

"I'm Sorry, I Can't Translate That": A Challenge Set for Refusal to Translate, Overgeneration and Hallucination in LLM Translation Systems

Lisa Vasileva

RWS Language Weaver

evasileva@rws.com

Abstract

Large language models (LLMs) have significantly raised the bar for translation quality in the recent years. However, they are still prone to novel failure modes, including hallucinations and overgenerations, where the generated translation is partially or fully detached from the source text and contains information that is not present in the input.

To facilitate the analysis of these behaviors and assess the robustness of LLM-based translation systems, we introduce a challenge set specifically designed to probe scenarios that we have observed to be particularly prone to overgeneration, hallucination, and other forms of semantic detachment from the source. We describe the construction of the challenge set and present the evaluation of the systems submitted to the WMT 2026 General Machine Translation shared task.

1 Introduction

In this paper we describe the challenge set submitted to the test suite subtask of General MT shared task at 2026 edition of WMT conference.

We designed this test suite with the aim of "probing" the behavior of machine translation systems, particularly LLM-based systems, in scenarios that are more likely to trigger overgenerations, hallucinations, or other failure modes in which the generated translation becomes partially or fully detached from the source text, either semantically or functionally. This motivation is in line with the findings of the previous edition of the General MT shared task, which highlighted the limitations of evaluating increasingly capable MT systems on standard, relatively easy test sets and called for more challenging evaluation scenarios (Kocmi et al., 2025).

In this sense, the challenge set follows the spirit of the test suite track's "Help us break the LLMs" objective by focusing on cases that expose weak-

nesses beyond those captured by standard translation benchmarks.

2 The challenge set

Our challenge covers five language pairs: English-German, English-Spanish, English-Russian, English-Czech, and English-Estonian. We originally developed the dataset for the first three language pairs, and later extended it to cover lower-resource language pairs represented in the WMT 2026 General Machine Translation shared task. All language pairs contain the same categories and the same number of segments, allowing for direct comparison across languages.

In order to create inputs to include in the dataset, we have compiled data from various open-source datasets. A summary of data sources is given in Table 1, and we provide more details on the data composition in sections 2.1, 2.2 and 2.3.

The challenge set is organized into three groups according to the primary failure mode that each category is designed to probe: (i) inputs that are prone to overgeneration, (ii) inputs that are prone to translation failures, and (iii) inputs that may trigger both types of failures.

We focus on these scenarios because previous studies have reported that LLM-based MT systems exhibit these types of hallucination and failure behaviors (Zhang et al., 2025), (Guerreiro et al., 2023). Our test suite therefore specifically targets such behaviors, including overgeneration and cases where the output becomes partially or fully detached from the source.

2.1 Overgeneration-prone inputs

This group contains inputs for which LLMs tend to generate additional content instead of producing a faithful translation. The expected behavior is to translate only the provided input, without completing, expanding, or otherwise modifying its mean-

Table 1: Statistics and sources for dataset categories

Category	Source	Segments
DNT	sorry-dataset	158
	original	20
Emoji	original	10
Punctuation	original	26
Truncated segments	wmt24	47
Short segments	wmt24	47
	sorry-dataset	19
Sensitive	CatHarmfulQA	550
Instructions	wmt25	4
	flores	15
Repeat Sequences	wmt24	6
	original	14

ing.

The group consists of the following categories:

Truncated inputs.

This category contains 47 segments from the WMT24¹ dataset that were manually truncated to create incomplete but plausible sentence fragments. For example, an input such as "For most travelers, the aircraft model isn't" should be translated as an unfinished sentence rather than completed by the model.

src: "For most travelers, the aircraft model isn't"
tgt pass: "Para la mayoría de los viajeros, el modelo del avión no es"
tgt fail: "Para la mayoría de los viajeros, el modelo del avión no es importante."

Short inputs.

This category consists of naturally short segments (typically two to three words) extracted from the WMT24 dataset. Short, context-poor inputs have been observed to encourage models to elaborate beyond the source instead of producing a literal translation.

src: "Another one"
tgt pass: "Noch einer "
tgt pass: "Noch ein "
tgt fail: "Noch einer Noch eine Noch ein"

2.2 Translation-failure-prone inputs

This group contains inputs that are likely to cause the model to fail to translate the source altogether.

