

How Many Raters Do We Need to Recognize a Bad Translation via Perplexity?

Bc. Antonín Otmar

Faculty of Mathematics and Physics, Charles University
tonda.otmar.x@gmail.com

prof. RnDr. Ondřej Bojar, PhD.

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
bojar@ufal.mff.cuni.cz

Abstract

The perplexity assigned by a large language model (LLM) to a translation is a cheap, reference-free signal of translation quality, but the judgements of any particular model are likely to include noise based on its specific weights. We treat each LLM as one *rater* of English to Czech machine-translation output and ask if we can use their pooled judgements to reliably separate good translations from faulty ones. We measure the AUROC, which reflects the separation of good and bad translations, of all possible ensembles of 13 open-weight raters on two error types: morphologically *broken* sentences and semantically *corrupted* ones — both types both from human references and LLM generated translations, giving a total of 4 classes. With the raters’ own machine translations as positives, the approach is reasonably successful — a well-chosen pair reaches a held-out AUROC of 0.887 and the optimum of 0.890 needs only *three* raters, after which further raters only erode the best score. In the second experiment we replace the machine-translated positives with human reference translations and the signal collapses for every damaged class written by LLMs: the best ensemble reaches only 0.727 and we see that when comparing human references against semantically damaged machine translations, most ensembles assign higher perplexities to the human references. Perplexity of two to three LLM raters is thus good for distinguishing good and bad machine translations but falls short for human translations.

1 Introduction

Reference-free estimation of machine-translation (MT) quality is attractive wherever references are unavailable or costly to obtain. A particularly cheap signal is *perplexity*: a fluent and well formed translation should be more probable under a language model of the target language than a damaged one. Furthermore, for a multilingual language

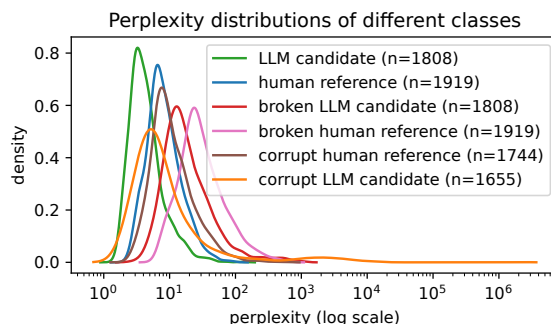


Figure 1: Perplexity densities (dolphin model, log scale) for good machine and human translations and their degraded versions: morphologically *broken* and semantically *corrupt*. The classes overlap heavily, motivating model ensembling. Human references score worse than semantically corrupt machine translations.

model with the context of source-translation pair (en: source sentence, cs: translated sentence), an adequate translation should be more probable than an inadequate one. Perplexity can be obtained from any open-weight LLM that has been trained on both languages of the language pair we’re interested in.

The weakness of the signal is equally obvious: it is the opinion of a single model, shaped by that model’s training data, tokenizer, and coverage of the target language. A straightforward approach to remedy is to use an ensemble of models (Dietrich, 2000). We therefore treat every LLM as one *rater* and pool the raters’ judgements to obtain the ultimate score. We ask how many raters to pick and how does our choice when picking them matter.

We answer this question exhaustively in an English–Czech setting. Starting from a collection of translations and a collection of systematically degraded variants, we score every candidate translation with 13 open-weight LLMs and evaluate every one of the $2^{13} - 1 = 8191$ possible rater ensembles.

To examine the tendency of LLMs to prefer LLM

output to human one, we conduct two experiments, one with LLM-generated candidate translations and one with manual references.

