

# Tilde’s Submission for the WMT2026 Shared Task on Terminology Translation

Mārcis Pinnis<sup>1,2</sup>, Martins Kronis<sup>1</sup>, Emīls Grims<sup>1</sup>,  
Ervīns Jakovelis<sup>1</sup>, Roberts Rozis<sup>1</sup>, Toms Bergmanis<sup>1,2</sup>

<sup>1</sup>Tilde, Vienības gatve 75a, Rīga, LV-1004, Latvia

<sup>2</sup>Faculty of Science and Technology, University of Latvia

Correspondence: [firstname.lastname@tilde.ai](mailto:firstname.lastname@tilde.ai)

## Abstract

This paper describes Tilde’s submissions to the WMT 2026 Shared Task on Terminology. For both subtasks, we use TildeOpen-30B, which we instruction-tune for translation using terminology glossaries and translation memories, and we detail the construction of the training data augmented with these resources. For Task 1, we compare three term recognition methods aimed at reducing large term collections to subsets relevant for translation: fuzzy search, stemming with exact-match search, and stemming with fuzzy search. For Task 2, we compare extracting a glossary from the provided translation memory and integrating it in context against using retrieved translation-memory entries directly as few-shot examples. As the shared task provides neither reference translations nor a development set, we validate our design decisions with an LLM-as-a-judge protocol. In-domain term collections reduce translation errors by up to 22% relative to translating without terminology, whereas out-of-domain collections yield only marginal gains, and glossaries extracted from the translation memory outperform few-shot translation with retrieved entries.

## 1 Introduction

Accurate and consistent use of domain terminology is essential in technical translation. Even an otherwise fluent translation may alter the meaning if specialised terms are mistranslated or used inconsistently. Such errors can lead to misunderstandings and, in sensitive domains, potentially serious consequences. Maintaining consistency is a particular challenge in document-level translation. However, translation is rarely performed on whole documents because the quadratic complexity of the transformer’s attention mechanism makes whole-document translation computationally impractical for large documents and webpages. Translating in chunks restores tractability but can reintroduce

inconsistencies in the translation of terms, brand language, and named entities. Translation informed by user-provided terminology, translation memories, or even previously machine-translated paragraphs from the same document can help reduce these inconsistencies.

Terminology integration in machine translation has been well studied for encoder–decoder neural machine translation models. For instance, (Dinu et al., 2019) proposed annotating source sentences with target-side term constraints during training, enabling terminology-aware neural machine translation models to accept term translation suggestions at inference time. (Bergmanis and Pinnis, 2021b) extended this approach with target-language lemma annotations, making it suitable for morphologically rich target languages. With the emergence of large language models, researchers have explored terminology-constrained post-editing of machine-translated hypotheses (Bogoychev and Chen, 2023; Moslem et al., 2023b), automatic bilingual terminology extraction (Moslem et al., 2023a), parameter-efficient instruction tuning on in-domain, terminology-augmented data (Rios, 2025), and direct terminology-aware translation through in-context learning (Moslem et al., 2023a; Lackner et al., 2025). The latter is particularly attractive in practice because it requires no model retraining and allows terminology to be updated dynamically during inference.

The WMT 2026 Shared Task on Terminology enables evaluation of methods that integrate terminology and translation memories into machine translation systems. We participate in both tasks with TildeOpen, a 30B large language model instruction-tuned for several translation scenarios, including terminology-aware translation through in-context learning and translation-memory integration through few-shot prompting. In Task 1, relevant entries are selected from the provided term collection and included in the translation prompt. In

Task 2, we investigate both terminology extracted from the provided translation memory and the direct use of retrieved translation-memory entries as few-shot examples. In both tasks, we participate with translations for the English-Polish translation direction.

In this paper, we describe the construction of the terminology- and translation-memory-augmented training data and the fine-tuning of the base TildeOpen model (Bergmanis et al., 2026). We further document the translation pipelines used for Tasks 1 and 2, including the evaluated term-recognition methods and the strategy for integrating translation memory.

## 2 Instruction-Tuning of TildeOpen

We follow Bergmanis et al. (2026) in adapting the TildeOpen-30B foundation model for general translation and introduce two additional tasks: translation with terminology glossaries and translation with translation memories. Below, we first describe the general translation data, then the other two data types, and finally the fine-tuning procedure.

### 2.1 General Parallel Data

The instruction-tuning corpus includes translation directions for all 34 languages supported by TildeOpen. It is compiled from publicly available sources: OpenSubtitles (Lison and Tiedemann, 2016), TED talks (Salesky et al., 2021) from OPUS (Tiedemann, 2012), and newly recrawled EUR-Lex<sup>1</sup> parallel data. We use document-level versions of these datasets and select consecutive segments so that each source and target text contains no more than 3,000 space-separated words. Additionally, we collect monolingual news articles in all supported languages from HPLT-v2 and back-translate them into English using the Gemma 2 27B IT model (Gemma Team et al., 2024). Note that we do not use domain-specific data to fine-tune TildeOpen. Domain adaptation will be solely achieved during inference through in-context learning.

