

RWS at WMT26 Terminology Translation Task

Jon Cambra Guinea and Andrea Alfieri

jcambra@rws.com

aalfieri@rws.com

Abstract

This paper presents RWS’ submission to the WMT26 terminology task: Translation-Queries-Memory (TQM), a prompt-based terminology and segment injection strategy for terminology translation across domains. We evaluate TQM on Basque to Spanish, English to Polish, and Traditional Chinese to English for WMT 2026 Terminology Shared Task 1 & 2. We demonstrate that injecting bilingual terminology and bi-text examples into recent Large Language Models’ (LLM) prompts improves both overall translation quality and terminology accuracy. Our results show that strong instruction-following ability allows a system to adapt to terminology and style constraints without the need for fine-tuning.

1 Introduction

LLMs have become strong general-purpose translation systems, matching or outperforming previous encoder-decoder models on several language pairs (Garcia et al., 2023; Vilar et al., 2023). However, fluency alone is often insufficient for professional industry translation workflows, where accurate domain terminology is key to determine whether a translation is fit for purpose. Ensuring adherence to a set of approved term translations is therefore a long-standing concern in machine translation (MT), as evidenced by the WMT26 Terminology Translation task (Charkiewicz et al., 2026).

An up-to-date method to drive LLM translation is in-context learning. Instead of modifying the decoder, fine-tuning, or retraining the model, relevant translation information is injected into the prompt. This idea has been explored in two directions in recent times. On the one hand, terminology injection provides the model with source-target term pairs expected to be used at inference time. Moslem et al. (2023b) proved that the addition of approved terms to the prompt substantially

increases the proportion of terms that appear correctly in the output. The benefit of terminology injection is not merely anecdotal. In WMT25’s sentence/paragraph-level terminology translation task (Semenov et al., 2025), the first ranked system was simply OpenAI’s o3 with terminology constraints injected into the prompt. Similarly, in the document-level terminology translation task, the winning system Erlendur (Ingólfssdóttir et al., 2025) incorporated glossary information as part of the prompt.

In addition, injecting translation examples conditions LLMs on retrieved translation-memory (TM) matches, exploiting the in-context learning ability of LLMs (Moslem et al., 2023a): fuzzy matches used as few-shot examples improve translation quality over both zero-shot translation and randomly sampled examples, extending the earlier idea of integrating fuzzy matches into neural MT (Bulte and Tezcan, 2019; Cai et al., 2021); the same principle has since been carried into the fine-tuning stage, where retrieval-augmented few-shot examples are combined with instruction tuning for domain-specific MT (Zheng et al., 2024). The advantages of this approach were also demonstrated in the WMT25 terminology translation task. The submission TiUTerm-V1 used XGLM-7.5B with 12 most similar in-context examples retrieved from the dev set (Semenov et al., 2025), and the system ranked fourth according to the CHRF score.

Despite these findings, terminology injection and TM example injection are typically studied in isolation and largely on high-resource and English-centric language pairs, where LLMs have been shown to perform strongly. It remains unclear how well these approaches transfer to lower-resource settings, where both term adherence and overall quality are harder to achieve.

We compare different LLMs and submit the optimal approach to WMT26 Terminology Translation Task for Spanish into Basque (es-eu), English into

Polish, and Traditional Chinese into English (zh-Hant-en) language pairs.

2 System description

TQM is our retrieval-augmented, LLM-based translation system designed to adapt to a particular style and terminology without sacrificing overall translation quality. TQM applies in-context learning: for each source segment, relevant information is retrieved and placed in the prompt before translation.

2.1 Terminology injection

We detect source terms with an in-house constraint-generation component. Given a source document and one or more bilingual dictionaries, it produces a set of terminology constraints for each translatable segment. At load time, each dictionary source term is tokenized and expanded into its inflected surface forms, so that the morphological variants of a term can also be matched. Detection then proceeds by greedy longest-match-first n-gram matching over the tokens of each source segment: the longest dictionary entry that fits a span is preferred, ensuring that multi-word terms take precedence over their constituent words, and overlapping candidates are resolved by a priority score. Each match yields the source span together with its prescribed target translation. These detected source-target term pairs are subsequently injected into the prompt.