Our contributions are:

- An exhaustive evaluation of all 8,191 perplexity-rater ensembles over 13 open-weight LLMs on Czech MT output, for two error types (morphological and semantic) and their union, using a 10-fold rotating holdout.
- The same enumeration with about 38 human references per source sentence as the positive class. Importantly, these diverse human references obtain *higher* perplexity than semantically corrupt machine translations and the best of all the 8k ensembles reaches a held-out AUROC of only 0.727.
- A quantitative answer to the title question: the optimum for the examined language pair is a triple (llama, dolphin, gptoss) at 0.890, the best pair is within 0.003 of it, and the full 13-rater pool (0.848) is worse than the most successful singleton (0.876).
- Evidence that rater *composition* dominates rater *count*: the marginal value of a rater is not always predicted by its singleton score, nearly half the raters subtract value from the ensembles they join, and indiscriminate pooling converges to a mid pack performance rather than becoming better than any of its parts in isolation.

2 Related Work

MT quality estimation. Reference-based metrics such as BLEU (Papineni et al., 2002) and chrF (Popović, 2015) require references and correlate imperfectly with human judgement (Mathur et al., 2020). Learned metrics — BERTScore (Zhang et al., 2020), COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020) — improve correlation at the cost of task-specific training, and reference-free quality estimation removes the reference requirement (Fomicheva et al., 2020; Ranasinghe et al., 2020). LLMs themselves have been used as direct quality judges (Kocmi and Federmann, 2023).

Perplexity as a quality signal. Language-model perplexity is an established filter for noisy corpora (Wenzek et al., 2020), and cross-entropy difference has long been used for data selection (Moore and Lewis, 2010; Junczys-Dowmunt, 2018).

Multiple raters. In human MT evaluation, aggregating several annotators is standard practice and the required number of raters is an explicit design decision (Freitag et al., 2021). Ensembles of learned models reduce variance when members err independently (Dietterich, 2000); our setting tests how much does this logic hold when the “raters” are frozen LLMs whose judgements are correlated by shared training data.

Scoring in isolation vs. choosing from a list. Picking the good translation out of several candidates is a multiple-choice question, and how it is put to an LLM matters. Robinson et al. (2023) contrast *cloze prompting*, where each option is scored on its own by the model’s likelihood, with *multiple choice prompting*, where all options share the context and the model names its pick — the multiple choice prompting being substantially stronger; closed and open answer formats differ for evaluation more generally (Myrzakhan et al., 2024). Our raters are squarely on the cloze side: a candidate is scored without seeing its competitors, which is what makes enumerating 8,191 ensembles affordable, but no rater here ever compares two translations directly.

3 Task and Data

3.1 Source corpus

In order to reasonably represent the *variance* of possible translations and avoid excessive cost, we needed an existing test set with a diverse collection of human references. A good option for this was the DepRefSet dataset of very many Czech references from 50 English source sentences from WMT11 data (Bojar et al., 2013). In DepRefSet, each of the source sentences is equipped with thousands to hundreds of thousands of manually revised possible translations. This makes our second experiment possible at all: it gives us a human positive class of the same order of magnitude as the machine-generated one, over exactly the same source content.

Czech as the target language is also a good choice because it is a morphologically rich language with good language processing tools that are allowed to reliably damage the good translations, see below.

LLM candidate translation To obtain a corpus of candidate LLM translations, the English source sentences from DepRefSet are translated

into Czech by open-weight LLMs — the same that we will use as raters, see Table 1 — with a few-shot prompt (see Appendix A) asking the LLM to provide the translations and demonstrating good translation pairs, giving the *LLM candidate* class.

Broken translation (morphological damage).

Czech translations from the *LLM candidate* class are first analyzed using the Czech morphological analyzer MorphoDiTa (Straková et al., 2014), then for randomly chosen content words, one morphological feature of the positional tag — gender, number, case, or person — is replaced and the surface form is regenerated. Every word remains a valid Czech form with a meaning that fits the context, but agreement with its context is violated.

We obtain these syntactically damaged sentences from both the reference translations in DepRefSet which are already in Czech and the *LLM candidate* class described above to get two separate classes *broken LLM candidate* and *broken human reference*.

