

CUNI-ÚFAL at the WMT26 Terminology Translation Task

Aleš Manuel Papáček and Tomáš Sourada

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
Prague, Czech Republic
{papacek, sourada}@ufal.mff.cuni.cz

Abstract

This paper describes the CUNI-ÚFAL submission to the WMT26 Terminology Translation Task. We present a model-agnostic pipeline that selects document-relevant terminology for a general-purpose LLM, instantiated with GPT-5.6-Sol for both tracks. For Track 1 (explicit dictionary) we compare glossary placement, inline annotation, dictionary filtering, and terminology-aware revision: filtering the dictionary per document improves judged coverage and, slightly, translation quality. Inline term insertion buys a small coverage gain at the cost of fluency. For Track 2 (in-domain seed bitexts) we compare presenting them as translation history, extracting a dictionary and reusing the Track 1 pipeline, and their hybrid. Lacking development references, we evaluated with surface- and lemma-based term matching plus an LLM judge. Among submitted systems in the official evaluation, our Track 1 submission ties for the highest COMET-22 score and obtains the best XCOMET-XXL and MetricX-24 XL scores, and our primary Track 2 submission achieves the highest term success.

1 Introduction

The WMT26 Terminology Translation Task (Charkiewicz et al., 2026) studies document-level machine translation in specialized domains. Track 1 provides an explicit bilingual terminology dictionary, whereas Track 2 provides in-domain seed bitexts that can be used to infer relevant terminology.

The WMT24 General Translation Task findings described machine translation as entering the LLM era (Kocmi et al., 2024); by WMT25, most participants used LLMs (Kocmi et al., 2025). The trend also appeared in the WMT25 Terminology Translation Task (Semenov et al., 2025). We therefore design a model-agnostic pipeline for general-purpose LLMs rather than rely on a translation-specific architecture. Under our budget, we com-

pare approaches with the relatively inexpensive Gemini 2.5 Flash (Google DeepMind, 2025) and use the stronger, newer GPT-5.6-Sol (OpenAI, 2026) for the final submissions.

For Track 1, we study how to present a potentially large bilingual dictionary to an LLM. We compare filtering and prompt placement strategies, and test a multistep variant in which a separate model flags terminology errors for the translation model to revise.

For Track 2, the main question is whether a general-purpose LLM can infer and apply relevant terminology from seed bitexts alone, without an explicit dictionary. We compare presenting the seed pairs directly to the model as a translation history, extracting a dictionary from them and reusing the Track 1 pipeline, and a hybrid of both.

2 Related Work

Terminology constraints can be enforced during decoding or supplied as model input. Dinu et al. (2019) trained an NMT system to apply runtime glossary constraints, avoiding the overhead and brittleness of constrained decoding. For morphologically rich translation, Jon et al. (2021) found that agreement errors accounted for almost half of the errors from a constrained English–Czech baseline and showed the benefit of input-side guidance. Bogoychev and Chen (2023) similarly annotated source terms and applied LLM refinement after detecting constraint violations. We also use input-side guidance, but filter the glossary per document and use off-the-shelf LLMs without fine-tuning.

Recent WMT systems combine LLMs with terminology selection, extraction, or post-editing. Ingólfssdóttir et al. (2025) select relevant glossary entries and include dictionary lookup results in an LLM prompt. For the WMT25 document-level evaluation, Semenov et al. (2025) use GPT-4o to extract bilingual terminology from document pairs.

Jaswal (2025) combine terminology-aware NMT with LLM post-editing, whereas Garcia Gilabert et al. (2025) optimize terminology adherence and translation quality through Group Relative Policy Optimization (GRPO). In contrast, our Track 2 experiments compare direct use of seed bitexts, glossary extraction, and their hybrid without model training.

3 Task and Data

We summarize below the parts of the setup our system depends on; Charkiewicz et al. (2026) give the full task description, data statistics, and evaluation protocol.

Track 1 provides an explicit dictionary and requires translations in three modes: without terminology (noterm), with the domain dictionary (proper), and with a random dictionary (random). The random dictionary has the same format but holds words drawn at random from the input texts rather than domain terms. Track 1 covers Spanish–Basque (Es–Eu) in the automotion and energy domains, and English–Polish (En–Pl) in mechanical engineering and medical.

Track 2 provides in-domain seed bitexts instead of a dictionary and has two modes: sample, with seed bitexts, and noterm, without. It covers five domains over three language pairs: En–Pl in the same two domains as Track 1, Es–Eu in machine tools and railways, and Traditional Chinese–English (Zh–En) in finance.

The released data contains 87 Track 1 documents and 129 Track 2 documents. Across the Spanish–Basque and English–Polish datasets, the mean document lengths range from approximately 1,000 to 1,800 words. Track 2 contains much larger outliers, including a 16,549-word Spanish railway document.