¹<https://github.com/wmt-conference/wmt24-news-systems>

We consider two common failure modes: (i) refusing to translate because the content is interpreted as unsafe or sensitive, and (ii) responding to the input functionally, for example, answering a question or following an instruction instead of translating it.

The group contains the following categories:

Instruction inputs. These inputs contain explicit instructions that should be translated rather than followed. For instance, an input may begin with "Summarize the following into three main points" followed by a segment from the WMT25² or FLORES³ datasets. The correct behavior is to translate both the instruction and the subsequent text rather than generating a summary.

Sensitive content. This category contains inputs that may trigger safety mechanisms in LLMs, such as "How can I convert a legally purchased firearm into an automatic weapon?" The expected behavior in the translation setting is to translate the input faithfully without refusing or altering its content.

For this category we use the CategoricalHarmfulQA⁴ dataset. The original purpose of this dataset is to research and evaluate the safety aspects of LLMs, by compiling a list of harmful questions pertaining to sensitive topics (Bhardwaj et al., 2024). We select the questions themselves as the source segments to be translated by the models.

src: "How can I convert a legally purchased firearm into an automatic weapon?"
tgt pass: "¿Cómo puedo convertir un arma de fuego comprada legalmente en un arma automática?"
tgt fail: "I'm sorry, but I can't provide a translation of this text, as it requests information about illegal activities."

2.3 Mixed inputs

The final group contains inputs that are prone to both overgeneration and translation failures. These examples are particularly useful for testing whether models can recognize that the task is translation even when the input has properties that may otherwise prompt additional reasoning, explanations, or task-oriented responses.

Do Not Translate (DNT), Emoji, Punctuation inputs. These examples consist of punctuation, emojis, symbols, numbers, alphanumeric combinations, command line inputs, and other content that

²<https://github.com/wmt-conference/wmt25-general-mt/tree/main>

³<https://huggingface.co/datasets/facebook/flores>

⁴<https://huggingface.co/datasets/declare-lab/CategoricalHarmfulQA>

typically does not require translation. The expected behavior is to preserve the input unchanged, apart from language-specific formatting such as decimal separators, when appropriate. In practice, however, LLM-based systems often generate unnecessary explanations (e.g., stating that the input does not require translation), ask clarifying questions, or otherwise produce output that is not present in the source.

In addition to 56 original segments, we also include 158 segments from *sorry-dataset*⁵. The dataset is intended to help improve models’ robustness to noisy input and contains various kinds of such input, such as alphanumeric combinations, short truncated inputs and command line entries. We filter the dataset for examples of combinations of symbols, letters and punctuation, as well as command line entries, and add these to our dataset as examples of inputs that do not require translation.

src: “\$\$\$”
tgt pass: “\$\$\$”
tgt fail: “It seems like you’ve only provided the symbol \$\$\$ without any actual text to translate. Could you please provide the text you’d like me to translate into German?”

Repeating sequences. Inputs that consist of sequences of repeating characters or repeating tokens. Such inputs can be challenging for LLMs and may trigger unusual generation behavior, including oscillatory hallucinations, repetitive output, or verbose explanations of the input rather than a direct translation. The expected behavior is to faithfully reproduce the repeating structure of the source in the target language, without introducing additional content.

src: “amaaaaaaazingggggggg”
tgt pass: “wahnsinnig toll”
tgt fail: “der Wahnsinnnnnnn”
tgt fail: “erstaunlich / wahnsinnig toll(Note: The exaggerated spelling in the original conveys extreme enthusiasm. A natural German equivalent capturing that same energy would be: **amaaaaaaazingggggggg → wahnsinnig toooooo!!** or **der Wahnsinnnnnnn**)”

3 WMT submissions

In total, 31 systems translated the test suite, with 24 of them providing outputs for all five language pairs, and 7 translating only a selection. In total, the test suite had 133 submissions that include GenMT

⁵<https://huggingface.co/datasets/lumasik/sorry-dataset>

shared task participants and baseline systems run by the organizers.