2.2 Sample sentence injection

When translation memories (TM) are available, we condition the model on retrieved parallel translation examples extracted from them. This differs from terminology injection: while term pairs guide lexical choices, full segment pairs serve as in-context examples of a given domain which can inform the vocabulary, style, grammar, register and patterns of the output. The user can therefore benefit from the in-context learning ability of LLMs.

For each source sentence in a text fragment to translate, we retrieve the k most similar source sentences from the TM by dense embedding similarity. Each source sentence in the TM is encoded into a multilingual vector using paraphrase-multilingual-MiniLM-L12-v2 from sentence transformers (Reimers and Gurevych, 2019, 2020), a multilingual sentence encoder that maps semantically similar sentences close together in a shared embedding space across

languages. We embed the TM source side offline, and store the resulting vectors in a vector index to allow efficient search. At inference time, we encode the source sentence to translate with the same model and compute its cosine similarity against the TM vectors. The k TM entries with the highest scores are retrieved, and the corresponding translation pairs are injected into the prompt as few-shot examples.

2.3 Model selection

For the task, we analyze the behavior of three LLMs across all settings.

LW Research Pro. RWS’ private research model finetuned for translation.

TranslateGemma-27B. Google’s Gemma3 finetuned for translation (Finkelstein et al., 2026), in a GGML 4-bit quantized version.

Qwen35-35B. Alibaba’s open-source Qwen3.5-35B-A3B Instruct model (Qwen Team, 2026), in a GGML 8-bit quantized version.

3 WMT submissions

We used TQM for both Document-Level Translation with Dictionaries (Track 1) and Document-Level Translation with Sample Sentences (Track 2)

3.1 Track 1: Document-Level Translation with Dictionaries

Track 1 involves document translation in Spanish-Basque (es-eu) for the energy and automation domains, and English-Polish (en-pl) for the medical and mechanical engineering domains. For each domain, a glossary is available to correctly translate terms when present in input documents. The solution needs to detect relevant terms in source segments and produce high-quality translations that use the correct term translation. To fully evaluate these abilities, three glossary modes are tested: noterm (no glossary provided), random (random, irrelevant terms provided), and proper (relevant, domain-specific terms provided).

3.1.1 Data pre-processing

Input documents differ between language pairs. In the es-eu data, each document is a concatenation of short paragraphs, sentences, and/or titles, whereas in the en-pl data, documents consist of concatenated long paragraphs. We segment the documents into the above units for each language, respectively.

	medical (en→pl)			mech.-eng. (en→pl)			energy (es→eu)			automotion (es→eu)		
System	TA	MetX	CK	TA	MetX	CK	TA	MetX	CK	TA	MetX	CK
<i>Proper mode</i>												
LW Research Pro	0.697	6.43	0.521	0.828	8.81	0.433	0.778	7.11	0.610	0.802	7.69	0.567
TranslateGemma-27B	0.661	7.27	0.518	0.831	9.77	0.427	0.812	6.09	0.641	0.871	6.70	0.607
Qwen35-35B	0.665	6.91	0.512	0.809	9.70	0.419	0.752	6.20	0.643	0.808	6.73	0.610
<i>Random mode</i>												
LW Research Pro	0.734	6.88	0.517	0.729	9.17	0.438	0.710	7.80	0.572	0.616	8.31	0.546
TranslateGemma-27B	0.728	8.27	0.471	0.718	10.13	0.434	0.749	6.46	0.627	0.643	7.43	0.586
Qwen35-35B	0.700	7.32	0.508	0.702	9.66	0.423	0.726	6.68	0.623	0.623	7.49	0.585
<i>Noterm mode</i>												
LW Research Pro	–	6.06	0.540	–	8.42	0.448	–	7.35	0.604	–	8.00	0.560
TranslateGemma-27B	–	7.13	0.525	–	9.23	0.462	–	6.16	0.640	–	6.91	0.601
Qwen35-35B	–	6.85	0.518	–	9.35	0.444	–	6.40	0.648	–	6.92	0.605

Table 1: Per-domain, per-language-pair comparison of systems in proper, random, and noterm modes. Best value per column *within each mode* in **bold**; term accuracy is undefined in no-terminology mode (–). TA = term accuracy (↑), MetX = MetricX-24-XL-QE (↓), CK = WMT23-CometKiwi-DA-XL (↑).