We apply several filters to ensure data quality. For OpenSubtitles and TED, we retain only segments with average COMET quality-estimation scores at least 25% higher than the average FLORES dev-set score for each language pair (Rei et al., 2022; NLLB Team, 2024). LangID filtering removes texts with low language confidence or identi-

cal source and target segments. For back-translated data, we verify that the generated English maintains paragraph structure, numerals, and terminal punctuation. Each sentence or text appears only once as a translation target across all language pairs.

From the filtered data, we sample 2 million documents, balancing target-language tokens so that each language serves as the target about 3% of the time. Parallel data is formatted using simple English instructions: “{source text}\n\nTranslate the given text from {source language} to {target language}. {target text}”. The final dataset contains approximately 462 million tokens: 65% back-translated news, 25% OpenSubtitles, 5% TED, and 5% EUR-Lex. The data is further wrapped in a chat template so that the trained model supports the prompt structure shown in Appendix B and Figure 2.

### 2.2 Terminology Data

To train the model to translate text using a user-provided term glossary, we created parallel training data in which sentence pairs are annotated with bilingual terminology suggestions. Producing such data requires (i) a parallel corpus, (ii) bilingual term collections, and (iii) a procedure for identifying term occurrences in both the source and the target side of the corpus. We describe each in turn.

**Parallel Corpus and Term Collection** We build on our previous work and reuse the parallel data prepared for TildeOpen, as described in Bergmanis et al. (2026). Among other corpora, this data includes EUR-Lex, TED Talks (Cettolo et al., 2012), and OpenSubtitles (Lison and Tiedemann, 2016). The European legal data includes terms from the Interactive Terminology for Europe (IATE) database (Zorrilla-Agut and Fontenelle, 2019), while the two conversational corpora are likely to include entries from the Microsoft Terminology Collection.

**Text and Term Normalisation** Matching glossary terms against running text requires abstracting away from surface form variation. We therefore normalised terms and corpus text with a shared pipeline that combined lemmatisation using the Stanza toolkit (Qi et al., 2020) and stemming with our in-house rule-based stemmer, originally based on the Snowball stemmer (Porter, 2001).

**Term Indexing** For efficient lookup, we organised each term collection in a trie structure over lemmatised words: terms were split into words,

<sup>1</sup><https://eur-lex.europa.eu/>

and each term was inserted in a left-to-right pass, so that the first level of the trie holds the first words of all terms, the second level – the second words, and so on. Terms sharing lemmatised prefixes thus collapse into common paths, keeping the trie compact and making term search fast. Importantly, every term carries an identifier that links its equivalents across languages. Terms with multiple meanings but matching surface forms share the same source identifier, allowing us to additionally build a translation alternative lookup. We later use this to provide ambiguous terms with several meanings and translation alternatives.

**Term Matching and Selection** We processed each document paragraph by paragraph, scanning the source and target sides independently. In each paragraph, we matched terms greedily from left to right, extending a match word by word as long as the running word sequence remained on a path in the trie. We then retained only those candidates whose identifiers were present on both the source and target sides.

The resulting term annotations sometimes contained overlapping terms. We selected longer multiword terms over single-word terms so that the model learns to use them. When terms from different collections overlapped, IATE was given priority over the Microsoft Terminology Collection; remaining ties were broken at random. Finally, we deduplicated the selected source–target term pairs so that no annotation appears twice in a training example. As above, the data is processed using a chat template so that the trained model supports the prompt structure shown in Appendix B and Figure 3.

### 2.3 Translation Memory Data

To teach the model to translate using translation memory examples, we created a dataset in which training samples are augmented with example translations. We use the EUR-Lex corpus to obtain document-level and corpus-level translation examples. For document-level translation examples, we randomly selected up to 20 examples from either the beginning or the end of the document. To obtain corpus-level examples, we created a translation memory index and queried it to retrieve sentences similar to the text being translated.

**Translation Memory Index** To implement the translation memory, we use Apache Lucene<sup>2</sup>. Instead of documents, we indexed aligned source–target sentence pairs, so that querying the index with a source sentence returns lexically similar sentences together with their translations. We indexed our entire EUR-Lex dataset, amounting to almost 18 million sentence pairs. For each source sentence, we queried the index for up to 1k candidate examples and kept only those whose Levenshtein similarity ratio to the query, computed on the source side, exceeded 0.7. To expose the model to exact matches, we sampled up to 20% of examples from the document itself for 10% of the documents. The data was further formatted so that the trained model supports the prompt structure given in Appendix B and Figure 4.

### 2.4 Finetuning

We instruction-tune the model in a supervised manner using a custom fork of GPT-NeoX framework<sup>3</sup> (Andonian et al., 2023) on 512 AMD MI250x GPUs of the LUMI supercomputer. The model is trained in 600 updates for a total of  $\sim 2.4\text{B}$  tokens. Our starting learning rate of  $8 \cdot 10^{-6}$  matches the final cool-down value in the foundation model’s pre-training schedule and is further cooled to 1% using 1-sqrt cooldown schedule.

Automatic evaluation results on standard test sets are given in Table 1. The results show that the model achieves overall translation quality that is competitive with current open-weight models of similar size.