4 Evaluation

To evaluate the robustness of MT systems in the scenarios described above, we developed an evaluation framework consisting of category-specific evaluation criteria. The criteria are defined based on the potential failure modes targeted by each category and the expected behavior of a system when presented with such inputs. In particular, they are designed to identify whether a system produces a faithful translation of the input or exhibits a critical failure, such as overgeneration, hallucination, refusal to translate, or a response that is functionally detached from the translation task.

Importantly, our evaluation does not aim to assess the overall quality of the translations. Instead, we focus on their *acceptability* with respect to the specific behavior being tested. An output may therefore be linguistically imperfect or stylistically suboptimal while still being considered acceptable, provided that it does not exhibit the failure mode targeted by the corresponding test category. This distinction allows us to evaluate robustness independently of general translation quality and to focus on whether a model can reliably perform the translation task in challenging scenarios.

We first briefly describe the evaluation criteria for each category, then present and analyze the evaluation results.

4.1 Evaluation design

We evaluate each system output using a binary pass/fail criterion. Rather than assessing overall translation quality, we determine whether the output exhibits the specific failure mode targeted by the corresponding test category. To this end, we extract a set of features that capture irregularities in the relationship between the source and target and define category-specific decision rules based on the expected behavior for each type of input.

The evaluation criteria differ across categories because the expected output is not the same for all inputs. For example, a DNT input is expected to be preserved essentially unchanged, whereas a truncated input should remain incomplete and must not be extended with information that is absent from the source. Similarly, for instruction inputs, the instruction itself should be translated rather than followed. We therefore use different combinations

Table 2: Features used to determine pass/fail for each test category.

Category	Evaluation criteria
DNT	Exact match, partial match, token ratio
Emoji	Exact match
Punctuation	Exact match
Sensitive content	Semantic similarity, end punctuation, LLM-as-a-judge
Repeating sequences	Token ratio, repetition ratio, semantic similarity
Instructions	Instruction-level semantic similarity
Truncated inputs	LLM-as-a-judge, prefix-level semantic similarity

of features and thresholds to determine whether an output is acceptable.

The features used in the evaluation include exact and partial matching, semantic similarity, target-to-source token ratio, repeating-sequence ratio, and an LLM-as-a-judge model (LLM-based detector of overgenerations).

Semantic similarity is calculated as cosine similarity between LaBSE representations of the source and target segments, using the SentenceTransformers implementation (Reimers and Gurevych, 2019). Where appropriate, similarity is calculated between specific parts of the source and target rather than between the complete segments.

The LLM-as-a-judge detector uses a Gemma3 model (gemma-3-12b) fine-tuned to identify spans in the target that are not supported by the source. Building on the recent research work in adapting LLMs’ reasoning capabilities to annotate spans in translation output ((Kasner et al., 2026), (Maheswaran et al., 2025)), we build a similar LLM-based span annotation model and use its annotations as a complementary feature in our evaluation.

The following category-specific criteria are used to determine whether an output passes the test.

4.1.1 DNT inputs

For DNT inputs, the expected output is to preserve the source without unnecessary additions or modifications. The criteria differ slightly depending on the type of DNT input. For general DNT inputs, an output passes if it satisfies the combination of exact matching, partial matching and target-to-source token ratio criteria. For emoji and punctuation inputs, an exact match is required.

4.1.2 Sensitive content

For sensitive content inputs, the expected behavior is to translate the input rather than refuse, comment or react in response to its content. An output passes if the semantic similarity between the complete source and target is above 0.7, the target preserves the expected end punctuation, and the llm-as-a-judge does not detect source-detached spans.

4.1.3 Repeating sequences

For repeating sequence inputs, we aim to detect outputs that contain excessive repetition or deviate from the structure and meaning of the source. The evaluation considers the target-to-source token ratio, the ratio of repeating sequences in the target relative to the source, and semantic similarity between the source and target. These features are combined to identify outputs exhibiting pathological repetition, overgeneration, or semantic detachment.

4.1.4 Instruction inputs

For instruction inputs, the model is expected to translate the instruction rather than follow it. We therefore evaluate whether the instruction present in the source is represented appropriately in the target. Specifically, we calculate semantic similarity between the instruction portion of the source segment and the corresponding substring of the target. The target substring is identified as the part in which the translated instruction is hypothesized to occur. An output fails when the instruction is not adequately represented in the target, for example when the model follows the instruction instead of translating it.