3.1.2 Results

In the absence of reference data, we estimate translation quality using reference-less metrics: MetricX-24-XL-QE (Juraska et al., 2024) and WMT23-CometKiwi-DA-XL (Rei et al., 2023). We also estimate term accuracy using an in-house implementation of a term-accuracy metric inspired on WMT25’s terminology translation metric.¹

Table 1 shows the performance of all systems across all modes. In term injection modes, TranslateGemma-27B leads the ranking in term accuracy in both es-eu domains. This aligns well with the language coverage in the training data. Indeed, the amount of Basque data in LW Research Pro training is scarce compared to a system like TranslateGemma-27B.

This is no longer the case for en-pl, LW Research Pro outperforms the other systems in 3 out of 4 cases according to term accuracy and is state-of-the-art in translation quality according to the reference-less metrics. Therefore, we submit translations with the winning system for each language pair: **TranslateGemma-27B in es-eu and LW Research Pro in en-pl**.

We note a clear degradation in translation quality for the random mode for all systems and language pairs. We hypothesize that injecting huge amounts of irrelevant terminology could be damaging translation quality.

3.2 Track 2: Document-Level Translation with Sample Sentences

Track 2 involves document translation in es-eu for the machine-tool and railways domains, en-

pl for the mechanical-engineering and medical domains, and Chinese-English (zh-Hant-en) for finance-domain data. For each domain, a sample bi-text file with in-domain translation examples is available, from which the system has to learn the terminology and translation style.

3.2.1 Data pre-processing

The data is also heterogeneous across languages. In the source documents, we observe the same discrepancies between es-eu and en-pl as in Track 1 and apply the same segmentation. Additionally, the zh-Hant-en document does not contain any clear paragraph separator; therefore, at inference time, we split the zh-Hant-en source documents into sentences to simplify the task.

In the sample bi-text files, translation pairs are sentence-level for es-eu and en-pl, but document-level for zh-en. We therefore pre-process the zh-en sample file to convert it into aligned parallel sentences: we divide the source and target documents into sentences, then ensure alignment by embedding the source and target sides with LaBSE (Feng et al., 2022) and maximizing the corresponding cosine similarity. The resulting sample segments can then be retrieved and injected into the prompts as described in Section 2.2. Figure 3 shows the similarity distribution for Track 2 between the source segments and the retrieved sample segments.

3.2.2 Results

Figure 1 shows translation quality for all systems and k values. Since translation-memory sample injection has a strong language-pair and system-dependent effect on quality, we select the optimal configuration separately for each language pair submission according to MetricX-24-XL-QE scores.

¹<https://github.com/wmt-conference/wmt25-terminology>

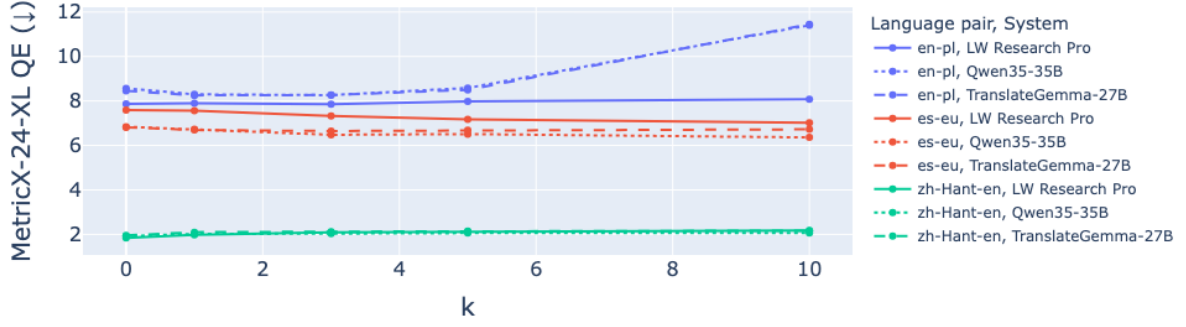


Figure 1: Effect of translation-memory sample injection on MetricX-24-XL QE (↓). Increasing the number of injected examples k does not uniformly improve quality: gains appear for zh-en and es-eu but en-pl degrades with larger k , indicating a strong language-pair and model dependence.

As measured by MetricX-24-XL-QE, generally, we observe similar behavior and translation quality across systems and language pairs.

