

TaT at WMT26 Terminology Translation Shared Task

Fan Zhang¹, Mei Tu¹, Mengchao Zhang¹, Jinting Wu¹ and Bowen Zhang^{2,*}

¹Samsung R&D Institute China-Beijing

²Beijing University of Technology

{zhang.fan, mei.tu, mch.zhang, jinting01.wu}@samsung.com

Abstract

The accurate translation of domain-specific terminology remains a critical bottleneck for ensuring the fidelity and professionalism of machine translation systems. This paper presents our submission to the WMT26 Terminology Translation Shared Task: the TaT (Terminology-aware Translation) system. Built upon a structured, pipelined architecture, TaT integrates five specialized components designed to seamlessly ingest, process, and translate ambiguous textual inputs while strictly adhering to the targeted terminology dictionary. Specifically, the framework comprises a sentence splitter, a terminology retriever, a lexical disambiguator, a baseline translator, and a dedicated terminology extractor tailored for Track 2 constraints. Both internal benchmarking and ablation studies demonstrate that our decoupled pipeline delivers robust performance, successfully meeting design expectations and substantially mitigating terminology omission and misalignment errors in domain-specific translations.

1 Introduction

Domain-specific machine translation requires strict adherence to specialized lexicons to ensure technical accuracy and cross-lingual consistency. While contemporary Large Language Models (LLMs) and advanced neural machine translation (NMT) architectures have dramatically elevated general translation fluency, they frequently falter when dealing with rare, highly technical, or polysemous technical terms. In production scenarios, a single mis-translated concept can compromise the entire document’s integrity, potentially leading to critical misunderstandings in specialized fields. Consequently, effectively constraining translation outputs to honor predefined terminology lists, while avoiding target-

side omissions or misalignments, remains a vital area of empirical study (Semenov et al., 2025).

To rigorously address these limitations, we developed the TaT (Terminology-aware Translation) system, which served as our official submission to the WMT26 Terminology Translation Shared Task. Departing from purely monolithic end-to-end translation frameworks, which often struggle to balance fluid phrasing with rigid lexical constraints, TaT leverages a decoupled, pipelined architecture. The system orchestrates five foundational components in a sequential manner: a sentence splitter to normalize sequence lengths and define localized windows, a terminology retriever to surface domain-specific matches from the target dictionary, a lexical disambiguator to resolve terminology polysemy by evaluating contextual target candidates, a core translator executing the final synthesis, and a specialized terminology extractor explicitly engineered to fulfill the requirements of the Track 2 benchmark by constructing a missing terminology dictionary directly from sample bitexts.

Our internal evaluation and empirical analysis validate the efficacy of this modular design. By isolating the retrieval and disambiguation phases prior to decoding, the TaT framework minimizes the processing load on the translation engine, ensuring highly reliable constraint adherence. These comparative self-evaluations primarily serve to identify the optimal strategy configuration for our pipeline. Across multiple tested tracks and evaluation subsets, the resulting core configuration achieves marked improvements in terminology alignment accuracy without degrading baseline linguistic fluency.

2 System Description

Unlike previous editions of the Terminology Translation task (Semenov et al., 2023, 2025), this iteration introduces a strict document-level setup for

*Work done during an internship at Samsung R&D Institute China-Beijing.