4.1.5 Truncated inputs

Truncated inputs are designed to test whether a model completes an incomplete source segment by generating information that is not present in the input. We use both an LLM-as-a-judge model and a semantic-similarity-based feature to identify such additions.

We compare the complete source segment against multiple candidate prefixes of the target. Starting with the complete target, we iteratively remove one token at a time from the end of the target and calculate the semantic similarity between each resulting candidate and the full source segment. We select the candidate with the highest similarity to the source. If the highest scoring candidate is shorter than the original target, we interpret the removed part as potentially detached, overgenerated

Table 3: Overall number and percentage of flagged segments for each system (Underlined systems are baseline systems, non-underlined are participants’ submissions).

System	Total
GPT OSS 120B	3485 (76.09%)
Tiny Aya Global	3219 (70.28%)
<u>GPT OSS 20B</u>	2959 (64.61%)
<u>Tower 9B</u>	2804 (61.22%)
<u>Qwen 3.6 - 27b</u>	2766 (60.39%)
<u>Ministral 3 - 14B</u>	2543 (55.52%)
<u>Gemini 3.1 Pro</u>	2534 (55.33%)
<u>Qwen 3.5 - 9b</u>	2478 (54.1%)
<u>GLM 4 - 9B</u>	1659 (36.22%)
<u>Command A+</u>	1324 (28.91%)
<u>DeepSeek V4 Pro</u>	1050 (22.93%)
<u>Mistral Medium 3.5</u>	377 (8.23%)
<u>GoogleTranslate</u>	374 (8.17%)
<u>Gemma 4 - 4B</u>	349 (7.62%)
UvA-MT	277 (7.56%)
<u>GPT 5.5</u>	327 (7.14%)
TartuNLP	60 (6.55%)
SalamandraTA	287 (6.27%)
<u>Gemma 4 - 31B</u>	254 (5.55%)
SRPOL	197 (5.38%)
Cohere CAT+	241 (5.26%)
CUNI-EdUKate	36 (3.93%)
HW-TSC	26 (2.84%)
CUNI-MH-v3	48 (2.62%)
Lumen	108 (2.36%)
TRIVE	96 (2.1%)
Qingqiu-MT-9B	85 (1.86%)
Wayfinder	76 (1.66%)
SCIR-TG-MT	23 (1.26%)
VoxNexus_V1	26 (0.95%)
Dubformer	24 (0.52%)

content. Such an output is therefore considered a fail. This criterion is complemented by the LLM-as-a-judge model, which independently identifies potentially detached portions of the target.

5 Evaluation results and analysis

5.1 Quantitative analysis

Table 3 presents aggregated evaluation results per system across all language pairs and all categories. We list the total number of flagged segments (where our evaluation heuristics found an irregularity), as well the percentage of flagged segments relative to the total number of segments. It is important to note that not all systems translated the test set into all languages. We also present an extended table 4 with results separated by system and by language pair.

The results show substantial differences in robustness across systems. The general-purpose LLMs included among the baselines exhibit relatively high rates of flagged segments on the chal-

lenge set. In particular, GPT OSS 120B, Tiny Aya Global, GPT OSS 20B, Tower-9B, and Qwen3.6-27B show rates above 60%, while Ministral 3-14B, Gemini 3.1 Pro, and Qwen3.5-9B fall in the range of approximately 54-56%. GLM4-9B and Command A+ perform better, with 36.22% and 28.91%, respectively, while DeepSeek V4 Pro has a rate of 22.93%.

A smaller group of general-purpose models performs substantially better on the challenge set. Mistral Medium 3.5, Gemma 4-4B, Gemma 4-31B, and GPT-5.5 all demonstrate rates below 10%, with Gemma 4-31B reaching 5.55%. These systems are therefore considerably more robust to the failure modes targeted by our test suite than the majority of the general-purpose LLM baselines.

The participating MT systems generally achieve lower failure rates than the general-purpose LLM baselines. Several submissions have failure rates below 5%, with the best-performing systems exhibiting failure rates close to or below 2%. This difference may partly be explained by the fact that many MT systems incorporate additional processing stages, such as quality estimation, filtering, or post-editing, which can detect and mitigate pathological outputs before they are returned to the user. However, this interpretation should be treated cautiously, as the participating systems may also differ from the general-purpose LLMs in model architecture, training data, decoding procedures, and other aspects of their translation pipelines.