Corrupt translation (semantic damage). To create semantically damaged sentences, an LLM is given a reference translation and prompted, with few-shot examples showing several rewritten sentences (for example: “The morning was cold, but sunny” → “The morning was neither cold nor rainy”), to produce rewrites that are grammatically well-formed but inadequate translations of the source sentence. This corruption is carried out monolingually, in Czech only, see the full prompt in Appendix B.

Applying this process we get the two classes *corrupt LLM candidate* and *corrupt human reference*.

After using the LLM-based methods to obtain the *LLM candidate* and *corrupt* candidate sets, we removed faulty sentences from them by filtering out empty lines and sentences in which the Czech morphological analyzer didn’t find any Czech words (typically LLM explanations of their answers) and then manually removing sentences where the model obviously misunderstood the task. After this step we generated the *broken* set from the *LLM candidate* set. We got about 3 sentences per class, model and source sentence.

This way we obtained 1,808 LLM candidates, 1,919 human references, 1,919 broken human references, 1,808 broken LLM candidates, 1,744 corrupt human references and 1,655 corrupt LLM generated candidates.

Short name	Checkpoint
gemmabig	google/gemma-3-12b-it
ayaexp	CohereForAI/aya-expense-8b
commandr	c4ai-command-r7b-12-2024
eu	EuroLLM-9B-Instruct
qwen	Qwen2.5-7B-Instruct
mistral7b	Mistral-7B-Instruct-v0.3
llama	Llama-3.1-8B-Instruct
dolphin	Dolphin-Mistral-24B-Venice
darwin	Darwin-36B-Opus
ibm	ibm-granite/granite-4.1-30b
gptoss	openai/gpt-oss-20b
aya	CohereLabs/tiny-aya-global
tower	Unbabel/Tower-Plus-9B

Table 1: Open-weight LLMs used in this experiment.

3.2 Raters

Table 1 lists the 13 open-weight LLMs used as raters. They span 3.35B–36B parameters, multilingual and English-centric training, and several model families.

4 Method

4.1 Ensembles of perplexity raters

Given a sentence s and a rater m with parameters θ_m , the rater’s raw score is the perplexity

$$\text{PPL}_m(s) = \exp\left(-\frac{1}{|s|} \sum_{t=1}^{|s|} \log p_{\theta_m}(w_t \mid w_{<t})\right). \quad (1)$$

While the perplexity is only summed over the tokens of the target sentence, the rater model is given each sentence in the context of a good few shot translation prompt including the appropriate source sentence. Tokens of the prompt do not themselves contribute to the perplexity sum but they have impact on the perplexity assigned to the tokens of the target sentence.

Different raters produce log-perplexities on very different scales, so before pooling we standardize each rater: the rater’s judgement of a sentence is its z-scored log-perplexity

$$z_m(s) = \frac{\log \text{PPL}_m(s) - \mu_m}{\sigma_m}, \quad (2)$$

where μ_m and σ_m are the mean and standard deviation of $\log \text{PPL}_m$ over the training sentences of all classes (Section 4.2), so that every rater’s judgements have zero mean and unit variance on the training data. An ensemble $E \subseteq \mathcal{M}$ of raters scores the sentence with the sum of its members’ z-scores,

$$\text{score}_E(s) = \sum_{m \in E} z_m(s), \quad (3)$$

i.e. an unweighted pooling of the raters’ standardized log-judgements.

4.2 10-fold cross-validation

Evaluating 8,191 candidate ensembles and reporting the maximum invites the winner’s curse: with thousands of comparisons, the top score is biased upward and the identity of the winner may be an artifact of noise. We therefore split the corpus, estimate the best ensembles and the marginal values of each rater using only a training partition and then evaluate the results on the held-out data.

Importantly, we make this split at the level of the source sentences and project it onto the candidate sentences generated from them to avoid contaminating the held-out partition with close variation of sentences in the training partition.