## 3 Task 1: Translation with Terminology

Task 1 analyses three translation scenarios: translation without a term collection (No terms), translation with an in-domain term collection (Proper), and translation with an out-of-domain or mixed-domain collection that may contain terms among other words and phrases (Random). We participate in Task 1 with the TildeOpen translation model, which has been instruction-tuned for translation with terminology, along with other capabilities.

To translate the evaluation data, we first perform term recognition to identify term entries present in each paragraph. Then, for each paragraph, we form a translation prompt that, for the No terms scenario, skips terms (an example prompt is provided

<sup>2</sup><https://lucene.apache.org>

<sup>3</sup><https://github.com/tilde-nlp/llm-gpt-neox>

| Test set           | ChrF++       |              |          | COMET         |               |          |
|--------------------|--------------|--------------|----------|---------------|---------------|----------|
|                    | TildeOpen    | Gemma 4      | Qwen 3.5 | TildeOpen     | Gemma 4       | Qwen 3.5 |
| FLORES-200 devtest | 49.83        | <b>51.06</b> | 48.83    | <b>0.8990</b> | 0.8906        | 0.8917   |
| NTEU               | <b>71.60</b> | 65.32        | 61.34    | <b>0.9339</b> | 0.9221        | 0.9126   |
| WMT24++            | 47.22        | <b>49.21</b> | 45.42    | 0.8328        | <b>0.8376</b> | 0.8251   |
| NTREX              | 53.03        | <b>54.44</b> | 50.48    | <b>0.8805</b> | 0.8738        | 0.8659   |

Table 1: ChrF++ and COMET scores across test sets for the English→Polish translation direction. We compare TildeOpen 30B, Gemma 4 31B, and Qwen 3.5 27B. FLORES-200 (Costa-Jussà et al., 2022) and NTREX (Federmann et al., 2022) test translation of news articles, NTEU (García-Martínez et al., 2021) – legal (EU policy) documents, and WMT24++ (Deutsch et al., 2025) represents a mix of news, fiction, social media communications, and speech transcripts. The best result for each test set and metric is shown in bold.

in Appendix B; Figure 2), and for the other scenarios includes the recognised terms (Appendix B; Figure 3). Finally, we generate a translation with the instruction-tuned TildeOpen model by setting the temperature and frequency penalty to 0.

For term recognition, we analyse three methods:

- **Fuzzy Search:** We lowercase both the translatable text and source terms, followed by a three-level term recognition method. First, we try to match term occurrences exactly. Then, we check whether a prefix of the term matches a token found in the text (or vice versa), with a minimum overlap of 0.7. Finally, we use the rapidfuzz library to calculate a fuzzy match score between the term and tokens in the text. If the score is 90% or higher, we accept the term as a match. The first and third methods are applied for both single-word and multi-word terms. The second method is applied only for single-word terms. This method can handle inflectional changes and typographical errors; however, it is prohibitively slow in practice with large term collections (see Appendix A for performance analysis).
- **Stemming and Exact Match Search:** First, both the translatable text and source terms from the term collection are lowercased and stemmed using a stemming tool<sup>4</sup>. Then we search for term occurrences in the text using a trie lookup algorithm. Stems are matched exactly. This method handles most inflectional

changes and is efficient for practical application with large (1M+ entries) term collections; however, it does not address changes that occur at the root of a word and fails with typographical errors.

- **Stemming and Fuzzy Search:** Here we combine the first two ideas. First, all text and terms are stemmed. Then we build a trie from the term collection to support multi-word term recognition, as in the second method. Finally, matching at each trie node is performed using fuzzy search rather than an exact hash table lookup. For this method, we perform fuzzy search using the SymSpell library<sup>5</sup>. This method can handle inflectional changes and typographical errors, and, thanks to the efficient fuzzy lookup provided by SymSpell, it can be applied in production workflows.

The methods are designed to reduce the original term collection to a manageable subset for each paragraph, rather than recognising only truly valid term entries. That is, we aim for high recall rather than high precision, because the translation model can handle disambiguation.

Since the organisers did not provide a development set with reference translations and terminology for each paragraph, on which we could analyse the model’s performance across different translation scenarios, we performed an automatic LLM-as-a-judge evaluation. For each paragraph, we prompted the GPT-5.6-luna<sup>6</sup> model to generate a list of all minor and major errors that could be identified in the hypothesis, using the given source

<sup>4</sup>For English, we use a slightly optimised C# port of the Porter2 stemming algorithm from Snowball(<https://snowballstem.org/>). The exact source code is available on GitHub (<https://github.com/pmarcis/dict-filtering/blob/master/FilterGizaDictionary/Stemmers.cs>).

<sup>5</sup><https://github.com/wolfgarbe/SymSpell>

<sup>6</sup><https://developers.openai.com/api/docs/models/gpt-5.6-luna>

text and a term collection as context (an example of the evaluation prompt is given in Appendix B; Figure 6). Since each term recognition method recognises only a subset of possible terms, we unified all subsets for each paragraph and used the resulting set for evaluation. This does not ensure that all valid term entries are provided as context; however, this ensures that all term recognition scenarios are evaluated using the same term collection.