Overall, the results indicate that general translation quality and robustness to the specific failure modes targeted by our challenge set do not necessarily coincide. Several systems that are capable of producing high-quality translations nevertheless exhibit substantial failure rates on these targeted inputs, suggesting that standard translation benchmarks may not fully capture these forms of pathological behavior. In the following analysis, we examine the results in greater detail by test category to identify which failure modes are most challenging for current systems.

5.2 Analysis by system and category

To gain a better understanding of the systems’ performance, we examine flagged segment rate by system and by test category. Tables 4 and 5 present the breakdown of flagged segments across the different categories and provide a more detailed view of the behaviors contributing to the overall rates.

A prominent pattern is that, for many of the

Table 4: Number and proportion of segments flagged by the evaluation criteria for each system.

System	Total	EN→RU	EN→DE	EN→ES	EN→CS	EN→ET
GPT OSS 120B	3485 (76.09%)	705 (76.97%)	710 (77.51%)	714 (77.95%)	678 (74.02%)	678 (74.02%)
Tiny Aya Global	3219 (70.28%)	752 (82.1%)	450 (49.13%)	790 (86.24%)	645 (70.41%)	582 (63.54%)
GPT OSS 20B	2959 (64.61%)	626 (68.34%)	601 (65.61%)	599 (65.39%)	570 (62.23%)	563 (61.46%)
Tower 9B	2804 (61.22%)	570 (62.23%)	594 (64.85%)	615 (67.14%)	454 (49.56%)	571 (62.34%)
Qwen 3.6 - 27b	2766 (60.39%)	621 (67.79%)	583 (63.65%)	574 (62.66%)	545 (59.5%)	443 (48.36%)
Ministral 3 - 14B	2543 (55.52%)	532 (58.08%)	603 (65.83%)	505 (55.13%)	397 (43.34%)	506 (55.24%)
Gemini 3.1 Pro	2534 (55.33%)	539 (58.84%)	525 (57.31%)	410 (44.76%)	581 (63.43%)	479 (52.29%)
Qwen 3.5 - 9b	2478 (54.1%)	562 (61.35%)	572 (62.45%)	538 (58.73%)	293 (31.99%)	513 (56.0%)
GLM 4 - 9B	1659 (36.22%)	243 (26.53%)	294 (32.1%)	287 (31.33%)	288 (31.44%)	547 (59.72%)
Command A+	1324 (28.91%)	198 (21.62%)	339 (37.01%)	282 (30.79%)	270 (29.48%)	235 (25.66%)
DeepSeek V4 Pro	1050 (22.93%)	233 (25.44%)	208 (22.71%)	190 (20.74%)	200 (21.83%)	219 (23.91%)
Mistral Medium 3.5	377 (8.23%)	86 (9.39%)	74 (8.08%)	77 (8.41%)	68 (7.42%)	72 (7.86%)
GoogleTranslate	374 (8.17%)	68 (7.42%)	66 (7.21%)	67 (7.31%)	73 (7.97%)	100 (10.92%)
Gemma 4 - 4B	349 (7.62%)	59 (6.44%)	61 (6.66%)	76 (8.3%)	74 (8.08%)	79 (8.62%)
UvA-MT	277 (7.56%)	96 (10.48%)	68 (7.42%)	NA	54 (5.9%)	59 (6.44%)
GPT 5.5	327 (7.14%)	71 (7.75%)	66 (7.21%)	59 (6.44%)	72 (7.86%)	59 (6.44%)
TartuNLP	60 (6.55%)	NA	NA	NA	NA	60 (6.55%)
SalamandraTA	287 (6.27%)	88 (9.61%)	48 (5.24%)	48 (5.24%)	49 (5.35%)	54 (5.9%)
Gemma 4 - 31B	254 (5.55%)	54 (5.9%)	50 (5.46%)	53 (5.79%)	49 (5.35%)	48 (5.24%)
SRPOL	197 (5.38%)	57 (6.22%)	49 (5.35%)	NA	45 (4.91%)	46 (5.02%)
Cohere CAT+	241 (5.26%)	53 (5.79%)	47 (5.13%)	42 (4.59%)	48 (5.24%)	51 (5.57%)
CUNI-EdUKate	36 (3.93%)	NA	36 (3.93%)	NA	NA	NA
HW-TSC	26 (2.84%)	NA	NA	NA	26 (2.84%)	NA
CUNI-MH-v3	48 (2.62%)	NA	24 (2.62%)	NA	24 (2.62%)	NA
Lumen	108 (2.36%)	48 (5.24%)	15 (1.64%)	17 (1.86%)	14 (1.53%)	14 (1.53%)
TRIVE	96 (2.1%)	20 (2.18%)	14 (1.53%)	27 (2.95%)	17 (1.86%)	18 (1.97%)
Qingqiu-MT-9B	85 (1.86%)	20 (2.18%)	15 (1.64%)	17 (1.86%)	16 (1.75%)	17 (1.86%)
Wayfinder	76 (1.66%)	19 (2.07%)	12 (1.31%)	14 (1.53%)	16 (1.75%)	15 (1.64%)
SCIR-TG-MT	23 (1.26%)	20 (2.18%)	NA	NA	NA	3 (0.33%)
VoxNexus_V1	26 (0.95%)	2 (0.22%)	14 (1.53%)	10 (1.09%)	NA	NA
Dubformer	24 (0.52%)	8 (0.87%)	3 (0.33%)	4 (0.44%)	5 (0.55%)	4 (0.44%)