We utilize a 10-fold cross-validation approach. The seeded permutation of the 50 sentences is cut into $N = 10$ blocks of five, each block is the development side of exactly one fold and the training side of nine, and the whole experiment is re-run ten times from scratch. A reported figure is the mean over all ten folds; because the folds partition the corpus, every unit contributes to exactly one development estimate, and the pooled development number is an average over the whole corpus.

4.3 Separation measure

We measure how well an ensemble’s score separates reference sentences from degraded sentences with the AUROC: the probability that a randomly drawn degraded sentence receives a higher (worse) score than a randomly drawn reference one (Hanley and McNeil, 1982). We compute it from the Mann–Whitney U statistic (Mann and Whitney, 1947), with ties counted as $\frac{1}{2}$; 0.5 means no discriminative power and 1.0 perfect separation.

To quantify uncertainty, we estimate AUROC using subsampling: within each split we repeatedly ($500\times$) draw 400 reference and 400 degraded scores (with replacement) and compute the AUROC of the draw. Alongside it we report the *fold spread*: the standard deviation of the ten per-fold means. The two answer different questions and differ by a factor of 1.6–1.9. (Table 2). The subsample interval says how precisely this quantity is measured on the units we have; the fold spread says how much the answer moves if the source sentences change to different ones.

	Ensemble	Subsample	Fold sd
E1	dolphin	$\pm.062$	.038
E1	dolphin,gptoss	$\pm.039$	.022
E1	llama,dolphin,gptoss	$\pm.041$	.024
E1	all 13	$\pm.057$	.034
E2	gptoss	$\pm.111$	.058
E2	dolphin,gptoss	$\pm.077$	.043
E2	all 13	$\pm.086$	.047

Table 2: Two measures of uncertainty on combined development AUROC: the half-width of the 90% interval over all 10×500 subsamples, and the standard deviation of the ten per-fold means. Across the folds, the Experiment 1 (E1) *optimum* ranges from 0.853 to 0.921 and the Experiment 2 (E2) optimum from 0.649 to 0.788. Differences smaller than the fold spread should not be read as differences between ensembles.

4.4 Exhaustive enumeration

We evaluate every non-empty subset of the 13 raters, i.e. 8,191 ensembles, for each of the damaged classes individually and their union (*combined*) using either the *LLM candidate* or the *human reference* class as the reference class. Since every ensemble is evaluated on the same sentence pool (Section 3.1), all scores below are directly comparable across ensemble sizes and compositions. Unless stated otherwise, the numbers we quote are holdout AUROCs of ensembles chosen on the training split.

5 Results

5.1 Experiment 1: Machine translation as the positive class

5.1.1 Singleton raters

Table 3 shows each rater’s singleton AUROC on the held-out split. Individual raters span 0.764–0.876 on the combined task: the spread between the best rater (dolphin, 0.876) and the weakest (ibm, 0.764) is more than twice the fold spread of either, so rater choice clearly has a significant impact. The four negative classes separate cleanly and in the same order for every rater without exception: broken human references are easiest (0.879–0.982), followed by broken machine candidates (0.803–0.942), then corrupted human references (0.762–0.883), and corrupted machine candidate are the hardest (0.517–0.698). They tell a clear story that perplexity is better as a *fluency* detector than as an *adequacy* detector and that it is a detector of authorship assigning higher probabilities to sentences written by machines.

Rater	BM	BH	CM	CH	Comb.	Marg.
dolphin	0.941	0.974	0.698	0.867	0.876	+0.011
llama	0.942	0.982	0.666	0.835	0.862	+0.008
commandr	0.928	0.978	0.645	0.842	0.855	+0.007
mistral7b	0.926	0.980	0.598	0.863	0.849	+0.005
gptoss	0.928	0.977	0.692	0.769	0.848	+0.019
qwen	0.906	0.969	0.593	0.883	0.845	+0.005
darwin	0.921	0.964	0.608	0.805	0.832	−0.001
eu	0.919	0.968	0.566	0.800	0.821	−0.004
aya	0.894	0.961	0.589	0.806	0.819	−0.004
tower	0.881	0.936	0.619	0.762	0.806	−0.007
ayaexp	0.874	0.963	0.536	0.819	0.806	−0.008
gemmabig	0.856	0.926	0.517	0.765	0.774	−0.015
ibm	0.803	0.879	0.580	0.770	0.764	−0.017