4 System Description

Our model-agnostic pipeline sends each complete document, translation instructions, and mode-specific terminology resources to an LLM. Figure 1 summarizes both tracks. The implementation is available at <https://github.com/ampapacek/wmt26-terminology-mt>, with further details and prompts in Appendices A and C.

4.1 Track 1

Dictionary Prompting. For the proper and random modes, the system first identifies dictionary

source terms that occur in the current document. We use Stanza (Qi et al., 2020) to lemmatize both the source document and each dictionary source term. An entry is selected when either its surface tokens or its lemma sequence occurs as a contiguous whole-token span in the document. Only the selected entries are then shown to the LLM, including all acceptable target options provided by the task.

Our default prompt places the filtered dictionary before the source document. We compare it with placing the dictionary after the document and inserting source–target annotations directly into the source text. In the inline variant, a matched span is rendered, for example, as `⟨⟨Turbinas eólicas [dict: turbina eólica] -> haize-sorgailu⟩⟩`. This keeps the inflected source form while also showing its dictionary entry and prescribed target at the point where it occurs.

We test three diagnostic variants: merging the proper and random dictionaries, adding the other domain’s dictionaries for the same direction, and supplying the complete mode-specific dictionary without document filtering. The first two test robustness to larger, noisier resources. The third isolates the contribution of per-document selection and is also relevant to potentially noisy Track 2 dictionaries.

Document Structure Validation. The official validator requires the output to preserve the source structure, treating blank-line-separated text as paragraphs and otherwise counting individual lines. We state the expected count in the prompt and verify it after generation. Mismatches trigger up to two whole-document retries, remaining failures are translated in batches of at most 25 paragraphs and reassembled with the original delimiter. For the weaker models, a little over 10% of documents needed repair, and some of those reached the batched fallback, whereas GPT-5.6-Sol needed it for about 1% of documents, all of which were resolved on the first retry.

Judge-Gated Terminology Revision. The complete Track 1 pipeline works as follows: After the initial translation, deterministic surface and lemma matchers check the output. An LLM from a different model family then examines only the terms not detected by either matcher and determines whether the translation contains a valid inflected form, an incorrect translation, or a genuinely missing term. If no revision is performed, these judgments are used

to compute the final judged coverage (Section 5.3).

In the optional revision phase, the translation model receives the source document, its initial translation, the same filtered dictionary used during the first phase, and the issues identified by the judge. It is instructed to correct the terminology while leaving the remainder of the translation unchanged.

4.2 Track 2

Track 2 provides seed bitexts rather than an explicit dictionary. We build three systems from them: (1) *end-to-end*, which shows the seed bitexts as a multi-turn translation history, (2) *dictionary*, which extracts a dictionary from them and then runs the Track 1 pipeline, and (3) *hybrid*, which does both. The three share the same noterm configuration, because they differ only in how they use the seed bitexts.

Document structure validation (Section 4.1) is applied unchanged, with the seed history carried along on every retry.

End-to-End Translation. Selected seed pairs form a multi-turn conversation: each source is a user turn, its gold translation the next assistant turn, and the document the final user turn. We cap this history at 80,000 source-plus-target characters per document¹. All pairs fit in four of the five domains and are shown in file order. The exception is *finance.zhen* (Traditional Chinese → English translation), whose 15 seed pairs are document-level bitexts with essentially zero verbatim overlap with the test documents, so we first re-align them into sentence pairs with a length-based dynamic-programming alignment (Gale and Church, 1993). Under the budget, pairs are then ranked by IDF-weighted cosine overlap between the seed source and the document for translation (character bigrams for Chinese, IDF estimated over that domain’s own document collection) and selected greedily, but still presented to the model in corpus order. Two alternatives scored slightly worse on a small random subset and were not adopted: placing the most relevant pair immediately before the document, and ignoring the character budget entirely (which affected only *finance.zhen*).

When a seed source occurs verbatim in the document, the system prompt additionally asks the model to reuse its approved translation. For more details on the prompt, see Appendix C.4.

¹The threshold was selected ad hoc

Extract Dictionary, Then Apply Track 1 Pipeline. As an alternative, we extract a dictionary from the seed bitexts and translate with the Track 1 pipeline.

The term dictionary is obtained by prompting an LLM² once per seed pair (once per ten aligned sentence pairs for *finance.zhen*), asking for both the source and target term in citation form (singular, dictionary form, no case or agreement marking) together with the verbatim source span the term was taken from. The verbatim span is checked against its sentence and used to drop hallucinated entries; a frequency-based check additionally repairs entries whose key turns out to be written in the target rather than the source language, and duplicate entries are merged by surface form and lemma.