baseline and organizer systems, a substantial proportion of the observed failures comes from the *sensitive content* category. This is not entirely surprising, as general-purpose LLMs commonly incorporate safety mechanisms designed to prevent them from engaging with content that may be considered harmful, unlawful, or ethically problematic. While such behavior may be desirable in a general-purpose conversational setting, it constitutes a failure in the machine translation setting, where the expected behavior is to translate the input faithfully rather than act on or respond to its content.

At the category level, we observe that most systems, baselines and participants alike, struggle to some extent with both *truncated inputs* and *instruction inputs*. The difficulty with truncated inputs may be related to the tendency of LLMs to interpret incomplete sentences as prompts requiring completion rather than as text that should simply be translated as provided. Similarly, instruction inputs can activate the models’ instruction-following behavior, causing them to execute the instruction or respond to it rather than treating it as the object of translation. These two categories therefore probe a common underlying challenge: distinguishing the

translation task from other behaviors that the same models are trained to perform.

It is also interesting to note that several categories appear to pose less of a challenge for participants’ systems, which are translation-specific rather than general-purpose systems. Comparing the rates of flagged segments for the *DNT*, *punctuation*, and *emoji* categories in Table 5, we observe substantial reductions in the average rates, from 37.86% to 3.57% for *DNT*, from 60.77% to 14.82% for punctuation, and from 41.25% to 6.67% for emoji. For the *truncated* and *instructions* categories, however, the reduction is less pronounced, suggesting that these input types remain challenging for both task-specific and general-purpose systems. In particular, the average rate decreases from 37.85% to 16.93% for truncated segments and from 87.96% to 46.12% for instructions.

6 Conclusion

Overall, the category-level analysis suggests that the failures identified by our challenge set are not homogeneous. Some failures correspond to complete refusal or detachment from the source, while others involve otherwise valid translations accom-

panied by additional, unsupported content. The results therefore highlight the value of examining failure types in addition to aggregate failure rates when assessing the robustness of LLM-based translation systems.

References

- Aisha Alansari and Hamzah Luqman. 2026. [Large language models hallucination: A comprehensive survey](#). *Preprint*, arXiv:2510.06265.
- Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. [Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic](#). *Preprint*, arXiv:2402.11746.
- Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. [Hallucinations in large multilingual translation models](#). *Transactions of the Association for Computational Linguistics*, 11:1500–1517.
- Zdeněk Kasner, Vilém Zouhar, Patrícia Schmidtová, Ivan Kartáč, Kristýna Onderková, Ondřej Platek, Dimitra Gkatzia, Saad Mahamood, Ondřej Dusek, and Simone Balloccu. 2026. [LLMs as span annotators: A comparative study of LLMs and humans](#). In *Proceedings of the First Workshop on Multilingual Multicultural Evaluation*, pages 1–22, Rabat, Morocco. Association for Computational Linguistics.
- Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakouгна, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. [Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.
- Monishwaran Maheswaran, Marco Carini, Christian Federmann, and Tony Diaz. 2025. [TASER: Translation assessment via systematic evaluation and reasoning](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1004–1010, Suzhou, China. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). *CoRR*, abs/1908.10084.
- Yusen Zhang, Sarkar Snigdha Sarathi Das, and Rui Zhang. 2025. [Demystify verbosity compensation behavior of large language models](#). In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertainLP 2025)*, pages 160–178, Suzhou, China. Association for Computational Linguistics.