Table 3: Experiment 1: cross-validated singleton AUROC per rater and class, sorted by the combined score. *BM* is morphological damage to a machine candidate, *BH* is morphological damage to a human reference, *CM* the LLM’s corruption of a machine candidate, *CH* its corruption of a human reference. Fold spreads are ~ 0.04 (BM), ~ 0.03 (BH), ~ 0.04 (CM), ~ 0.08 (CH) and ~ 0.04 (combined). *Marg.* is the rater’s size-matched marginal contribution to combined dev AUROC when added to an ensemble (Section 5.1.1).

The last column of Table 3 reports each rater’s size-matched marginal value: the mean combined AUROC of all ensembles containing the rater minus that of all equally sized ensembles without it. Only six raters have positive marginal value (a seventh, *darwin*, is essentially neutral), and one rater notably differs in its marginal rating from its singleton ranking: *gptoss* (+0.019) consistently contributes the most despite ranking only fifth of thirteen as a singleton — its judgements complement the others (Section 5.1.5).

5.1.2 AUROC as a function of ensemble size

Figure 2 summarises the central result. For each size k we plot the distribution of combined dev AUROC over all size- k ensembles (median and interquartile band) together with the dev scores of the best and worst ensembles of that size *as selected on the training split*.

Three interesting observations are as follows: First, the *best* ensembles peak almost immediately (Table 4): from 0.876 ($k=1$) to 0.887 ($k=2$), with the optimum of 0.890 already at $k=3$ using the raters (llama, dolphin, gptoss). From there the best achievable score starts to *decline* all the way down to 0.848 for the full 13-rater sum, which would place it fifth or sixth of fourteen if it was a singleton rater. The best pair already delivers about 80% of the total gain between the best singleton rater and the optimum.

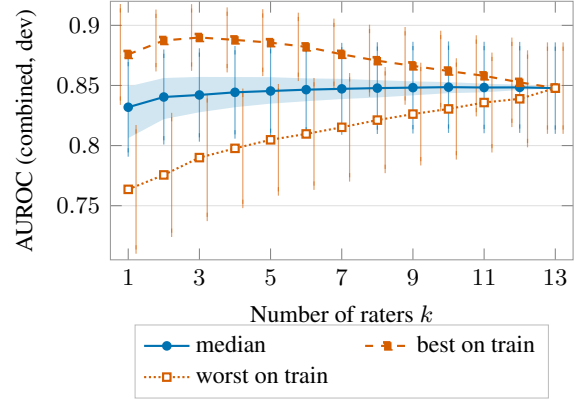


Figure 2: Experiment 1: cross-validated combined dev AUROC versus ensemble size k : the size- k ensembles that are best (dashed) and worst (dotted) on the *training* split, evaluated on held-out sentences, and the median over all size- k ensembles with the interquartile band (solid; shaded). Whiskers are the fold spread — one standard deviation of the plotted quantity over the ten folds (0.022–0.050 here) — while the shaded band is the spread across ensembles at that size: the band narrows with k , the whiskers do not.

Second, the *median* ensemble improves only from 0.832 at $k=1$ to 0.848 at $k=13$ and never reaches the performance of the best singleton rater. We see a tendency towards mediocre results — the whole distribution narrows toward the full-pool score of 0.848 — pooling avoids the very weak results but it avoids the very strong ones as well.

Third, the selections done on training partition *generalize* to held-out sentences: the ranking of all 8,191 ensembles on the training split carries over to the development split almost unchanged (Spearman $\rho = 0.998$), and the triple selected on the training split is *exactly* the best ensemble found anywhere on held-out content (llama, dolphin, gptoss, 0.890). All of these differences are however very small in the context of the $\sim \pm 0.024$ fold spread: the bests of sizes 1–6 are mutually indistinguishable. The consistent shape of the curves across sizes is what supports the trend, not any single pairwise comparison.