The results of the LLM-as-a-judge evaluation are provided in Table 2. Across both domains, using the Proper terminology for term recognition consistently reduces translation errors compared with translating without terminology, whereas the Random terminology yields only marginal or inconsistent improvements. Differences among the three term recognition methods are relatively small, although Fuzzy search generally achieves the strongest results with the Proper terminology, while the stemmer-based methods show comparable but slightly lower results. For the shared task, we made two submissions: Fuzzy search as our primary submission (named `to+fuzzy_tr`), as it achieved the highest error reduction for the proper term collection, and Stemming and fuzzy search as our secondary submission (named `to+stem_fuzzy_tr`).

## 4 Task 2: Translation with a Translation Memory

Task 2 analyses two translation scenarios: translation without a term collection (No terms) and translation with a translation memory (translation memory) that features in-domain terminology. We participate in Task 2 with the TildeOpen translation model. However, we split the translation memory scenario into two further scenarios – one where we automatically construct a term collection from the provided translation memory and translate the documents using terminology integration through in-context learning (similarly to Task 1), and the other where we perform translation memory integration using few-shot translation.

To create the term collections from the translation memories for the two domains, we use Gemma 4 31B. The model is prompted to extract term pairs in canonical forms and surface forms from translation memory entries. An exact prompt example is provided in Appendix B (Figure 5). The extraction process resulted in term collections of 418 and 956 terms for the medical and mechanical

engineering domains, respectively.

For the No terms scenario and the terminology integration scenario, we use the same methods for translation generation as described in Section 3. For the translation memory integration scenario, we split each translatable paragraph into sentences and extract the N most similar sentences from the translation memory (where N equals 5, 10, or 15). Then, in a round-robin manner, we select the top-most similar candidates for each sentence till N candidates are selected. If N is larger than the number of sentences in the paragraph, some sentences will have more than one translation memory entry assigned. If N is smaller, then some sentences will lack assigned translation memory entries. We use Levenshtein distance as the similarity metric. Once translation memory entries are selected, we form a few-shot translation prompt (see Appendix B; Figure 4 for an example), and generate the output.

Similar to Task 1, we perform an LLM-as-a-judge evaluation to identify which scenario yields better translations. However, this time we do not have access to a reference term collection. Therefore, we prompt the GPT-5.6-luna LLM with only the source text, the hypothesis translation, and the domain name (see Appendix B; Figure 7 for an example). Given the limited context, we expect the evaluations to be noisier than those for Task 1.

The results of the LLM-as-a-judge evaluation are provided in Table 3. Across both domains, incorporating information from the translation memory consistently reduces translation errors compared with translating without additional context. However, automatically extracting terminology from the translation memory and integrating it as an in-context glossary generally yields larger improvements than few-shot translation using retrieved translation memory examples. That being said, the results may indicate a capability deficiency in the instruction-tuned TildeOpen model that we will have to address in a future run. For the shared task, we made three submissions - Top 15 candidates few-shot translations (named - `to+tm_top15`) as our primary submission, and Top 5 candidates few-shot translations (named - `to+tm_top5`) and Fuzzy search (named - `to+g4terms`) as our secondary submissions.

## 5 Official Results

Table 4 summarises the official results of our English→Polish submissions to the WMT 2026

| Domain     | Terms    | Term recognition method | Errors |       |       | Reduction (%)<br>vs. no terms |
|------------|----------|-------------------------|--------|-------|-------|-------------------------------|
|            |          |                         | Minor  | Major | Total |                               |
| Medical    | No terms |                         | 4.2    | 1.0   | 5.2   |                               |
|            | Proper   | Fuzzy search            | 3.8    | 0.7   | 4.5   | 13.5%                         |
|            |          | Stemmer + exact search  | 4.1    | 0.8   | 4.9   | 5.8%                          |
|            |          | Stemmer + fuzzy search  | 4.0    | 0.6   | 4.6   | 11.5%                         |
|            | Random   | Fuzzy search            | 4.4    | 0.8   | 5.2   | 0.0%                          |
|            |          | Stemmer + exact search  | 3.8    | 1.2   | 5.0   | 3.8%                          |
|            |          | Stemmer + fuzzy search  | 4.0    | 1.0   | 5.0   | 3.8%                          |
| Mech. eng. | No terms |                         | 4.6    | 3.0   | 7.6   |                               |
|            | Proper   | Fuzzy search            | 4.3    | 1.6   | 5.9   | 22.4%                         |
|            |          | Stemmer + exact search  | 4.4    | 1.9   | 6.3   | 17.1%                         |
|            |          | Stemmer + fuzzy search  | 4.5    | 1.6   | 6.1   | 19.7%                         |
|            | Random   | Fuzzy search            | 4.8    | 2.6   | 7.4   | 2.6%                          |
|            |          | Stemmer + exact search  | 4.6    | 2.4   | 7.0   | 7.9%                          |
|            |          | Stemmer + fuzzy search  | 4.4    | 2.5   | 6.9   | 9.2%                          |

Table 2: LLM-as-a-judge evaluation results for Task 1 (English→Polish translation direction).

