

Krosny at WMT26 General Translation Task: Dictionary Augmentation and Iterative Refinement for English–Belarusian Translation

Oksana Volchek, Vladislav Poritski
Independent researchers, Vilnius, Lithuania
{volchekoa,v.poritski}@gmail.com

Abstract

This paper describes our submission to the WMT26 General Machine Translation shared task for the English $\rightarrow$ Belarusian direction. In lower-resourced morphologically rich languages like Belarusian, open-weight large language models (LLMs) frequently struggle with orthographic fidelity, hallucinated vocabulary, and negative transfer from related high-resource languages. To address these issues, we implemented Krosny,¹ an iterative, resource-augmented pipeline built on top of TranslateGemma 12B. The system translates each segment in three stages: (1) dictionary-augmented generation, (2) refinement using a reference grammatical database, and (3) rule-based post-processing followed by an LLM-as-a-judge candidate selection. We find a small but consistent improvement over the baseline TranslateGemma 12B. In comparison with other constrained-track systems of WMT26, the performance of Krosny is mid-tier. We release the source code of our implementation.²

1 Introduction

The advent of large language models has shifted the machine translation paradigm towards prompting pre-trained models (Zhang et al., 2023; He, 2024). However, for a low-resource language like Belarusian, which is susceptible to negative transfer from related languages (Russian and Ukrainian) and LLM hallucinations, relying on the model’s parametric memory alone may yield suboptimal translation quality in resource-constrained settings.

Our submission to the WMT26 shared task is limited to English–Belarusian language pair. The WMT26 blind set for this language pair consists of 198 texts across four domains: 133 speech transcripts, 44 social media posts, 14 news articles, and 7 software localization JSONs. To translate

these diverse texts, we chose TranslateGemma 12B (Finkelstein et al., 2026), a strong multilingual local model fine-tuned for machine translation, as our base model and aimed to determine whether it is possible to meaningfully improve its outputs by wrapping it into a language-specific scaffold without any additional training or fine-tuning. We designed a pipeline that augments the model’s context with relevant entries from English–Belarusian dictionaries and a grammatical database, applies rule-based post-processing and invokes LLM as a judge for candidate selection.

2 Linguistic Resources and Pre-processing

Our pipeline relies on several existing Belarusian linguistic resources. Their intended purpose is to complement the model’s parametric knowledge. LLMs are known to be bad at memorizing “long-tail” knowledge (Kandpal et al., 2023); in a low-resource language, TranslateGemma could be therefore expected to know rare vocabulary and morphology poorly.

Dictionary Context and Orthography Normalization. Dictionary hints have been shown to improve translation quality between English and low-resource languages (Ghazvininejad et al., 2023). We utilize two English–Belarusian dictionaries available in digital form:³ one edited under the supervision of T. Susha (2013), another authored by V. Pashkevich and edited by S. Shupa (2006).

Belarusian has two competing written standards: the classical orthography *taraškievica*, named after Branislaŭ Taraškievič’s 1918 grammar, was the norm until the Soviet spelling reform of 1933, while the reformed, official orthography (informally *narkamaŭka*), later revised in 1957/1959 and by the 2008 orthography law, is what is taught

¹In Belarusian: *кросны* ‘hand loom’.

²<https://github.com/volchek/Krosny-WMT26>

³<https://github.com/verbumbyslouniki>

in schools, used in official and most media texts, and can be reasonably expected in the WMT26 test data translations. Pashkevich’s dictionary uses *taraškievica*, so injecting entries from it directly into the prompt context could cause the LLM to output non-compliant orthography. To prevent this, we applied an offline “detarashkevization” script to Pashkevich’s dictionary, converting classical spellings into the official modern orthography with a sequence of regular expressions.

Filtering by Frequency. To prevent the prompt context window from being overwhelmed by high-frequency vocabulary (which the LLM is already likely to translate accurately), we implemented a frequency-based filter. We lemmatized the entire input set and counted occurrences of all English nouns, proper nouns, verbs, adjectives, and adverbs. Only words belonging to these parts of speech and appearing ≤ 50 times in the inputs were eligible for dictionary lookup. While the filter is derived in our implementation directly from the test set, it could as well be based on a generic corpus of texts from the same distribution. Its purpose is to ensure that the lemmas whose dictionary entries are injected into the context are sufficiently specific or rare.