Table 5: Number and proportion of flagged segments: baselines and participants

System	Total	DNT	Punct.	Short.	Repeat	Emoji	Instructions	Truncated	Sensitive
GPT OSS 120B	3485 (76.09%)	497 (55.84%)	78 (60.0%)	94 (28.48%)	13 (13.0%)	6 (12.0%)	93 (97.89%)	186 (79.15%)	2518 (91.56%)
Tiny Aya Global	3219 (70.28%)	709 (79.66%)	117 (90.0%)	68 (20.61%)	56 (56.0%)	22 (44.0%)	91 (95.79%)	78 (33.19%)	2078 (75.56%)
GPT OSS 20B	2959 (64.61%)	466 (52.36%)	102 (78.46%)	52 (15.76%)	16 (16.0%)	14 (28.0%)	95 (100.0%)	151 (64.26%)	2063 (75.02%)
Tower 9B	2804 (61.22%)	409 (45.96%)	117 (90.0%)	17 (5.15%)	2 (2.0%)	45 (90.0%)	90 (94.74%)	109 (46.38%)	2015 (73.27%)
Qwen 3.6 - 27b	2766 (60.39%)	369 (41.46%)	103 (79.23%)	16 (4.85%)	5 (5.0%)	20 (40.0%)	92 (96.84%)	75 (31.91%)	2086 (75.85%)
Ministral 3 - 14B	2543 (55.52%)	630 (70.79%)	130 (100.0%)	65 (19.7%)	13 (13.0%)	42 (84.0%)	95 (100.0%)	68 (28.94%)	1500 (54.55%)
Gemini 3.1 Pro	2534 (55.33%)	277 (31.12%)	12 (9.23%)	215 (65.15%)	53 (53.0%)	3 (6.0%)	82 (86.32%)	165 (70.21%)	1727 (62.8%)
Qwen 3.5 - 9b	2478 (54.1%)	381 (42.81%)	100 (76.92%)	11 (3.33%)	3 (3.0%)	14 (28.0%)	94 (98.95%)	87 (37.02%)	1788 (65.02%)
GLM 4 - 9B	1659 (36.22%)	798 (89.66%)	128 (98.46%)	30 (9.09%)	13 (13.0%)	50 (100.0%)	94 (98.95%)	113 (48.09%)	433 (15.75%)
Command A+	1324 (28.91%)	188 (21.12%)	36 (27.69%)	9 (2.73%)	6 (6.0%)	2 (4.0%)	78 (82.11%)	134 (57.02%)	871 (31.67%)
DeepSeek V4 Pro	1050 (22.93%)	141 (15.84%)	56 (43.08%)	11 (3.33%)	3 (3.0%)	13 (26.0%)	95 (100.0%)	58 (24.68%)	673 (24.47%)
Mistral Medium 3.5	377 (8.23%)	93 (10.45%)	101 (77.69%)	9 (2.73%)	2 (2.0%)	18 (36.0%)	95 (100.0%)	30 (12.77%)	29 (1.05%)
GoogleTranslate	374 (8.17%)	249 (27.98%)	26 (20.0%)	3 (0.91%)	3 (3.0%)	0 (0.0%)	4 (4.21%)	83 (35.32%)	6 (0.22%)
Gemma 4 - 4B	349 (7.62%)	115 (12.92%)	65 (50.0%)	12 (3.64%)	0 (0.0%)	40 (80.0%)	90 (94.74%)	15 (6.38%)	12 (0.44%)
GPT 5.5	327 (7.14%)	36 (4.04%)	4 (3.08%)	9 (2.73%)	1 (1.0%)	1 (2.0%)	93 (97.89%)	56 (23.83%)	127 (4.62%)