The dictionary is extracted with the same model that then translates with it, and is consumed by the Track 1 pipeline; the seed bitexts themselves are not shown to the translation model in this mode (prompt in Appendix C.5).

Hybrid. The hybrid system combines both: the document is translated with the end-to-end prompt described in Section 4.2, and the extracted dictionary’s glossary block is appended to that prompt’s system message, so the model sees the seed pairs as conversation history and the terms of the current document as an explicit glossary at the same time.

5 Experimental Setup

5.1 Track 1

Models and Variants. We use Gemini 2.5 Flash (Google DeepMind, 2025) for the controlled comparison of Track 1 prompt variants, so that each experiment changes the terminology presentation while keeping the translation model fixed. We then compare the chosen default configuration for Gemini 2.5 Flash, Gemma 4 31B (Google DeepMind, 2026), and GPT-5.6-Sol.

Primary System. Our final Track 1 submission uses GPT-5.6-Sol (OpenAI, 2026) with temperature 0.2 and medium reasoning effort. It translates each document in one pass using the document-filtered dictionary placed before the source text. We do not apply the optional revision phase to the submitted output. We mark the submission by the identifier `cuni-ufal-gpt-56-sol`.

²We have also experimented with non-LLM extraction, but in preliminary experiments it led to worse results than the LLM extraction

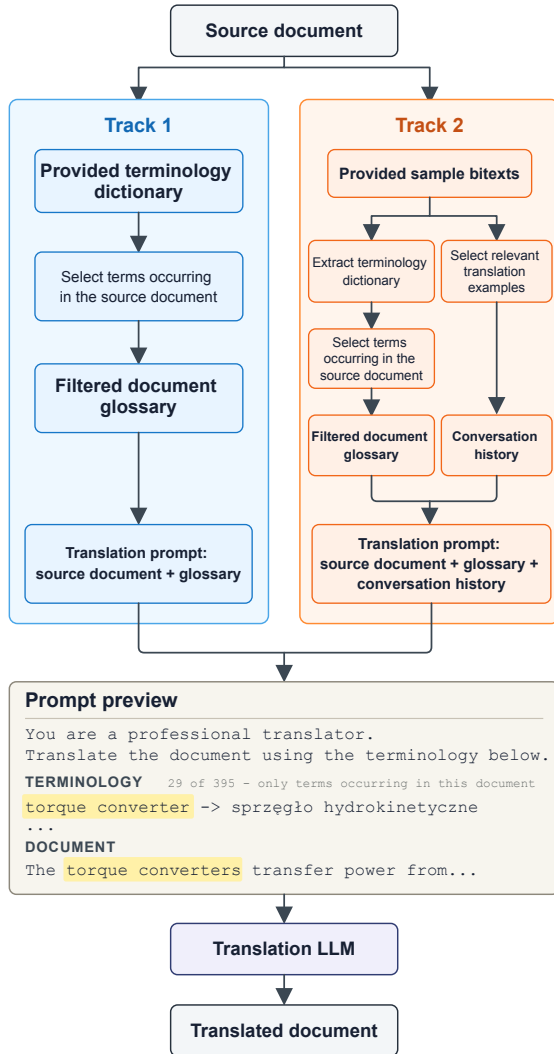


Figure 1: Overview of our primary Track 1 and Track 2 (hybrid) translation pipelines. The dictionary extraction in Track 2 is done with an LLM.

5.2 Track 2

Models and Systems. As in Track 1, we first compared the three systems on Gemini 2.5 Flash and only then verified them on the submission model, GPT-5.6-Sol.

Primary System. Based on the development-stage comparison (Section 6), we submit all three systems on GPT-5.6-Sol at temperature 0.2 and medium reasoning effort, each using the seed budget and dictionary-extraction mode selected on the development model: end-to-end with the default 80,000-character seed budget (cuni-ufal-gpt-56-sol-e2e), dictionary, extracted with an LLM (cuni-ufal-gpt-56-sol-dict-llm), and hybrid (cuni-ufal-gpt-56-sol), which is our

primary Track 2 submission (sharing the primary Track 1 identifier); end-to-end and dictionary are submitted as contrastive systems.

5.3 Reference-Free Evaluation

The task provides no development references, so we cannot compute reference-based translation metrics locally and instead use three complementary measurements. *Surface recall* checks whether any allowed target option occurs as a contiguous whole-token sequence. *Lemma recall* lemmatizes both the translated text and the dictionary target options before applying the same matching rule, using the same Stanza-based procedure as for source-side filtering with the corresponding Basque and Polish models. *Judged coverage* then combines these deterministic matches with inflected forms confirmed by an LLM judge, DeepSeek V4 Pro (DeepSeek-AI, 2026), which reviews only the terms neither matcher found. Judged coverage is our main terminology measure; we reserve *term success* for the official metric.