GrammarDB and Error-Weighted Levenshtein Distance. To validate the LLM’s outputs, we used GrammarDB,⁴ the reference morphological database of Belarusian. Rather than just rejecting non-attested words, we designed an error-correction module. During the development phase, we manually accumulated a list of ≈ 150 common morphological and spelling errors produced by TranslateGemma and its base model Gemma 3 12B (e.g.: *тэхналогічнаму* instead of *тэхналагічнаму*). By calculating the statistics of edit operations between these generated errors and their manual corrections, we derived a custom set of weights ranging between 0 and 1: frequent edits, such as *o/a* replacement in the above example, get proportionally smaller weights. For a generated token that is missing from GrammarDB, Levenshtein distances with these custom weights are calculated against all tokens of similar length from GrammarDB. Lower penalties assigned to frequent edits increase the chances of suggesting valid GrammarDB counterparts for Gemma-specific hallucinated Belarusian wordforms.

⁴<https://github.com/Belarus/GrammarDB>

3 Translation Pipeline

Our inference pipeline executes in three stages. To optimize context length, input texts are split into segments with mixed granularity depending on the domain (e.g., all transcripts are sufficiently short and don’t require splitting); typically, segment length is limited to 500 characters. All LLM inferences were performed with an instance of TranslateGemma 12B in 8-bit GGUF quantization⁵ via llama.cpp⁶ in the default non-deterministic configuration: temp=1.0, top_p=0.95, top_k=64.

3.1 Dictionary-Augmented Generation

For each input segment, we use the respective domain-specific system prompt, provided by the organizers, to generate $N = 3$ “naive” baseline translations. Next, we prepend the filtered English–Belarusian dictionary entries for the vocabulary present in this segment to the prompt and generate N “guided” translations, yielding $2N$ initial candidate translations per segment. At this stage, candidate generation is retried if the output is not in Belarusian (as detected by Lingua⁷) or its length is below 0.5x or above 2x of the input length, i.e. the model either truncates the input or falls into a hallucination loop.

3.2 GrammarDB-Guided Merging

We pair the naive and guided translations. From each pair, we extract all tokens eligible for GrammarDB lookup: they should contain Cyrillic characters and shouldn’t be too short or belong to one of pre-defined classes, such as hashtags, spellings with digits, proper names (as detected by Stanza⁸). Eligible tokens that do not exist in GrammarDB are flagged. For each flagged token, we retrieve suggestions from GrammarDB, ranked by the error-weighted Levenshtein distance – up to 100 suggestions but typically much less, as the similarity threshold increases with token length. The LLM is then prompted as an “expert editor”: given the English source, both candidate Belarusian translations, and a glossary that maps the unverified generated words to GrammarDB suggestions, the model synthesizes a single refined translation. This yields N refined candidates.

⁵<https://huggingface.co/mradermacher/translategemma-12b-it-GGUF>

⁶<https://github.com/ai-dock/llama.cpp-cuda>

⁷<https://github.com/pemistahl/lingua>

⁸<https://github.com/stanfordnlp/stanza>

3.3 Post-Processing and Final Selection

To account for the negative transfer from Russian and, to a lesser extent, Ukrainian orthography in the LLM-generated Belarusian texts, we apply a suite of regex corrections to the refined outputs (e.g., converting *u* to *i/bi* depending on the context, *u* to *uu*, enforcing Belarusian verb endings like *u*_b). Consistent formatting (valid HTML or JSON, same number of paragraphs in the output as in the respective input) is also enforced at this stage.

Finally, we use the LLM as a judge to evaluate the post-processed candidates. The model is asked to output only the index (1...*N*) of the most adequate and correct translation. To mitigate positional bias, this selection prompt is run $K = 20$ times per segment with randomized order of candidates. The final choice is determined by majority vote.