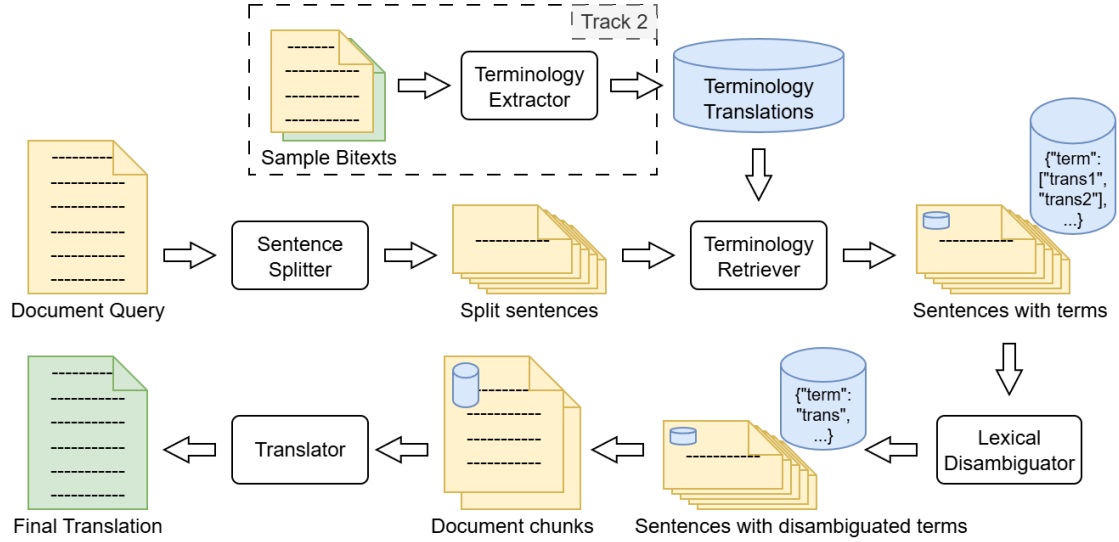


Figure 1: Overall pipeline of the proposed TaT system, consisting of five core components: the sentence splitter, terminology retriever, lexical disambiguator, translator, and terminology extractor (for Track 2).

both evaluation tracks. Furthermore, Track 2 does not provide a predefined terminology database, requiring participants to either utilize the provided sample bitexts directly or extract a structured terminology dictionary from them. The architecture coordinates five functional components: a sentence splitter, a terminology retriever, a lexical disambiguator, a core translator, and a specialized terminology extractor for Track 2.

2.1 Sentence splitter

The sentence splitter processes raw documents into segments through four cascading phases: (1) Paragraph boundary detection, which identifies layout structures to maintain document integrity; (2) Multilingual sentence tokenization, offering native support for parsing syntax variations in Chinese, English, and Spanish; (3) Dynamic chunk merging, which groups consecutive brief sentences to fulfill configurable minimum length thresholds; (4) Hierarchical splitting, which applies fallback syntactic rules to decompose exceptionally long sentences at linguistically viable boundaries.

Additionally, the module implements a non-destructive token tracking mechanism. By logging exact separator indices and structural metadata during splitting, the system guarantees the lossless reconstruction of the original document formatting once translation is complete.

2.2 Terminology retriever

The terminology retriever extracts relevant terms from a dictionary using a hybrid approach that

combines fuzzy matching with LLM verification. First, the module applies fuzzy matching to identify all potential candidate terms within the segment. While this step ensures broad coverage, fuzzy matching inevitably introduces noise and false positives due to word overlapping or morphological variations. Second, to filter out these invalid candidates, the module passes the text and the matched terms to an LLM for validation.

As shown in Appendix A (Table 3), we use a dedicated verification prompt that forces the LLM to confirm if the concepts are genuinely present in the context. The LLM filters out mismatched pairs and outputs a clean JSON array of valid terms. The final output consists of verified term-translation pairs, which are then passed to the downstream tasks to ensure consistent and accurate rendering.

2.3 Lexical disambiguator

The terminology lexical disambiguator resolves translation ambiguities when a source term maps to multiple candidate translations in the dictionary. To ensure document-level consistency, the module evaluates these polysemous terms within their global contexts through a two-step process.

First, the module identifies ambiguous terms that appear across different sentences and aggregates their corresponding target candidates. Second, the system leverages an LLM to perform contextual selection. The LLM is provided with the complete sentence-level contexts alongside all translation candidates for each ambiguous term. Its task is to determine and select the single most appropriate

Table 1: Prompt template for document-level translation with terminology constraints.