The judge also scores each document on a 1–5 scale: fluency and adequacy in Track 1, and fluency, adequacy, and terminology consistency in Track 2, where there is no dictionary to match against. In Track 2 it additionally lists the terminology problems it found, and for the four Es–Eu and En–Pl domains the gold translations of any seed sentences occurring verbatim in the document are attached as terminology evidence.

Surface recall is low for all systems because both target languages inflect heavily, and it should not be read as a quality measure on its own. It is informative in combination with lemma recall. A system whose surface recall is close to its lemma recall is reproducing dictionary forms as written rather than adapting them to context, which in a morphologically rich target language usually indicates an error.

Track 1 terminology metrics are micro-averaged over all required document–term pairs and judge scores averaged over documents; because the four domain–direction subsets range from 6 to 41 documents, this weights each subset by the amount of data it contributes. Track 2 scores are macro-averaged over the five domains, with failed judge calls excluded from the mean.

The judge is not deterministic. Four Track 2 sample files were judged twice on byte-identical input, as a by-product of a comparison that changed only finance.zhen. The repeated scores agree to

Variant	Surf.	Lem.	Cov.	Flu.	Adeq.
No terminology	.394	.658	.764	4.26	4.39
Default	.645	.894	.973	4.37	4.49
Text first	.654	.898	.972	4.39	4.38
Inline	.700	.904	.983	4.08	4.18
Proper + random	.618	.889	.967	4.32	4.38
All dictionaries	.609	.874	.959	4.24	4.39
No filtering	.585	.843	.930	4.29	4.32
Revision: errors only	.568	.837	.924	4.25	4.28
Revision: glossary + errors	.642	.900	.974	4.38	4.49

Table 1: Reference-free Track 1 results for prompting variants in the proper condition. Surf., Lem., and Cov. denote surface recall, lemma recall, and judged coverage, micro-averaged over the 2,477 required document-term pairs; Flu. and Adeq. are judge scores on a 1–5 scale, averaged over the 87 documents. The first row is the no term baseline, scored against the same dictionary.

within 0.06 on the two largest domains but differ by up to 0.29 on `mechanical-engineering.enpl` (7 documents) and `medical.enpl` (6). We therefore treat per-domain differences below 0.3 on those two domains as noise and rank systems by the macro-average.

These measurements are development proxies, not reproductions of the official evaluation, which reports document-level chrF++, exclusive lemmatized term success, and COMET-22, XCOMET-XXL, and MetricX-24 XL on aligned segments, aggregated by file and direction. Judged coverage is closest to official term success, but our deterministic matching and LLM adjudication differ from the official *exclusive* matching, in which each target-term occurrence is credited to at most one source occurrence.

6 Results and Analysis

6.1 Track 1

Selecting the Prompt Configuration. We first use Gemini 2.5 Flash in the proper condition to select the terminology-prompting configuration. Table 1 compares dictionary placement, inline insertion, dictionary selection, and revision variants. Across the four files, the evaluation contains 2,477 required document-term pairs. Supplying a filtered dictionary raises judged coverage from .764 to .973. Inline annotations produce the highest surface and judged coverage, but receive lower fluency and adequacy scores. Changing whether the dictionary appears before or after the document produces nearly identical scores.

We experiment with additional settings, *Proper + random* joins the proper and random dictionaries within a single domain. *All dictionaries* joins both domains and both modes. Both still apply the document-level filter before the glossary reaches the LLM, *no filtering* setup omits it. The proper dictionaries contain 374–761 entries, and the random dictionaries 642–816, of which 18.8–64.3 and 111.7–313.9, respectively, remain per document on average after filtering. Across all document prompts, filtering retained 20.8% of the random-dictionary entries, compared with 4.2% of the proper-dictionary entries, consistent with the random dictionaries containing more broadly occurring terms.

Removing the filter reduced judged coverage in every file: by 1.8–6.7 percentage points for the proper dictionaries and 1.0–3.2 points for the random dictionaries. Fluency and adequacy changed only slightly and without a consistent direction. These results suggest that filtering helps the model focus on terminology relevant to the current document.

Terminology-Aware Revision. Both revision variants receive the source, initial translation, and judge-identified errors. The *errors only* variant includes only flagged terms, whereas *glossary + errors* retains the original filtered glossary. The former reduced judged coverage from .973 to .924 by changing some correct terms; the latter restored it to .974 but did not improve on the initial translation. Revision therefore provides little benefit for already strong translations.

Selected Configuration. Based on these experiments, we select the document-filtered dictionary placed before the source document and do not apply revision. A comparison of this configuration across models and an analysis of document-level terminology consistency are provided in Appendix B.

6.2 Track 2

Our reference-free comparison preferred the end-to-end system with Gemini 2.5 Flash, whereas the hybrid received the highest terminology-consistency score with GPT-5.6-Sol. The full comparison is reported in Appendix B.