- **es-eu** Quality improves monotonically with k for all 3 systems. As in Track 1, LW Research Pro lags behind the open-source systems given the low amount of Basque data it was trained on. Qwen35-35B reaches the best scores for all values of k . We therefore **select Qwen35-35B with $k=10$** .
- **zh-Hant-en** Translation quality is almost identical across systems for each value of k , and the quality curve is almost flat. However, as shown in Figure 3, identical or near-identical segments were included in the prompt, and the presented technique increases translation quality on exact matches, as discussed in Figure 2. This suggests that our metric may not fully capture the domain-specific requirements of this file. As a trade-off, **we submit Qwen35-35B with $k=3$** to ensure the LLM is extensively exposed to Chinese data and to sufficient translation style and terminology.
- **en-pl** Behavior differs drastically from the other pairs: adding more than 3 examples can harm MetricX-24-XL-QE scores, with the sole exception of LW Research Pro. We attribute this general degradation to input length. As described in Section 3.1.1, translation units correspond to paragraphs, but the k examples are injected per sentence, resulting in extremely long prompts for long paragraphs. Open-source LLMs are less able to adequately handle this long context to adapt style and terminology at the paragraph level. The best configuration in terms of translation quality

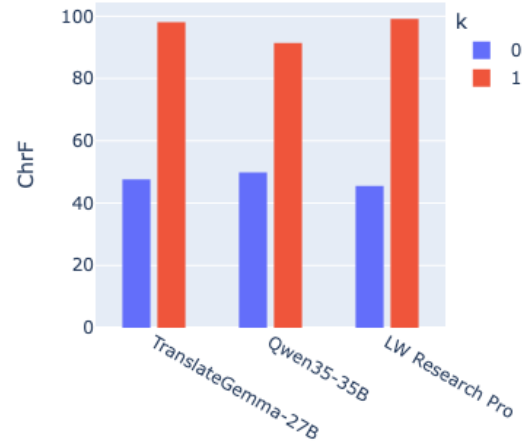


Figure 2: ChrF scores by model at $k=0$ vs. $k=1$ for segments with source-sample similarity > 0.99 (sample used as reference). Suggesting that the method is able to inject correctly the translation information from the bitext samples.

is achieved with LW Research Pro at $k = 3$. However, we could not submit that last result on time for the competition. As a consequence, **we submit a suboptimal solution, i.e. TranslateGemma-27B with $k=3$** .

4 Conclusion

We investigated terminology quality in MT by submitting two methods of in-context prompt injection: for Track 1, approved terminology to force term translation, and for Track 2, retrieved translation examples to improve overall quality. We evaluated this approach in different domains and language pairs. For Track 1, we submit TranslateGemma-27B in es-eu and LW Research Pro in en-pl with term injection; for Track 2, we

submit TranslateGemma-27B with $k = 3$ in en-pl and Qwen35-35B with $k = 10$ and $k = 3$ for es-eu and zh-Hant-en respectively. We show that this technique, applied to different types of LLMs, can lead to high quality results.

Limitations

Our RAG-based solution has several limitations. First, it depends heavily on the LLM’s language pair coverage, its accuracy on instruction-following tasks, and prompt format used. Secondly, this ability might not generalize to some domains or language pairs. Additionally, the search for the optimal k was implemented at the document level but could be better adjusted at the segment level based on similarity thresholds. Another avenue for improvement is the exploration of more recent embedding models and alternative similarity metrics. Finally, in the absence of reference or human-labeled data, our evaluation and configuration selection relied on reference-less metrics, which may not fully reflect in-domain style and terminology precision.

Acknowledgments

We thank Amos Kariuki for providing the terminology detection component used for this work. We are grateful to William Tambellini, Justin Beavers and Matias Lin for the efficient open-source LLM versions used for the experiments.