4 Computational Footprint

Our iterative approach trades compute for quality. Translating a single segment requires at least $3N$ “heavy” generation calls (with retries) and K “lightweight” calls for the candidate selection phase, where typically only a single token is generated. The English–Belarusian subset of the WMT26 blind set (198 texts split into 729 segments) was processed on Kaggle using 2xT4 GPUs within the 30-hour free weekly quota, with the GrammarDB suggestions computed locally on a CPU.

5 Preliminary Evaluation

As manual gold translations weren’t available in the English–Belarusian pair at the time of submission, we used the pseudo-reference provided by the organizers (PseudoRef.jsonl). Using the default settings of SacreBLEU (Post, 2018), we calculated BLEU and chrF2 scores against this pseudo-reference for the “naive” and “guided” outputs of TranslateGemma, for our final submission, and for the outputs of five other constrained-track systems that support English–Belarusian translation and two strong proprietary LLMs: Gemini 3.1 Pro and GPT-5.5. The results are shown in Table 1.

The injection of dictionary context provided a small but consistent improvement over the naive baseline across all runs. Our final submission, which adds GrammarDB-informed candidate merging, rule-based post-processing and LLM candidate selection, achieved further incremental gain. However, this improvement falls short of matching the quality delivered by the best constrained systems,

System	Run	BLEU	chrF2
Tr.G., naive	1	32.18	56.06
	2	32.69	56.21
	3	32.80	56.14
Tr.G., guided	1	33.49	56.60
	2	33.73	56.82
	3	33.83	56.63
Krosny		36.36	57.98
Lumen		52.61	68.57
Qingqiu-MT-9B		47.04	64.74
ReMova		33.39	50.43
SalamandraTA		36.87	56.71
Yandex-8B		40.99	59.29
Gemini 3.1 Pro		51.63	67.50
GPT 5.5		54.21	69.35

Table 1: Automated evaluation scores.

such as Lumen or Qingqiu-MT-9B, which perform at or around the level of proprietary LLMs.

Additionally, for all of the systems listed above, plus the pseudo-reference, we re-used our implementation of GrammarDB lookup to calculate the ratios of generated tokens missing from it. While there may be legitimate reasons for a valid word form not to be listed in the reference database (e.g.: it is a recent neologism, a calque, or its spelling follows the older standard), high out-of-vocabulary token ratio is generally correlated with low fluency. The ratios are shown in Table 2, along with the total counts of word tokens produced by each system.

System	Run	Tokens	N/E	+	−
Tr.G., naive	1	38158	47.56	49.75	2.69
	2	37926	47.24	50.11	2.64
	3	37871	47.46	49.98	2.56
Tr.G., guided	1	38086	46.57	51.32	2.11
	2	38083	46.66	51.28	2.06
	3	37873	46.53	51.47	2.00
Krosny		37197	46.74	51.35	1.91
Lumen		35073	45.77	53.01	1.22
Qingqiu-MT-9B		35182	45.85	51.50	2.65
ReMova		31120	46.37	52.47	1.17
SalamandraTA		36001	46.24	52.96	0.81
Yandex-8B		35472	47.09	50.22	2.70
Gemini 3.1 Pro		35083	45.85	52.91	1.24
GPT 5.5		34279	46.16	52.75	1.10
Pseudo-reference		35628	45.54	53.36	1.10

Table 2: Word tokens in the outputs and their percentage breakdown with respect to GrammarDB lookup: not eligible (N/E), hit (+), or miss (−).

Dictionary augmentation alone reduced GrammarDB misses by approximately a fifth, from 2.63% to 2.06%. Stage 2 editing and stage 3 post-processing further reduce this ratio only slightly, indicating that TranslateGemma doesn’t make use of the GrammarDB suggestions to the full possible extent. In the outputs of proprietary models and the pseudo-reference, out-of-vocabulary token counts

are even lower (1.1–1.2%). Low or high ratio of GrammarDB misses isn’t by itself a reliable predictor of translation quality, as can be seen from the scores of e.g. SalamandraTA and Qingqiu-MT-9B. As a tendency, we observe that SalamandraTA is more fluent in Belarusian but frequently distorts the meaning of the input, while Qingqiu-MT-9B is more faithful, despite many hallucinated or coined words and grammatical errors. The example below illustrates this point, with lexical and grammatical errors highlighted.