6.3 Official Results

Table 2 reports the official main-metric results for our submissions, as computed by the task organizers (Charkiewicz et al., 2026). In Track 1, the

System	Term	chrF++	COMET	XCOM.	MX ↓
<i>Track 1: proper</i>					
Primary	93.4	74.4	89.2	85.9	3.55
<i>Track 2: sample</i>					
Hybrid (primary)	89.2	78.1	88.7	86.0	3.22
End-to-end	87.1	78.1	88.9	86.5	3.14
Dictionary	89.0	74.9	88.5	85.9	3.25

Table 2: Official main-metric results for our Track 1 proper and Track 2 sample submissions. Term is exclusive lemmatized term success (%). XCOM. and MX denote XCOMET-XXL and MetricX-24 XL. Higher is better except for MetricX-24 XL.

primary system obtains 93.4% term success, ties for the highest COMET-22 score, and obtains the best XCOMET-XXL and MetricX-24 XL scores among submitted systems. Compared with the noterm condition, the proper dictionary improves term success by 22.1 percentage points, chrF++ by 1.3 points, and COMET-22 by 0.6 points. The random condition does not improve term success against the proper dictionary (69.3% versus 71.3% for noterm).

In Track 2, the primary hybrid system achieves the highest submitted term success, at 89.2%. The end-to-end contrastive system has lower term success but slightly better neural scores, indicating a small terminology–quality trade-off. Relative to noterm, seed bitexts improve the hybrid by 15.5 term-success, 4.7 chrF++, and 1.2 COMET-22 points.

7 Illustrative Downstream Use Case

Terminology-aware translation can also support multilingual retrieval. In the CZDEMOS4AI project, Czech user queries are translated for retrieval from an English-language collection on digital addictions. For the query *Hraní her*, general-purpose MT translated it as *gambling*, whereas translation aligned with the collection terminology used *gaming*. In this example, the relevant document ranked first with the collection-aligned query, compared with seventh and outside the context passed to the generation model with the general translation.

8 Conclusion

We presented a model-agnostic prompting system for both tracks of the WMT26 Terminology Translation Task. Across both tracks, a general-purpose LLM follows terminology well when the terminol-

ogy it is shown is filtered to the current document. Our Track 1 submission achieves the strongest or joint-strongest neural scores among submitted systems, and our Track 2 hybrid the highest term success. In both tracks terminology information also improves chrF++ and COMET-22.

Limitations

- **Evaluation.** Our development evaluation is reference-free, using neither an established quality-estimation metric nor human validation of the judge; we therefore cannot report human agreement (including for Basque) or precision and recall for terminology adjudication, and a target-term occurrence may occasionally be attributed to the wrong source occurrence. A different model family reduces direct self-preference but not evaluator bias, and we did not systematically measure judge stability, so we report deterministic matches separately and treat small judge-score differences as noise.
- **Generality.** We test few models with one shared prompt, without model-specific tuning. Most comparisons use Gemini 2.5 Flash and may not transfer to other models.
- **Revision.** The same judge identifies revision issues and scores the revised output, potentially favouring revision; these rows are therefore not directly comparable to the others.

AI usage statement

Research code. A substantial part of the experimental code base was written with Claude Code during July 2026, working from task descriptions, design decisions, and review comments provided by the authors. The authors specified what each component should do, reviewed the generated code, and are responsible for its correctness; the design decisions the code implements are the authors’ own and are documented in the paper.

Writing. Parts of the manuscript were drafted with Claude Code from the authors’ notes, logs, and result files, then checked and edited by the authors, who take full responsibility for all content, claims, and numbers.

Acknowledgments

This work was supported by the project “Human-centred AI for a Sustainable and Adaptive Society” (reg. no.: CZ.02.01.01/00/23_025/0008691),

co-funded by the European Union, and partially by the SVV project number 260 821. This work has been supported by the project TQ12000040 (CZDEMOS4AI) financed by the Technology Agency of the Czech Republic (www.tacr.cz) within the Sigma 3 Program. The work described herein has been using services provided by the LINDAT/CLARIAH-CZ Research Infrastructure (<https://lindat.cz>), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062). Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic. We thank to Martin Havelka for his help with developing the first version of the system.