<pre> # System Prompt You are a professional translator from {src_lang} to {tgt_lang}. Your task is to translate documents with the highest quality standards. Translation Quality Standards: 1. Accuracy: Faithfully convey the original meaning without adding, omitting, or distorting content; 2. Fluency: Produce natural, readable text that follows target language grammar and style conventions ; 3. Consistency: Use consistent terminology and style throughout the document; 4. Cultural Appropriateness: Adapt expressions appropriately for the target culture while preserving the original intent; 5. Professional Standards: Maintain the tone, register, and formatting appropriate for the document type; Output ONLY the translation. Do not include explanations, notes, or commentary. # User Prompt ### Translation Instructions 1. Translate the source document from {src_lang} to {tgt_lang} ### Required Terminology - {src_term_1}: {tgt_translation_1} - {src_term_2}: {tgt_translation_2} ### Critical Format Requirement The source text below contains <SEP> markers that represent line breaks (\n). You MUST preserve the EXACT number and position of all <SEP> markers in your translation. ### Source Document {document_with_sep_markers} </pre>
--

target translation that fits the specific local context while maintaining global consistency.

The output provides a refined dictionary where each source term maps to exactly one contextually correct translation. This elimination of multi-choice ambiguity guarantees that the downstream translator receives deterministic constraints.

The complete prompt template used for terminology disambiguation is provided in Appendix A (Table 4).

2.4 Translator

The translator implements a document-level translation strategy to preserve context while avoiding performance degradation caused by long sequences.

To maintain document coherence, the module processes the entire document in a single request whenever possible. However, excessively long contexts can dilute the model’s attention, leading to omissions or hallucinations. To mitigate this, the module applies a paragraph-based chunking strategy for documents that exceed a configured length threshold. The text is split at paragraph boundaries into smaller chunks, balancing contextual completeness with manageable sequence lengths.

To maintain paragraph structures during this pro-

cess, all newline characters in the input chunk are replaced with <SEP> markers before being sent to the LLM. The system instructs the LLM to preserve these markers in the target output.

In practice, when handling long outputs, LLMs frequently introduce unexpected text-internal line breaks, which can cause paragraph misalignment between the input and output. To resolve this formatting error, the module applies a post-processing step: it removes all accidental line breaks generated by the LLM, and then uniformly replaces the verified <SEP> markers back into standard newlines. This design ensures that the translated paragraphs align perfectly with the original layout without losing contextual quality.

The complete prompt template used for the document-level translation with terminology constraints is provided in Table 1.

2.5 Terminology extractor

Unlike Track 1, Track 2 only provides sample bitexts without a predefined terminology dictionary. To bridge this gap, our terminology extractor automatically constructs a structured dictionary from the sample bitexts, allowing the subsequent processing pipeline to align entirely with the Track 1

Table 2: Self-evaluation results of different translation strategies.

Translation Granularity	Base Model	track1			track2	
		proper	random	noterm	sample	noterm
Sentence-level	Deepseek-V4-Flash	-14.21	-94.83	-56.89	-33.50	-44.42
Document-level (Chunked)	Deepseek-V4-Flash	-13.44	-69.54	-46.95	-16.26	-45.07
	Gemini-3.6-Flash	-9.07	-67.71	-41.29	-9.64	-41.24
	Gemini-3-Pro-Preview	-6.67	-79.79	-42.48	-4.02	-22.45

framework.

During preprocessing, the module dynamically adapts to the text granularity of different language tracks: it utilizes BertAlign (Liu and Zhu, 2023) to perform fine-grained sentence alignment for the paragraph-level zh-en data, while directly utilizing the native short segments for tracks like en-es and en-pl to avoid redundant parsing. Next, an LLM is used to extract domain-specific terminology from these segments. The extraction prompt targets technical nouns, multi-word phrases, specialized verbs, and domain adjectives, while explicitly filtering out generic vocabulary (see Table 5 in Appendix A).