5.1.3 Four error classes

Figure 3 splits the analysis into four classes:

Broken human references: the easiest class, and the one control that holds. Morphological damage to a *human* reference is the easiest of the four classes for every rater without exception: singletons span 0.879–0.982, the weakest rater on this class (ibm, 0.879) beats the best rater on *cor-*

k	Best ensemble (train)	Train	Dev	sd
1	dolphin	0.870	0.876	.038
2	+ gptoss	0.887	0.887	.022
3	+ llama	0.888	0.890	.024
4	+ qwen	0.886	0.888	.024
5	+ commandr	0.884	0.886	.024
6	+ mistral7b	0.881	0.882	.025
13	all thirteen raters	0.845	0.848	.034

Table 4: Experiment 1: best ensemble per size, selected by combined AUROC on the training splits, with its cross-validated score on both sides and the standard deviation of its ten per-fold dev means. Every row extends the row above by one rater — the chain is nested here, which it need not be — and coincides with what greedy forward selection picks.

rupt LLM candidate by 0.18, and the best pair (llama+gptoss) reaches 0.993. In this class the singleton picked on the training split is not the one that turns out best on held-out data, so the $k=1$ point of Figure 3 sits at 0.980 rather than at the 0.982 of Table 3. The optimum is a triple (commandr, qwen, gptoss) at 0.994, after which the usual decline sets in, to 0.976 for the full pool. The comparison with *broken candidate* is the informative one. The two classes are the same operation applied to differently authored text, and the human-authored arm is *easier* at every size, because morphological corruption of an already-improbable human sentence pushes it further from the model’s expectations than the same corruption of a fluent machine translation does.

Broken machine candidates: one complementary pair suffices. Every rater detects morphological damage well (singletons 0.80–0.94), so the signal is broadly shared, and one complementary pair — dolphin+gptoss scoring 0.967 — is effectively the best ensemble of *any* size, an improvement of +0.025 over the best singleton llama (0.942) Adding this best singleton to the ensemble yields further 0.002 (0.969) and after that the best size- k score declines monotonically to 0.930 at $k=13$.

Corrupt human references: the signal lives in one rater. Semantic corruption is a different story: qwen (0.883) beats the second best (dolphin, 0.867) by a margin of 0.16, and the best size- k corrupt ensemble contains qwen at *every* size. The best ensemble selected on the training split is qwen, mistral7b, dolphin, gptoss at $k=4$ (dev 0.887) — a mere +0.004 over qwen alone,

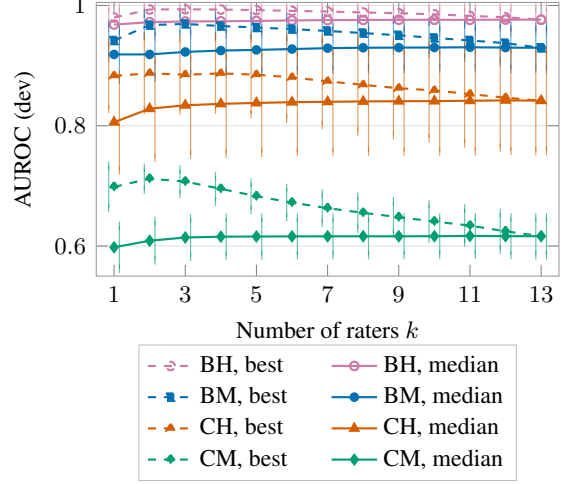


Figure 3: Experiment 1: best (dashed) and median (solid) dev AUROC per ensemble size for the four error types; best is selected on the training splits. *BH* is the morphologically broken human reference, *BM* a broken machine translation. *CH* is the LLM’s corrupted rewrite of a human reference, *CM* its corrupted write of the machine’s own translation. The four sit at clearly separated levels that no amount of pooling closes, and all four peak at $k \leq 4$; what adding raters buys is the convergence of best and median, not a higher ceiling.

and indistinguishable on held-out data from the best pair qwen, dolphin, which the training split ranks lower but which reaches the same 0.887 in Figure 3. From $k=5$ the best ensemble declines steadily to 0.842. It seems our ceiling on adequacy detection is limited by the rater capabilities, not by rater noise: pooling cannot manufacture a signal that most raters simply lack.