Terminology Translation Task (Charkiewicz et al., 2026). For Task 1, fuzzy search achieves the highest scores among our submissions on all four metrics. It outperforms stemming and fuzzy search by 2.8 percentage points on the term success metric; however, the differences in paragraph-level chrF++ and COMET are comparatively small.

For Task 2, no single configuration performs best across all metrics. Few-shot translation with 15 retrieved translation-memory examples achieves the highest term success and paragraph-level ChrF++, while using five examples yields the highest COMET score. The glossary-based configuration achieves the highest lemmatised term success. Both few-shot configurations outperform glossary integration on ChrF++ and COMET, in contrast to the overall preference for glossary integration in our LLM-as-a-judge evaluation.

The results suggest that term success improves with in-context learning via a glossary or a translation memory by up to 28.7 percentage points (e.g., from 52.1% without terms to 80.8% with proper terms and fuzzy search in Task 1). However, the results also show that there is still room for improvement, as in 19.2% of cases the model decided not to follow the glossary entries and mistranslated. The results indicate a need for deeper analysis of the mistranslation cases. However, we leave this for future work.

## 6 Conclusions and Discussion

We described Tilde’s submissions to the WMT 2026 Shared Task on Terminology. For both tasks we used TildeOpen-30B, instruction-tuned for terminology-aware translation through in-context glossaries and for translation-memory integration through few-shot prompting, and we detailed the construction of the terminology- and translation-memory-augmented data used for this instruction-tuning. For Task 1, we compared three term recognition methods – fuzzy search, stemming with exact search, and stemming with fuzzy search – as means of reducing a large term collection to a per-paragraph subset. For Task 2, we compared two ways of exploiting the provided translation memory: extracting a glossary from it and integrating that glossary in context, or using retrieved translation-memory entries directly as few-shot examples.

Lacking a development set with reference translations, we evaluated all scenarios with an LLM-as-a-judge protocol. In-domain term collections reduced errors by 13.5–22.4% relative to translating without terminology, while out-of-domain collections yielded only marginal gains. Differences among the three recognition methods varied, but pointed towards a preference for methods that have high recall. In Task 2, glossaries extracted from the translation memory outperformed few-

| Domain     | Scenario  | Configuration          | Errors |       |       | Reduction (%)<br>vs. no terms |
|------------|-----------|------------------------|--------|-------|-------|-------------------------------|
|            |           |                        | Minor  | Major | Total |                               |
| Medical    | No terms  |                        | 4.2    | 0.8   | 5.0   |                               |
|            | Term int. | Fuzzy search           | 3.5    | 0.7   | 4.2   | 16.0%                         |
|            |           | Stemmer + exact search | 3.1    | 0.6   | 3.7   | 26.0%                         |
|            |           | Stemmer + fuzzy search | 3.8    | 0.6   | 4.4   | 12.0%                         |
|            | TM int.   | Top 5 candidates       | 3.8    | 0.6   | 4.4   | 12.0%                         |
|            |           | Top 10 candidates      | 3.8    | 0.3   | 4.1   | 18.0%                         |
|            |           | Top 15 candidates      | 3.6    | 0.5   | 4.1   | 18.0%                         |
| Mech. eng. | No terms  |                        | 4.6    | 3.0   | 7.6   |                               |
|            | Term int. | Fuzzy search           | 3.6    | 2.0   | 5.6   | 21.1%                         |
|            |           | Stemmer + exact search | 3.5    | 2.2   | 5.7   | 19.7%                         |
|            |           | Stemmer + fuzzy search | 3.0    | 2.4   | 5.4   | 23.9%                         |
|            | TM int.   | Top 5 candidates       | 3.9    | 2.6   | 6.5   | 8.5%                          |
|            |           | Top 10 candidates      | 3.6    | 2.8   | 6.4   | 9.9%                          |
|            |           | Top 15 candidates      | 3.7    | 2.7   | 6.4   | 9.9%                          |

Table 3: LLM-as-a-judge evaluation results for Task 2 (English→Polish translation direction). Note that the test sets for Task 1 and Task 2 differ; therefore, the *No terms* results are slightly different in both experiments.

| Configuration             | Term success | Term success (lemmatized) | ChrF++ (paragraph) | COMET        |
|---------------------------|--------------|---------------------------|--------------------|--------------|
| <b>Task 1</b>             |              |                           |                    |              |
| Fuzzy search              | <b>77.8%</b> | <b>80.8%</b>              | <b>71.8</b>        | <b>0.881</b> |
| Stemming and fuzzy search | 75.0%        | 78.0%                     | 71.5               | 0.880        |
| <b>Task 2</b>             |              |                           |                    |              |
| Fuzzy search              | 77.4%        | <b>79.8%</b>              | 71.5               | 88.5         |
| Top 15 candidates         | <b>80.4%</b> | 78.7%                     | <b>77.2</b>        | 89.7         |
| Top 5 candidates          | 73.2%        | 73.7%                     | 75.4               | <b>89.9</b>  |

Table 4: Official results of our submissions. The best result for each task and metric is shown in bold.

shot translation using retrieved entries, which may point to a capability gap in the current instruction-tuned model rather than a limitation of few-shot translation-memory integration per se.