To prevent semantic drift, a strict cross-lingual matching criterion is enforced: a source term candidate is retained only if its target equivalent carries the exact same domain-specific meaning within that parallel pair. Finally, all extracted raw terms are lemmatized to their dictionary base forms, such as singular nouns and infinitive verbs, before being deduplicated and merged into a unified terminology dictionary where each source term maps to its verified target translations.

3 Self-Evaluation

To optimize system performance, we conducted internal evaluations on key strategy choices, employing DeepSeek-V4-Flash (Xu et al., 2026) as the base LLM for all modular pipeline components except the core translator. Guided by the findings of the WMT25 general machine translation task, where Gemini-2.5 demonstrated superior performance across almost all language tracks (Kocmi et al., 2025), we chose to benchmark leading frontier architectures for our core translator. Specifically, our translation experiments evaluate three candidate models: DeepSeek-V4-Flash (Xu et al., 2026), Gemini-3.6-Flash, and Gemini-3-Pro-Preview (Gemini Team et al., 2023).

3.1 Evaluation Metrics

We employ an LLM-based Multidimensional Quality Metrics (MQM) framework (Kocmi and Federmann, 2023) to evaluate translation quality. Source texts and translations are concatenated in their original sequence and processed by an evaluator LLM with document-level context. The LLM identifies translation errors across six standard categories (Accuracy, Fluency, Terminology, Style, Locale, and Other) and assigns severity levels: Critical, Major, and Minor (weighted as -25, -5, and -1, respectively). The final MQM score is programmatically calculated as a cumulative penalty, where scores closer to 0 indicate higher quality.

Since a reference terminology dictionary is unavailable during testing, the evaluator LLM relies entirely on its internal knowledge to judge terminology correctness. While this approach may introduce some deviations from human judgment and cannot reliably distinguish marginal score differences, it provides a sufficient and reproducible signal for guiding internal strategy selection when score gaps are substantial.

3.2 Results

As shown in Table 2, the self-evaluation scores reveal clear trends regarding the impact of translation granularity and base models. A direct comparison between the sentence-level and document-level strategies using the same DeepSeek model demonstrates that providing richer contextual information yields substantial translation gains across nearly all tracks. This suggests that a broader document-level context is critical for improving overall translation accuracy and helping models resolve complex lexical ambiguities. Furthermore, we observe noticeable performance variations among different base models under the chunked-document setup, where Gemini-3-Pro-Preview delivers the most robust overall performance and secures the highest MQM scores in three out of five test categories.

The final competition results can be found in the official report (Charkiewicz et al., 2026).

4 Conclusion

This paper presented the TaT (Terminology-aware Translation) system for the WMT26 Terminology Translation Shared Task. By decoupling the workflow into five sequential stages—sentence splitting, terminology retrieval, lexical disambiguation, core translation, and Track 2 terminology extraction—our framework effectively integrates terminology dictionary constraints into the translation loop. Internal evaluations confirm that utilizing document-level context via paragraph-based chunking significantly reduces omission and alignment errors compared to sentence-level baselines, with Gemini-3-Pro-Preview yielding the highest overall translation quality.

Limitations

While the TaT system delivers robust translation performance, its multi-stage framework introduces certain practical limitations. The architecture relies heavily on Large Language Models (LLMs) across almost all core components. Although we intentionally mitigate computational overhead by deploying smaller, lightweight LLMs for upstream tasks like retrieval and disambiguation, the pipeline inevitably accumulates inference latency compared to traditional single-pass end-to-end NMT models. This latency overhead could present challenges in production environments with strict real-time response constraints.

Meanwhile, due to the tight development schedule of the shared task, several hyperparameter choices and prompt configurations were determined empirically. While these configurations fully met our immediate design expectations, a more exhaustive hyperparameter sweep might yield better-optimized settings. We plan to address pipeline acceleration and systematic optimization in future iterations of the framework.

References

Adrian Charkiewicz, Pinzhen Chen, Thierry Etchegoyhen, Harritxu Gete Ugarte, Kamil Guttman, Xu Huang, David Ponce, Artur Nowakowski, Frédéric Odernatt, 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.