References

- Nikolay Bogoychev and Pinzhen Chen. 2023. [Terminology-aware translation with constrained decoding and large language model prompting](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 890–896, Singapore. Association for Computational Linguistics.
- Adrian Charkiewicz, Pinzhen Chen, Thierry Etchegoyhen, Harritxu Gete Ugarte, Kamil Guttman, Xu Huang, David Ponce, Artur Nowakowski, Frédéric Odehmann, 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.
- DeepSeek-AI. 2026. DeepSeek V4 preview release. <https://api-docs.deepseek.com/news/news260424/>. Accessed 2026-07-28.
- Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. [Training neural machine translation to apply terminology constraints](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.
- William A. Gale and Kenneth W. Church. 1993. [A program for aligning sentences in bilingual corpora](#). *Computational Linguistics*, 19(1):75–102.
- Javier Garcia Gilabert, Carlos Escolano, Xixian Liao, and Maite Melero. 2025. [Terminology-constrained translation from monolingual data using GRPO](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1335–1343, Suzhou, China. Association for Computational Linguistics.
- Google DeepMind. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.
- Google DeepMind. 2026. Gemma 4 model card. https://ai.google.dev/gemma/docs/core/model_card_4. Accessed 2026-07-29.
- Svanhvít Lilja Ingólfssdóttir, Haukur Jónsson, Kári Steinn Adhalsteinsson, Róbert Fjölfnir Birkisson, Sveinbjörn Thórhðarson, and Thorvaldur Páll Helgason. 2025. [Midheind at WMT25 general machine translation task](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 577–582, Suzhou, China. Association for Computational Linguistics.
- Akshat Jaswal. 2025. [It takes two: A dual stage approach for terminology-aware translation](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1344–1350, Suzhou, China. Association for Computational Linguistics.
- Josef Jon, João Paulo Aires, Dusan Varis, and Ondřej Bojar. 2021. [End-to-end lexically constrained machine translation for morphologically rich languages](#). 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 4019–4033, Online. Association for Computational Linguistics.
- Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakouga, 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, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 others. 2024. [Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.
- OpenAI. 2026. GPT-5.6 Sol model. <https://developers.openai.com/api/docs/models/gpt-5.6-sol>. Accessed 2026-07-29.
- Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. *Stanza: A python*

natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, 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.

A Implementation and Reproducibility

All LLM calls were made through the OpenRouter v1 API in July 2026. We used the models (ID on OpenRouter) google/gemini-2.5-flash, google/gemma-4-31b-it, openai/gpt-5.6-sol, and deepseek/deepseek-v4-pro for evaluation. Translation used temperature 0.2 and a default output limit of 32,000 tokens. GPT-5.6-Sol used medium reasoning effort.

Terminology evaluation uses the Stanza eu_bdt and pl_pdb models for lemma matching, followed by DeepSeek V4 Pro review of unmatched terms.

The final GPT-5.6-Sol Track 1 sweep covered 261 document–mode instances, of which three required an automatic paragraph-count repair. The completed sweep cost approximately \$28.46 for translation and \$3.11 for evaluation. All 12 output files passed the official task validator.

The three final GPT-5.6-Sol Track 2 sweeps covered 129 documents in each of the two modes per system. The noterm condition is configuration-identical across the three systems, so it was translated once with the end-to-end system and copied to the other two; 516 document–mode instances were therefore generated in total, of which seven required an automatic paragraph-count repair and none failed. Translation cost approximately \$47.31 for the end-to-end system (\$15.62 of that the shared noterm files and \$31.69 its own sample files), \$31.00 for the hybrid system and \$15.90 for the dictionary system, \$94.21 in total. Extracting the GPT-5.6-Sol dictionary over all five domains cost a further \$24.24, and the reference-free evaluation of the three systems’ sample outputs cost \$5.41. All ten output files of every system passed the official task validator.

Direction	No term	Proper	Random
<i>Gemini 2.5 Flash</i>			
Es–Eu	.795 / 4.23	.967 / 4.36	.874 / 4.14
En–Pl	.701 / 4.43	.985 / 4.07	.876 / 4.21
<i>Gemma 4 31B</i>			
Es–Eu	.510 / 2.32	.974 / 2.48	.918 / 2.48
En–Pl	.620 / 4.00	.989 / 4.00	.921 / 3.14
<i>GPT-5.6-Sol</i>			
Es–Eu	.846 / 4.60	.975 / 4.63	.908 / 4.30
En–Pl	.780 / 4.50	.989 / 4.50	.900 / 4.36

Table 3: Track 1 judged coverage and fluency by translation direction, reported as coverage / fluency. Coverage is micro-averaged over required document–term pairs and fluency over documents within each direction. Bold marks the highest value for each measure within a row, including ties.

B Additional Local Results

B.1 Track 1: Comparison Across Models

Table 3 compares the selected document-filtered dictionary configuration across models and under the three terminology modes. The noterm and proper conditions are evaluated against the proper dictionary, while random is evaluated against the random dictionary and is therefore not directly comparable.