The practical value of terminology-aware machine translation has long been limited by the quality of the term collections themselves. In [Bergmanis and Pinnis \(2021a\)](#), we documented the noise typical of term collections repurposed for machine translation: entries where the source and target are not translations of each other, several possible translations with no indication of which one to use, unrelated terms and spelling variants collapsed into a single entry, terms listed in inflected rather than dictionary forms, and entries that explain a concept

in a full sentence instead of naming it. We argued that reformatting human-oriented term collections into machine-readable files does not make their content machine-usable, and that collections intended for machine translation must therefore meet a higher standard than those intended for humans – unless substantially more intelligent translation systems emerge, a condition that, in 2021, sounded akin to waiting for artificial general intelligence.

Ironically, no general intelligence was required: the caveat now simply describes the present. The systems described in this work interpret ambiguous entries in context, select among competing translation equivalents, and inflect dictionary forms as required by the target morphosyntax; definition-style

entries, once noise to be filtered out, could now even serve as useful context. While terminology-aware neural machine translation systems demanded cleaner term collections, terminology-aware translation with language models can now meet collections where they are. This reversal suggests that the burden of quality has shifted: rather than curating collections to suit rigid systems, the remaining challenge is selecting, from imperfect collections, the entries worth showing to a model that can handle the rest.

## Limitations

The primary limitation of this work stems from the Shared Task competition-like setting, which understandably leads to working without reference translations. The shared task did not provide a development set either. Participants were therefore left without any means of validating design decisions against ground truth: every choice we report – which term recognition method to use, how many translation-memory candidates to retrieve, whether to extract a glossary or prompt with retrieved entries – had to be made on the basis of an LLM-as-a-judge protocol applied on the test set itself. This is a shortcoming of the task design rather than of any individual submission, and it limits what the shared task as a whole can conclude about terminology integration: without development sets and references, participants cannot reliably compare methods, and system rankings rest solely on the organisers’ evaluation. Our own conclusions consequently depend on the judge model’s sensitivity to terminological adequacy, a property we could not verify. The Task 2 evaluation is weaker still: lacking a reference term collection, the judge received only the source text, the hypothesis, and the domain name, so terminological errors had to be recognised from its domain knowledge alone. We consider those results correspondingly noisier.

## Acknowledgments

The research leading to these results has received funding from the project “Competence Centre of Information and Communication Technologies for Research” of the Recovery and Resilience Facility, contract No. 5.1.1.2.i.0/2/24/A/CFLA/007 signed between IT Competence Centre and Central Finance and Contracts Agency, Research No. 2.6 “European Large Language Models for Ensuring Digital Content Accessibility”.

We acknowledge that this work also benefited from the EuroHPC JU computing award, granting access to LUMI HPC in Finland; project number EHPC-AI-2024A06-029.

## References

- Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, and 2 others. 2023. [GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch](#).
- Toms Bergmanis and Mārcis Pinnis. 2021a. Dynamic terminology integration for covid-19 and other emerging domains. In *Proceedings of the Sixth Conference on Machine Translation*, pages 821–827.
- Toms Bergmanis and Mārcis Pinnis. 2021b. 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.
- Toms Bergmanis, Ingus Jānis Pretkalniņš, Martins Kronis, Davis Nicmanis, Jelīzaveta Jelinska, Roberts Rozis, Rinalds Viksna, and Marcis Pinnis. 2026. [TildeOpen LLM: Leveraging curriculum learning to achieve equitable language representation](#). In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 10634–10652, Palma de Mallorca, Spain. ELRA Language Resource Association.
- 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.
- Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. [WIT3: Web inventory of transcribed and translated talks](#). In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.
- Adrian Charkiewicz, Pinzhen Chen, Thierry Etchegoyhen, Harritxu Gete Ugarte, Kamil Guttmann, 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.
- Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard,

- et al. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation.
- Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, et al. 2025. WMT24++: Expanding the Language Coverage of WMT24 to 55 Languages & Dialects. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12257–12284.
- 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.
- Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128-News Test References for MT Evaluation of 128 Languages. In *Proceedings of the First Workshop on Scaling up Multilingual Evaluation*, pages 21–24.
- Mercedes García-Martínez, Laurent Bié, Aleix Cerdà, Amando Estela, Manuel Herranz, Rihards Krišlauks, Maite Melero, Tony O’Dowd, Sinéad O’Gorman, Mārcis Pinnis, et al. 2021. Neural Translation for European Union (NTEU). In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, pages 316–334.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. [Gemma 2: Improving open language models at a practical size](#). Preprint, arXiv:2408.00118.
- Anna Lackner, Alena Vega-Wilson, and Christian Lang. 2025. An evaluation of terminology-augmented generation (tag) and various terminology formats for the translation use case. *Journal of Digital Terminology and Lexicography*, 1(JDTL-Volume 1, issue 2):31–47.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023a. [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. European Association for Machine Translation.
- 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.
- NLLB Team. 2024. [Scaling neural machine translation to 200 languages](#). *Nature*, 630:841–846.
- Martin F. Porter. 2001. [Snowball: A language for stemming algorithms](#). Accessed: 2026-08-05.
- 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.
- Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. [CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Miguel Rios. 2025. [Instruction-tuned large language models for machine translation in the medical domain](#). In *Proceedings of Machine Translation Summit XX: Volume 1*, pages 162–172, Geneva, Switzerland. European Association for Machine Translation.
- Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W. Oard, and Matt Post. 2021. [The multilingual TEDx corpus for speech recognition and translation](#). In *Proceedings of Interspeech 2021*, pages 3655–3659.
- Jörg Tiedemann. 2012. [Parallel data, tools and interfaces in OPUS](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).
- Paula Zorrilla-Agut and Thierry Fontenelle. 2019. [IATE 2: Modernising the EU’s IATE terminological database to respond to the challenges of today’s translation world and beyond](#). *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 25(2):146–174.