Gemini Team, Rohan Reid, Mariko Marone, Valerii Teplyashin, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, et al. 2025. Findings of the wmt25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the tenth conference on machine translation*, pages 355–413.

Tom Kocmi and Christian Federmann. 2023. Gemba-mqm: Detecting translation quality error spans with gpt-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775.

Lei Liu and Min Zhu. 2023. Bertalign: Improved word embedding-based sentence alignment for chinese–english parallel corpora of literary texts. *Digital Scholarship in the Humanities*, 38(2):621–634.

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.

Kirill Semenov, Vilém Zouhar, Tom Kocmi, Dongdong Zhang, Wangchunshu Zhou, and Yuchen Eleanor Jiang. 2023. Findings of the wmt 2023 shared task on machine translation with terminologies. In *Proceedings of the Eighth Conference on Machine Translation*, pages 663–671.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*.

A Prompt Templates

This appendix contains all the prompt templates used throughout our TaT system. For clarity and reproducibility, we consolidate here every prompt employed in our pipeline components, including the terminology retriever (Table 3), the lexical disambiguator (Table 4), and the terminology extractor (Table 5).

Table 3: The verification prompt template used in the Terminology Retriever.

```
# System Prompt
You are a terminology verification expert. Your task is to verify whether the candidate terms are genuinely present in the query text. A term is valid only if the query text actually contains or clearly implies that concept.

# User Prompt
[Query Text]
{query_text}

[Candidate Terms to Verify]
- {term_1}
- {term_2}
- ...
- {term_n}

Verify which terms are genuinely present in the query text. Output ONLY a JSON array of valid term names.
```

Table 4: Prompt template for terminology disambiguation.

```
# System Prompt
You are an expert translator specializing in {src_lang} to {tgt_lang} translation. Your task is to select the most appropriate translation for an ambiguous term that appears in multiple sentences.

# User Prompt
[Ambiguous Term]
"{src_term}"

[Candidate Translations]
["{candidate_1}", "{candidate_2}", ..., "{candidate_n}"]

[Contexts - {N} Sentences containing the term]
1. {context_sentence_1}
2. {context_sentence_2}
...
{N}. {context_sentence_N}

Select ONE translation that is most appropriate for ALL contexts. The same translation will be used consistently in all sentences.
Output ONLY the selected translation (a single word or phrase).
```

Table 5: Prompt template for terminology extraction from sample bitexts.

```
# System Prompt
You are a professional terminology extraction expert [specializing in {domain} domain] for {src_lang}
to {tgt_lang} translation.
Your expertise lies in precisely identifying technical terms, domain-specific concepts, and
culturally unique expressions from parallel corpora.

# User Prompt
### Rules (Read First)

**What to Extract:**
1. Technical Nouns & Noun Phrases: Core domain concepts, multi-word terms;
2. Domain-Specific Verbs: Action words SPECIFIC to the domain (e.g., 'chamfer', 'broach');
3. Technical Adjectives: Only when part of a fixed technical term;

**What to SKIP:**
1. Common Verbs: use, make, do, have, be, allow, enable, provide, perform, etc.;
2. Generic Nouns: system, machine, part, component, process, method, way, etc.;
3. No clear translation: If target paraphrases or omits, SKIP it;
4. Length: Skip words <3 or >50 characters;

**Semantic Alignment Check:**
1. Does target have a DIRECT lexical equivalent? (not paraphrase)
2. Is meaning preserved exactly? (no semantic drift)
3. Can you point to EXACT word(s) in target?
-> If ANY fails, SKIP the term

**Lemmatization:**
- Nouns: singular, nominative
- Verbs: infinitive form
- Adjectives: remove gender/number markers

### Output Format
{"terms": {"[src_term]": "[tgt_term]"}}

### Input Sentences
SOURCE: {source_sentence}
TARGET: {target_sentence}

Output ONLY valid JSON.
```