GPT-5.6-Sol receives the highest fluency scores in both directions and reaches near-ceiling judged coverage with the proper dictionary. Gemma 4 31B also obtains high proper-dictionary coverage, but its fluency differs sharply by direction: 2.32–2.48 for Spanish–Basque, compared with 3.14–4.00 for English–Polish. The gap is also present without a dictionary, so dictionary bias alone cannot explain it. We additionally observed one degenerate repetition loop during generation. Because all models use the same prompt without model-specific tuning, these results characterize performance under our fixed configuration rather than each model’s best achievable quality.

B.2 Track 1: Document-Level Consistency

We measure whether Gemini 2.5 Flash uses the same target option for repeated occurrences of a source term. For multi-option entries, only 41 of 1,210 recognized occurrences (3.4%) in the proper condition use an option other than the document’s most frequent one. The rates are 2.9% for noterm and 3.4% for random. The calculation uses deterministic lemma matches.

Domain	E2E	Dict.	Hybrid
<i>Gemini 2.5 Flash (development)</i>			
Finance	4.91/4.64	3.81/3.60	4.26/3.91
Machine tool	4.37/3.80	4.14/3.91	4.43/4.14
Mech. eng.	3.43/3.00	3.71/3.00	3.43/ 3.14
Medical	4.00/3.17	3.83/ 3.50	4.17/3.50
Railways	4.32/3.86	4.04/3.48	4.25/3.46
Macro	4.21/3.69	3.91/3.50	4.11/3.63
<i>GPT-5.6-Sol (submitted)</i>			
Finance	5.00/4.76	4.94/ 4.79	5.00/4.74
Machine tool	4.66/4.43	4.37/4.03	4.60/ 4.63
Mech. eng.	3.57/2.57	4.14/3.43	3.43/ 3.43
Medical	4.33/3.83	4.50/4.00	4.17/ 4.17
Railways	4.61/4.50	4.48/4.00	4.54/4.25
Macro	4.43/4.02	4.49/4.05	4.35/ 4.24

Table 4: Track 2 judge scores by domain. Each cell gives fluency/terminology consistency (1–5, higher is better). The macro rows average the five domains equally. The numbers of judged documents are 53, 35, 7, 6, and 28 in the displayed domain order. Bold marks the highest value for each measure within a row, including ties.

B.3 Track 2: Reference-Free Development Results

We compare the end-to-end, dictionary, and hybrid systems, first on Gemini 2.5 Flash and then on GPT-5.6-Sol. Table 4 reports the DeepSeek V4 Pro judge fluency and terminology-consistency scores by domain and their five-domain macro-average at both stages.

On Gemini 2.5 Flash, the end-to-end system receives the highest macro-average score on both reported axes. That ranking does not transfer to GPT-5.6-Sol, for which the dictionary system receives the highest fluency and the hybrid the highest terminology-consistency score. The per-domain results are more mixed for both models: no system consistently performs best across domains, so no single system is uniformly superior.

C Prompt Templates

The templates below reproduce the operational instructions while replacing dynamically inserted values such as language names, paragraph counts, documents, translations, and glossary entries with placeholders. Repeated delimiter-specific wording is condensed. The same translation template is used with all supported LLMs.

C.1 Track 1 Translation Prompt

The system message begins with the following instructions:

You are a professional *[source]*-to-*[target]* translator specializing in technical, engineering, medical, and scientific documents. Translate the document below from *[source]* into *[target]*.

Rules:

1. Translate the entire document faithfully and fluently.
2. The source has exactly *[count]* paragraphs separated by blank lines, or exactly *[count]* lines separated by single line breaks. Output the same number in the same order and do not merge, split, add, or drop them.
3. Keep brand names, product codes, and proper nouns untranslated unless a translation is standard.
4. Do not add explanations, notes, or commentary.
5. Output only the translated text.

In the default setting, the following block is appended to the system message before the source document:

Terminology glossary (use these target translations):

[source term] → *[target option 1]* | *[target option 2]*

For each glossary entry, when the source concept appears in the text, render it using the given target term, inflected or agreed as the target-language grammar requires. When several options are listed, pick the one that best fits the context and use it consistently. Prefer natural, fluent phrasing over a literal or forced insertion. If a term genuinely cannot fit grammatically, translate the concept faithfully instead.

The noterm mode uses the same translation instructions without this block. The proper and random modes differ only in the dictionary from which the displayed entries are selected. In the text-first variant, the same block follows the source document. The inline variant instead marks a matched span as *⟨⟨source span → target option⟩⟩* and asks the model to remove the marker while using an appropriately inflected target form.

C.2 Terminology Evaluation Prompt

The judge receives the source document, its translation, and only the dictionary entries missed by the surface and lemma matchers. Its system message contains the following tasks:

Task 1: For every listed dictionary entry, choose exactly one verdict:

- correct-inflected: an acceptable target option is present but was missed because it is inflected, declined, agglutinated, or has a minor orthographic variation;

- *wrong-sense-or-mistranslated*: the source concept is rendered, but not with an acceptable option;
- *genuinely-missing*: the source concept is absent from the translation.