## A Term Recognition Efficiency Analysis

The three different term recognition methods analysed in our work differ not only in term recognition recall, but also in efficiency. Figure 1 shows that the trie-based method that uses a stemmer and

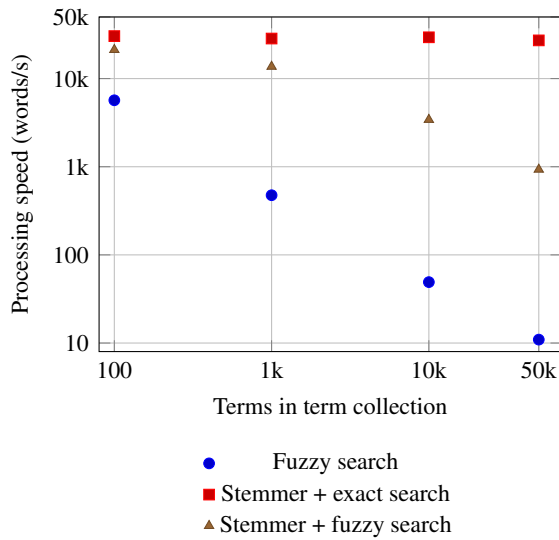


Figure 1: Term-recognition throughput for different term-collection sizes. Both axes use logarithmic scales.

exact match search is the most efficient and is applicable for large term collections. The Trie-based method that uses fuzzy search is applicable for term collections of up to 1k entries; beyond that, its performance degrades. The fuzzy search method is not applicable in practice.

## B Prompt examples

Figures 2, 3, 4, 5, 6, and 7 provide prompt examples used for translation generation, terminology extraction, and automatic translation evaluation.

```

messages = [
  {"role": "system", "content": "You are a translator."},
  {"role": "user", "content": "A measurable reduction in olfactory sensitivity, clinically
  ↳ known as hyposmia, often precedes total olfactory failure, which is termed anosmia.
  ↳ Upper respiratory tract infections, particularly those involving viral binding to the
  ↳ sustentacular cells of the nasal mucosa, frequently cause transient hyposmia, whereas
  ↳ severe shearing trauma to the cribriform plate during a traumatic brain injury can result
  ↳ in absolute and permanent anosmia.\n\nTranslate the given text from English to Polish."}
]

```

Figure 2: Example of a TildeOpen translation prompt without terminology

```

messages = [
  {"role": "system", "content": "You are a translator."},
  {"role": "user", "content": "A measurable reduction in olfactory sensitivity, clinically
  ↳ known as hyposmia, often precedes total olfactory failure, which is termed anosmia.
  ↳ Upper respiratory tract infections, particularly those involving viral binding to the
  ↳ sustentacular cells of the nasal mucosa, frequently cause transient hyposmia, whereas
  ↳ severe shearing trauma to the cribriform plate during a traumatic brain injury can result
  ↳ in absolute and permanent anosmia.\n\nTranslate the given text from English to Polish,
  ↳ using this glossary {\\"reduction\\": [\\"zmniejszenie\\"], \\"olfactory sensitivity\\":
  ↳ [\\"wrażliwość węchowa\\"], \\"hyposmia\\": [\\"hiposmia\\"], \\"olfactory failure\\": [\\"utrata
  ↳ węchu\\"], \\"failure\\": [\\"niewydolność\\"], \\"anosmia\\": [\\"anosmia\\"]."}
]