	... with the broken shoulder he sustained
Input	proving untreatable and he was euthanised by the veterinary team on Saturday.
	У выніку ў яго адламалася лапатка,
SalamandraTA	што не паддалася лячэнню. У суботу ветэрынары адвялі яму эўтаназію.
	‘As a result, his scapula broke off and could not be treated. On Saturday, the veterinarians euthanized him.’
	Атрыманая ім <i>пераломленне</i> пляча
Qingqiu-MT-9B	аказалася невылечнай, і ў суботу ветэрынарная група <i>эўтаназіявала</i> яго.
	‘The shoulder fracture he sustained proved untreatable, and on Saturday the veterinary team euthanized him.’

6 Discussion and Limitations

Out-of-Domain Use of a Fine-Tune. TranslateGemma is a specialized, single-task fine-tune with an opinionated prompt template. In our pipeline, we discard this template and force the model into diverse roles: a context-augmented translator, an expert editor, and a zero-shot evaluator. A priori, it wasn’t clear how much of the base model’s general instruction-following capability would survive the translation fine-tuning. Our findings suggest that at least partially it remains robust: e.g., the model is able to take dictionary context into account.

Ablation of Post-Processing. In our current evaluation, the impacts of GrammarDB-guided editing (stage 2) and rule-based post-processing (stage 3) aren’t analyzed separately. We currently lack the numeric evidence to estimate exactly how much of the quality gain is attributable to the GrammarDB context injection vs. the regex corrections.

Regression to the Mean. Multi-stage refinement using LLMs carries an inherent risk of “regression to the mean”: in each subsequent step (merging, evaluating), the model may inadvertently overwrite or “forget” accurate translation choices made in earlier steps, homogenizing the output. Quantifying

the extent of this homogenization in our pipeline is left to future analysis.

Pipeline Portability and Future Architectures.

Stage 2 of our pipeline is calibrated to the specific failure modes of the Gemma model family. While in principle it is possible to swap the base model (e.g., with a more aggressively quantized version of TranslateGemma), choosing a totally different model may require recalibrating the Levenshtein weights. Furthermore, as newer LLMs demonstrate stronger agentic capabilities, a more dynamic pipeline design might fit them better: instead of forcing offline dictionary and GrammarDB context into the prompts, both types of resources could be exposed via MCP, allowing the model to query them at will.

7 Conclusion

We presented Krosny, a scaffold for LLM-based English–Belarusian translation, designed to mitigate LLM hallucinations and improve output fluency. By augmenting TranslateGemma-12B with dictionary context and spell-check suggestions from a reference morphological database, we sought to limit the scope of the model’s deviations from the modern Belarusian literary standard. Preliminary evaluation shows that our iterative pipeline reduces the ratio of out-of-vocabulary tokens in the output, yields a +3.8 BLEU / +1.8 chrF2 improvement over a zero-shot baseline, but doesn’t help achieve translation quality comparable with the best constrained-track systems. Echoing the bitter lesson (Sutton, 2019), this may be an inherent limitation of our approach, i.e. injecting language-specific knowledge into the prompt context without modifying the weights. Still, we observe that leveraging language-specific resources is a viable path to some quick wins when computational resources are not sufficient for large-scale training or fine-tuning runs.

References

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.

- Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. [Dictionary-based phrase-level prompting of large language models for machine translation](#). *Preprint*, arXiv:2302.07856.
- Sui He. 2024. [Prompting ChatGPT for translation: A comparative analysis of translation brief and persona prompts](#). In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 316–326, Sheffield, UK. European Association for Machine Translation (EAMT).
- Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. [Large language models struggle to learn long-tail knowledge](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Rich Sutton. 2019. The bitter lesson. <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>.
- Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. [Prompting large language model for machine translation: A case study](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.