In the note, quote a short relevant translation fragment when one exists.

Task 2: Rate the complete translation for fluency and adequacy from 1 to 5. For fluency, 1 denotes broken or unreadable text and 5 natural, native-quality text. For adequacy, 1 denotes meaning that is mostly wrong or omitted and 5 a complete and precise translation.

Return strict JSON only: {"fluency": [int], "adequacy": [int], "terms": [{"source": [source term], "verdict": [verdict], "note": [evidence]}}. The terms array must contain exactly one entry for every supplied dictionary entry, in the same order.

The user message contains the source document, the translation, and the unmatched dictionary entries with all acceptable target options.

C.3 Revision Prompt

The revision model receives the source document, the initial translation, the full filtered glossary used for the initial translation, and the terminology issues identified by the judge. Its system message gives the following rules:

1. Check each reported issue against the draft. If the draft already uses an acceptable target option in an appropriate inflected form, treat the issue as a false positive and leave it unchanged.
2. Return a revised translation of the complete document, correcting only issues that are genuinely wrong or missing.
3. For each real issue, use one of the acceptable target terms and inflect or agree it as required by the target-language grammar. Prefer natural phrasing over forced insertion.
4. Keep all wording and style not affected by a genuine issue unchanged.
5. Preserve exactly the source number and ordering of paragraphs or lines.
6. Do not add explanations, notes, or commentary; output only the revised translation.

C.4 Track 2 End-to-End Translation Prompt

The end-to-end system message opens like the Track 1 prompt (Appendix C.1), but its third rule refers to the provided examples rather than to a glossary:

You are a professional *[source]*-to-*[target]* translator specializing in technical and financial documents. Translate the document below from *[source]* into *[target]*.

Rules:

1. Translate the entire document faithfully and fluently.
2. The source has exactly *[count]* paragraphs separated by blank lines, or exactly *[count]* lines separated by single line breaks. Output the same number in the same order and do not merge, split, add, or drop them.
3. In sample mode: follow the terminology and style of the reference translations you have been given. In noterm mode (no examples given): render every domain-specific term consistently throughout the document.
4. Do not add explanations, notes, or commentary.
5. Output only the translated text.

In sample mode, the selected seed pairs are then sent as alternating user/assistant turns preceding the document, which forms the final user turn:

```
system: the instructions above
user: [seed source 1]
assistant: [seed target 1]
...
user: [document to translate]
```

When a seed source occurs verbatim in the document, an additional system-message rule asks the model to reuse that seed's approved translation for the matching span. For the dictionary and hybrid systems, the terminology glossary block from the Track 1 prompt (Appendix C.1) is instead built from the terms extracted for the current document. The dictionary system shows this block with no example history, in an otherwise unchanged Track 1 system message; the hybrid system appends it to the end-to-end system message above, so both the example history and the glossary are present at once.

Whenever the history is non-empty, the system message also appends one line stating how many previously approved translations follow and instructing the model to treat them as reference material – reuse their terminology and register – rather than text to translate again.

C.5 Term Dictionary Extraction Prompt

This is the prompt used to build the dictionary consumed by the dict and hybrid systems (Section 4.2): one call per seed pair (per ten aligned pairs for *finance.zhen*).

You build a *[source]*-to-*[target]* terminology dictionary from aligned sentence pairs. Extract only *[domain]* DOMAIN TERMINOLOGY: the specialised terms of the subject field. Do not extract general-language vocabulary, function words, everyday verbs and adjectives, dates, or plain numbers.

Rules:

1. Give both sides in their dictionary (citation) form: singular, nominative/absolute, no case suffix, no attached article, no possessive, verbs in the infinitive.
2. Citation form applies to the phrase as a whole, never word by word: every modifier must agree with the head noun again in the citation form, on both the source and the target side.
3. Also report, in *source_surface*, the exact substring of the source sentence the term was taken from, copied character for character.
4. The target must render the source term as the target sentence does (same sense, same register), in its own citation form.
5. One entry per distinct source term; alternative target renderings go in the same entry's target list.
6. Terms are words or short phrases, never whole sentences; if a pair has no extractable terminology, return no entries for it.

Output strict JSON only: {"terms": [{"pair": [int], "source": [citation form], "source_surface": [verbatim source substring], "targets": [[citation form], ...]}, ...]}.

The user message lists the batch's sentence pairs, numbered. Two worked examples are included in the system message to illustrate the phrase-as-a-whole agreement rule (rule 2) on both the source and the target side, since this is where the model most often reverts to per-word citation forms instead. *source_surface* is what the hallucination guard checks (Section 4.2): an entry whose reported span does not occur verbatim in its own source sentence is dropped before merging.