Corrupt LLM candidates: hardest, one signal masks other. The model’s corruption of its own translation is by a wide margin the hardest class, with singletons at 0.517–0.698 and a ceiling of 0.712 (dolphin+gptoss at $k=2$). The damage is the same rewriting operation that produces *corrupt*, which every rater except gptoss detects 0.14–0.29 AUROC better. What differs is the input’s authorship. A corrupted human reference is a non-model-typical sentence with a semantic mistake on top, while a corrupted machine translation is more model-typical, which is a signal that masks the semantic inadequacy by working in the opposite direction.

5.1.4 Reliability of larger ensembles

Although the best-performing ensembles are small, larger ensembles are *safer*: the worst-on-train size- k ensemble (the dotted curve in Figure 2) improves

from 0.764 ($k=1$) to 0.839 ($k=12$) on the dev split. Size buys insurance but not performance — the floor rises, but the ceiling falls.

A simple approach of greedy forward selection on the training split (add the rater that most improves combined train AUROC) finds the train-optimal triple in three steps: it picks dolphin, then gptoss, then llama (train 0.888, dev 0.890) and peaks there, which is the same triple and the same score the exhaustive search finds.

5.1.5 Rater correlation

Table 5 reports the Spearman rank correlation between the per-sentence perplexities of every rater pair. Correlations are computed within each class separately and then averaged with sentence-count weights: pooling the classes first would let the shared reference-versus-degraded signal, which every competent rater carries, inflate all pairwise correlations, whereas the within-class correlation asks the question that actually matters for ensembling — do two raters find the *same* sentences suspicious?

The matrix splits the pool into twelve like-minded raters and one outlier. Every pair among the twelve non-gptoss raters correlates between .69 and .90 (median .82): whatever their family, size, or training mix, these raters largely rank the same sentences the same way. gptoss, by contrast, correlates at most .39 with any other rater (median .305). This substantiates our reading of Table 3: gptoss is only a mid-table singleton, yet the most valuable rater to add, because it contributes genuinely new information rather than another copy of the consensus.

5.2 Experiment 2: Human references as the Positive class

Experiment 1 asked the raters to tell good *machine* translations from damaged ones. Experiment 2 changes exactly one thing: the positive class consists of human reference translations per source sentence instead of the LLMs’ own output. The rest of the experiment setup — the negative classes, the raters, the prompts, the pooling rule and the AUROC protocol are all unchanged.

5.2.1 The performance worsens considerably

Table 6 gives the singleton scores. Every number is dramatically worse than its Experiment 1 counterpart. The best single rater on the combined task, gptoss, reaches 0.707 against 0.876 for the best rater in Experiment 1, and the weakest (gemmabig,

0.556) is barely above chance.

When we look at the breakdown by error type, the reason becomes clear. On *broken human references* — morphologically damaged human references, judged against undamaged human references — the performance stays reasonable: the raters span 0.74–0.93, the best ensemble of any size reaches 0.954, and even the full 13-rater pool scores 0.911. On morphologically *broken LLM candidate* sentences a real if weakened signal survives: the raters span 0.59–0.83 and the pool still detects that agreement violations make Czech improbable. On *corrupt human reference* sentences — the LLM’s rewrites of human references — the signal is nearly gone: the raters span 0.48–0.63, the best ensemble of any size reaches 0.631, and that best ensemble is a singleton (qwen), the same rater that leads this class in Experiment 1. On *corrupt LLM candidate* the signal does not just weaken but reverse. LLMs prefer inadequate machine translations to adequate human ones and twelve of the thirteen raters score below 0.5, and so do 8,190 of the 8,191 ensembles; the median ensemble sits at 0.339 and the best reaches 0.501 — gptoss alone, at essentially chance.