References

- Bram Bulte and Arda Tezcan. 2019. [Neural fuzzy repair: Integrating fuzzy matches into neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.
- Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. [Neural machine translation with monolingual translation memory](#). 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 7307–7318, Online. Association for Computational Linguistics.
- Adrian Charkiewicz, Pinzhen Chen, Thierry Etchegoyhen, Harritxu Gete Ugarte, Kamil Guttmann, 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.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- 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, and 2 others. 2026. [TranslateGemma technical report](#). *Preprint*, arXiv:2601.09012.
- Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. [The unreasonable effectiveness of few-shot learning for machine translation](#).
- Svanhvít Lilja Ingólfssdóttir, Haukur Páll Jónsson, Kári Steinn Aðalsteinsson, Róbert Fjölfnir Birkisson, Sveinbjörn Þórðarson, and Þorvaldur Páll Helgason. 2025. [Miðeind at WMT25 general machine translation task and terminology translation task](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 577–582. Association for Computational Linguistics.
- Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. [MetricX-24: The Google submission to the WMT 2024 metrics shared task](#). In *Proceedings of the Ninth Conference on Machine Translation*, Miami, Florida, USA. Association for Computational Linguistics.
- Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023a. [Adaptive machine translation with large language models](#). *Preprint*, arXiv:2301.13294.
- Yasmin Moslem, Gianfranco Romani, Mahdi Molaei, John D. Kelleher, Rejwanul Haque, and Andy Way. 2023b. [Domain terminology integration into machine translation: Leveraging large language models](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 902–911, Singapore. Association for Computational Linguistics.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. [Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, 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. Association for Computational Linguistics.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. [Prompting PaLM for translation: Assessing strategies and performance](#). *Preprint*, arXiv:2211.09102.

Jiawei Zheng, Hanghai Hong, Feiyan Liu, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. 2024. Dragft: Adapting large language models with dictionary and retrieval augmented fine-tuning for domain-specific machine translation. *arXiv preprint arXiv:2402.15061*.

A Prompts

The prompts used for terminology injection (Track 1) and sample sentence injection (Track 2) are shown below.

A.1 Track 1: Terminology Injection Prompt

```
You are a professional {source_lang} to
{target_lang} translator. Your goal is to
accurately convey the meaning and nuances of
the original {source_lang} text while
adhering to {target_lang} grammar,
vocabulary, and cultural sensitivities.
Produce only the {target_lang} translation,
without any additional explanations or
commentary. Please translate the following
{source_lang} text into {target_lang} where
'{source_term_1}' translates into
'{target_term_1}',
'{source_term_2}' translates into
'{target_term_2}',
...,
'{source_term_n}' translates into
'{target_term_n}':

{source_segment}
```

A.2 Track 2: Sample Sentence Injection Prompt

```
You are a professional {source_lang} to
{target_lang} translator. Your goal is to
accurately convey the meaning and nuances of
the original {source_lang} text while
adhering to {target_lang} grammar,
vocabulary, and cultural sensitivities.
Produce only the {target_lang} translation,
without any additional explanations or
commentary. Please translate the following
{source_lang} text into {target_lang} where
'{source_match_1}' translates into
'{target_match_1}',
'{source_match_2}' translates into
'{target_match_2}',
...,
'{source_match_k}' translates into
'{target_match_k}':

{source_segment}
```

B Cosine similarity distribution in Track 2

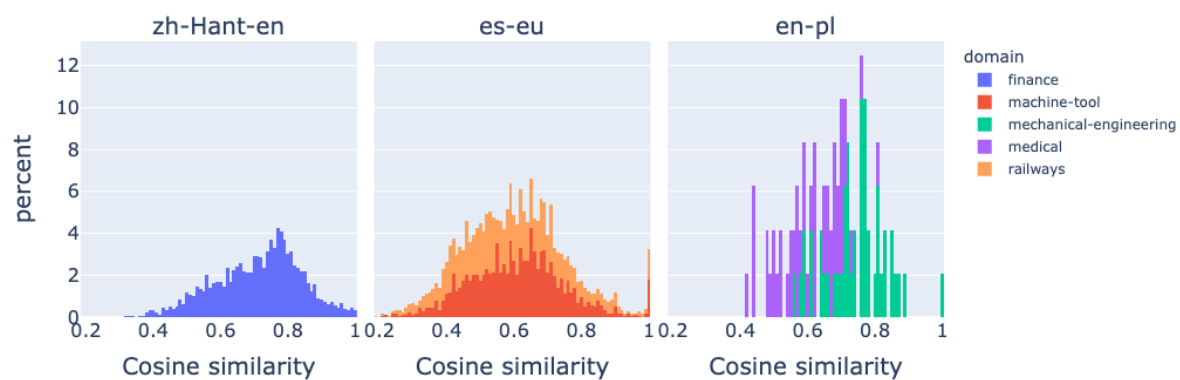


Figure 3: Cosine similarity distribution across language pairs between source segment and most similar source sample.