the new source–target pair, and returns add, update, and delete operations. The *glossary classifier* labels each term fixed-form or inflecting. The *proper-name check* asks whether a matched span is part of a company, brand, or organisation name. The *ledger extraction* prompt asks which proper nouns and fixed technical terms the translation used. The *heading revision* and *target-language repair* prompts perform the guard repairs and return only the corrected span.

B Additional development results

B.1 Qualitative examples

Table 8 shows one delivery example, one consistency example, and one wrong commitment that the ledger repeats identically.

B.2 Blind-test run statistics

Table 9 shows how the deterministic components behaved on the blind test. The REUSE bypass was rarely used. The context check rejected almost all repeats proposed by the plan. As a result, only five paragraphs in the entire submission were copied without a model call. The en→pl documents contained no such repeats.

GUIDED prompting was used for 19 zh-Hant→en units and was not used for the other language pairs. Near-duplicate content cannot be handled with guided NLLB prompting. The ledger and repair loop were used for all language pairs. After the single repair step, between 4 and 33 percent of the required term slots were still unmet, with the highest rates in the Polish cells.

The output guards also made several corrections. For es→eu, they rewrote 75 of 1,544 automation paragraphs and 31 of 881 energy paragraphs that still contained source-language text. They accepted 6 of 25 repetition repairs and stopped 17 looping paragraphs. For en→pl, they recased 67 of 78 flagged medical headings. For zh-Hant→en, they rewrote two paragraphs that still contained Chinese.

B.3 Backbone scale

Table 10 compares the three main configurations with a larger 31B backbone on all four Track 1 test cells. On en→pl the larger model replaces the translator. On es→eu it replaces only the corrector, because NLLB cannot read a prompt.

Without a glossary the larger backbone is better on both metrics, gaining 1.9 to 5.9 chrF++ and 3.9 to 12.5 term-success points. Adding the glossary lifts term success sharply at either size, and at 31B it reaches 96.7, 94.7, 90.9 and 92.5 on the four cells. The full pipeline then scores 95.2, 92.9, 90.2 and 92.4, slightly below the glossary alone everywhere, at equal translation quality.

The pattern differs from the submitted backbone, where the full pipeline is ahead on three of the four cells. Deterministic enforcement therefore helps most when the backbone is small, and a model large enough to follow a glossary by itself has little

left for it to add. A 31B corrector also cannot share a 24 GB GPU with NLLB, which limits its use in our setup.

B.4 Implementation details

The translator loads in bfloat16 and the separate maintainer, where used, in 4-bit NF4, which together fit one 24 GB RTX 4090. The zh-Hant→en Track 2 documents arrive without line breaks, so each is split into sentences for translation and rejoined afterwards. Generation length is $\min(2048, \max(512, |\text{source}|/2))$ new tokens. Stem verification accepts a lowercased token whose prefix matches the term word minus up to three trailing characters, with a minimum stem of four characters, and every word of a multi-word term must match. Both constants were fixed once and not tuned. The inflection classifier is cached per glossary and the proper-name check per term-paragraph pair. BLEU and chrF++ use sacrebleu 2.6.0 (Post, 2018), with the zh tokenizer for Chinese targets and 13a otherwise, and chrF++ with character order 6 and word order 2. The official term-success function is ported verbatim and validated to produce identical scores. Documents are scored as single segments for BLEU and chrF++, and COMET is averaged over aligned paragraphs.

金管局 → dictionary “HKMA” (2016)	
Source	金管局公布，由2017年1月1日起香港適用的逆周期緩衝資本比率將上調至1.25%。
+ glossary	The Monetary Authority announced that the Countercyclical Capital Buffer applicable in Hong Kong from January 1, 2017, will be raised to 1.25%.
STRA-MT	The HKMA announced that the Countercyclical Capital Buffer applicable in Hong Kong from January 1, 2017, will be raised to 1.25%.
Reference	The HKMA announces that the Countercyclical Capital Buffer rate for Hong Kong will increase to 1.25% with effect from 1 January 2017.
人民幣 → “renminbi” (2016): three nearby paragraphs of one document. The term occurs ten times in the document. Prompting drifts to “RMB” twice, STRA-MT delivers “renminbi” every time.	
Source	# 離岸人民幣業務 圖1 人民幣客戶存款及存款證 圖2 人民幣融資活動
+ glossary	# Offshore Renminbi Business Figure 1 RMB Customer Deposits and Certificates of Deposit Figure 2 RMB Financing Activities
STRA-MT	# Offshore Renminbi Business Figure 1 renminbi customer deposits and deposit certificates Figure 2 renminbi financing activities
Reference	# Offshore renminbi business Chart 1 Renminbi customer deposits and certificates of deposit Chart 2 Renminbi financing activities
金管局 → committed as “Monetary Authority of Taiwan” (2018): a wrong choice, propagated. Two distant paragraphs of one document carry the same invented name.	
Source	金管局與波蘭金融監管局交換《諒解備忘錄》，促進雙方的金融科技合作。 金管局與巴西中央銀行簽署《合作協議》，促進兩間中央銀行在金融科技事項的合作。
STRA-MT	The Monetary Authority of Taiwan and the Polish Financial Supervision Authority exchanged a Memorandum of Understanding... The Monetary Authority of Taiwan and the central banking of Brazil have signed a Co-operation Agreement...
Reference	The HKMA and the Polish Financial Supervision Authority exchange an MoU... The HKMA and the Central Bank of Brazil sign a Co-operation Agreement...

Table 8: Qualitative zh-Hant→en examples, glossary in prompt against the full pipeline, mandated term in bold.

Track	Cell	Units	REUSE		GUIDED		Repaired	Unmet (%)
			found	copied	found	used		
1	medical en→pl	55	0	0	0	0	42	21.3
1	mech. eng. en→pl	57	0	0	0	0	48	23.6
1	energy es→eu	881	7	0	63	0	452	11.1
1	automotion es→eu	1,544	16	0	101	0	939	14.6
2	medical en→pl	48	0	0	0	0	32	33.2
2	mech. eng. en→pl	48	0	0	0	0	34	7.4
2	machine-tool es→eu	2,104	14	1	260	0	1,213	15.9
2	railways es→eu	1,385	18	2	109	0	781	14.7
2	finance zh-Hant→en	3,119	7	2	19	19	725	3.8

Table 9: What the deterministic components did on the submitted runs. Units are paragraphs, or sentences for zh-Hant→en. *Found* is what the plan proposed and *copied* or *used* what survived the context check. Unmet is the share of required term-unit pairs still missing after the repair.

Backbone	Configuration	medical en→pl		mech. eng. en→pl		energy es→eu		automotion es→eu	
		chrF++	TS ₂₆	chrF++	TS ₂₆	chrF++	TS ₂₆	chrF++	TS ₂₆
submitted	Bare	69.3	53.2	65.6	25.1	65.4	51.7	63.4	60.1
	+ glossary	73.2	90.9	72.3	89.1	67.9	84.9	65.6	86.7
	Full	73.4	91.9	72.3	91.5	67.0	85.8	64.5	87.8
31B	Bare	73.3	62.9	71.5	37.6	67.6	61.1	65.3	64.0
	+ glossary	75.8	96.7	75.5	94.7	69.0	90.9	67.2	92.5
	Full	76.0	95.2	75.6	92.9	69.0	90.2	67.0	92.4

Table 10: Backbone scale on all four Track 1 test cells, scored with the organisers’ code. The submitted backbone is E2B on en→pl and NLLB with the E4B corrector on es→eu. The 31B block replaces the translator on en→pl and the corrector on es→eu, quantised to 4 bits in both cases.