```

Figure 3: Example of a TildeOpen translation prompt with an in-context term collection

```

messages = [
  {"role": "system", "content": "You are a translator."},
  {"role": "user", "content": "A measurable reduction in olfactory sensitivity, clinically
  ↳ known as hyposmia, often precedes total olfactory failure, which is termed
  ↳ anosmia.\n\nTranslate the given text from English to Polish."},
  {"role": "assistant", "content": "Mierzalne zmniejszenie wrażliwości węchowej, klinicznie
  ↳ znane jako hiposmia, często poprzedza całkowitą utratę węchu, zwaną anosmią."},
  {"role": "user", "content": "These gustatory deficits are clinically classified into three
  ↳ distinct categories: hypogeusia (a diminished sense of taste), ageusia (a complete and
  ↳ total loss of taste), or dysgeusia (a severe distortion of taste, where benign foods
  ↳ often taste intensely metallic, bitter, or foul).\n\nTranslate the given text from
  ↳ English to Polish."},
  {"role": "assistant", "content": "Te deficyty smakowe klasyfikuje się klinicznie do trzech
  ↳ odrębnych kategorii: hipogeuzja (osłabienie odczuwania smaku), ageuzja (całkowita i
  ↳ całkowita utrata smaku) i dysgeuzja (poważne zniekształcenie smaku, w którym łagodne
  ↳ pokarmy często mają intensywnie metaliczny, gorzki lub nieprzyjemny smak)."},
  {"role": "user", "content": "A measurable reduction in olfactory sensitivity, clinically
  ↳ known as hyposmia, often precedes total olfactory failure, which is termed anosmia.
  ↳ Upper respiratory tract infections, particularly those involving viral binding to the
  ↳ sustentacular cells of the nasal mucosa, frequently cause transient hyposmia, whereas
  ↳ severe shearing trauma to the cribriform plate during a traumatic brain injury can
  ↳ result in absolute and permanent anosmia. Because the human perception of complex flavor
  ↳ relies heavily on retronasal olfaction, a patient suffering from anosmia or even
  ↳ profound hyposmia will frequently present to the clinic complaining primarily of taste
  ↳ abnormalities rather than smell loss. These gustatory deficits are clinically classified
  ↳ into three distinct categories: hypogeusia (a diminished sense of taste), ageusia (a
  ↳ complete and total loss of taste), or dysgeusia (a severe distortion of taste, where
  ↳ benign foods often taste intensely metallic, bitter, or foul)."}
]

```

Figure 4: Example of a TildeOpen translation prompt with translation memory examples as few-shot examples

```

messages = [
  {"role": "user", "content": """Extract all English-Polish (en-pl) term pairs from the texts
  ↳ below.

  A term pair must contain reciprocal translations that are specific to the medical domain.
  ↳ Extract nouns, noun phrases, verbs, and verb phrases.

  Keep in mind that dictionary forms are lowercase, nominative, singular, etc. except for proper
  ↳ nouns and acronyms.
  - Include only terms that are supported by both texts.
  - Do not invent translations.
  - Return JSON only. Do not include explanations or commentary.

  Return exactly this JSON structure:

  {{
    "src_lang": "en",
    "trg_lang": "pl",
    "term_pairs": [
      {{
        "src_term": {{
          "dict_form": "source dictionary form",
          "surface_form_in_text": "exact source surface form"
        }},
        "trg_term": {{
          "dict_form": "target dictionary form",
          "surface_form_in_text": "exact target surface form"
        }}
      }}
    ]
  }}

  When no valid domain-specific term pairs exist, return:

  {{
    "src_lang": "en",
    "trg_lang": "pl",
    "term_pairs": []
  }}

  Texts:
  English: ...
  Polish: ..."""}
]

```

Figure 5: Example of a term extraction prompt for the Gemma 4 31B LLM

```

messages=[
  {"role": "system", "content": "You are a translation evaluation agent."},
  {"role": "user", "content": "List all minor, major, and insignificant translation errors
  ↳ (mistranslation, omission, wrong lexical choice, syntax, spelling, terminology, style,
  ↳ punctuation, etc.) for the following English-Polish (en-pl) hypothesis (translation) of
  ↳ the source text. The domain is: medical. Use the following term collection for reference:
  ↳ {"neoplasm": ["guz"], "carcinogenesis": ["karcynogeneza"]}.
  ↳ Source text
  ↳ (English): The clinical genesis of any neoplasm is intrinsically tied to the multifaceted,
  ↳ sequential biological cascade occurring through carcinogenesis.
  ↳ Hypothesis (Polish):
  ↳ Kliniczne podłoże każdego nowotworu jest nierozzerwalnie związane z wieloaspektową,
  ↳ sekwencyjną kaskadą biologiczną zachodzącą w trakcie karcynogenezy."}
]

```

Figure 6: Example of an LLM-as-a-judge prompt for translation error annotation using the GPT-5.6-luna LLM (for the scenario with a reference term collection; Task 1)

```

messages=[
  {"role": "system", "content": "You are a translation evaluation agent."},
  {"role": "user", "content": "List all minor, major, and insignificant translation errors
  ↳ ( mistranslation, omission, wrong lexical choice, syntax, spelling, terminology, style,
  ↳ punctuation, etc.) for the following English-Polish (en-pl) hypothesis (translation) of
  ↳ the source text. The domain is: medical.\nSource text (English): The clinical genesis of
  ↳ any neoplasm is intrinsically tied to the multifaceted, sequential biological cascade
  ↳ occurring through carcinogenesis.\nHypothesis (Polish): Kliniczne podłoże każdego
  ↳ nowotworu jest nierozzerwalnie związane z wieloaspektową, sekwencyjną kaskadą biologiczną
  ↳ zachodzącą w trakcie karcynogenezy."},
]

```

Figure 7: Example of an LLM-as-a-judge prompt for translation error annotation using the GPT-5.6-luna LLM (for the scenario without a reference term collection; Task 2)