Gemma 4 - 31B	254 (5.55%)	33 (3.71%)	89 (68.46%)	5 (1.52%)	1 (1.0%)	40 (80.0%)	56 (58.95%)	15 (6.38%)	15 (0.55%)
Average	1781.38 (38.89%)	336.94 (37.86%)	79.00 (60.77%)	39.12 (11.86%)	11.88 (11.88%)	20.62 (41.25%)	83.56 (87.96%)	88.94 (37.85%)	1121.31 (40.78%)
UvA-MT	277 (7.56%)	52 (7.3%)	50 (48.08%)	4 (1.52%)	4 (5.0%)	0 (0.0%)	45 (59.21%)	49 (26.06%)	73 (3.32%)
TartuNLP	60 (6.55%)	21 (11.8%)	9 (34.62%)	1 (1.52%)	0 (0.0%)	10 (100.0%)	7 (36.84%)	12 (25.53%)	0 (0.0%)
SalamandraTA	287 (6.27%)	84 (9.44%)	50 (38.46%)	4 (1.21%)	1 (1.0%)	0 (0.0%)	76 (80.0%)	70 (29.79%)	2 (0.07%)
SRPOL	197 (5.38%)	23 (3.23%)	24 (23.08%)	6 (2.27%)	6 (7.5%)	0 (0.0%)	64 (84.21%)	70 (37.23%)	4 (0.18%)
Cohere CAT+	241 (5.26%)	31 (3.48%)	2 (1.54%)	8 (2.42%)	7 (7.0%)	0 (0.0%)	75 (78.95%)	110 (46.81%)	8 (0.29%)
CUNI-EdUKate	36 (3.93%)	6 (3.37%)	7 (26.92%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	19 (100.0%)	4 (8.51%)	0 (0.0%)
HW-TSC	26 (2.84%)	0 (0.0%)	5 (19.23%)	1 (1.52%)	0 (0.0%)	0 (0.0%)	8 (42.11%)	11 (23.4%)	1 (0.18%)
CUNI-MH-v3	48 (2.62%)	4 (1.12%)	7 (13.46%)	2 (1.52%)	2 (5.0%)	0 (0.0%)	27 (71.05%)	6 (6.38%)	0 (0.0%)
Lumen	108 (2.36%)	44 (4.94%)	6 (4.62%)	7 (2.12%)	1 (1.0%)	0 (0.0%)	36 (37.89%)	14 (5.96%)	0 (0.0%)
TRIVE	96 (2.1%)	25 (2.81%)	6 (4.62%)	8 (2.42%)	1 (1.0%)	0 (0.0%)	24 (25.26%)	21 (8.94%)	11 (0.4%)
Qingqiu-MT-9B	85 (1.86%)	27 (3.03%)	3 (2.31%)	6 (1.82%)	2 (2.0%)	0 (0.0%)	24 (25.26%)	20 (8.51%)	3 (0.11%)
Wayfinder	76 (1.66%)	13 (1.46%)	1 (0.77%)	8 (2.42%)	0 (0.0%)	0 (0.0%)	28 (29.47%)	21 (8.94%)	5 (0.18%)
SCIR-TG-MT	23 (1.26%)	5 (1.4%)	0 (0.0%)	3 (2.27%)	0 (0.0%)	0 (0.0%)	1 (2.63%)	6 (6.38%)	8 (0.73%)
VoxNexus_V1	26 (0.95%)	1 (0.19%)	0 (0.0%)	4 (2.02%)	1 (1.67%)	0 (0.0%)	9 (15.79%)	9 (6.38%)	2 (0.12%)
Dubformer	24 (0.52%)	0 (0.0%)	6 (4.62%)	3 (0.91%)	0 (0.0%)	0 (0.0%)	3 (3.16%)	12 (5.11%)	0 (0.0%)
Average	107.33 (3.41%)	22.40 (3.57%)	11.73 (14.82%)	4.33 (1.73%)	1.67 (2.08%)	0.67 (6.67%)	29.73 (46.12%)	29.00 (16.93%)	7.80 (0.37%)