5.2.2 Ensembling does not significantly help

Figure 4 and Table 7 repeat the size analysis. The shape of the curves is recognizably the same as in Experiment 1 — a fast rise to a small optimum, a decline afterwards, a rising floor, a distribution narrowing on the full-pool score — but the whole family of curves has dropped by roughly 0.18 AUROC.

5.2.3 Why: the perplexity of the classes

The cause is visible directly in the raw perplexities. Table 8 gives, per rater, the geometric-mean perplexity of each of the six classes over the same 50 source sentences.

We can see that the order of the pooled mean perplexities is not the one we hoped to obtain. Sorted by probability, the classes come out: machine translations (geometric mean 6.6), the semantically corrupted machine translations (11.5), human references (13.8), semantically corrupted human references (16.7), morphologically broken machine translations (25.3) and morphologically broken human references (43.7). Authorship is more important for perplexity than semantic damage. The mechanism is exactly the one exploited by machine-generated-text detectors (Gehrmann et al.,

	gemma3b	ayaexp	commandr	eu	qwen	mistral7b	llama	dolphin	darwin	ibm	gptoss	aya
ayaexp	.83											
commandr	.75	.88										
eu	.87	.89	.83									
qwen	.72	.76	.70	.76								
mistral7b	.75	.80	.77	.81	.81							
llama	.83	.86	.82	.87	.77	.83						
dolphin	.85	.86	.80	.88	.78	.83	.88					
darwin	.82	.84	.81	.86	.76	.81	.85	.88				
ibm	.84	.78	.69	.81	.78	.77	.82	.84	.81			
gptoss	.22	.27	.32	.29	.31	.39	.30	.31	.38	.25		
aya	.79	.86	.82	.84	.74	.78	.83	.83	.83	.77	.28	
tower	.86	.85	.81	.90	.74	.79	.85	.88	.86	.83	.31	.82

Table 5: Pairwise Spearman rank correlation of the raters’ per-sentence perplexities, computed within each class and averaged with sentence-count weights ($n = 10,853$ aligned sentences per pair). While every other pair of raters correlates .69–.90, gptoss (bold) correlates at most .39 with any other rater.

Rater	BM	BH	CM	CH	Comb.	Marg.
gptoss	0.826	0.930	0.501	0.535	0.707	+0.027
dolphin	0.819	0.932	0.417	0.612	0.706	+0.016
llama	0.812	0.927	0.412	0.581	0.692	+0.012
tower	0.798	0.897	0.436	0.581	0.688	+0.011
eu	0.812	0.921	0.357	0.596	0.684	+0.010
commandr	0.761	0.920	0.382	0.582	0.673	+0.008
darwin	0.789	0.904	0.366	0.532	0.659	+0.003
mistral7b	0.729	0.907	0.317	0.569	0.643	−0.002
qwen	0.675	0.865	0.303	0.631	0.629	−0.004
aya	0.671	0.854	0.321	0.509	0.599	−0.015
ayaexp	0.620	0.847	0.266	0.543	0.580	−0.020
ibm	0.595	0.742	0.331	0.537	0.559	−0.022
gemma3b	0.644	0.809	0.243	0.483	0.556	−0.025

Table 6: Experiment 2: cross-validated singleton AUROC per rater and class with human references as the positive class, sorted by the combined score. Fold spreads are ~ 0.06 (BM and BH), ~ 0.10 (CH), ~ 0.05 (CM) and ~ 0.06 (combined). Values below 0.5 mean the rater ranks human references as *worse* than damaged sentences more often than not; the CM column is below 0.5 for twelve of the thirteen raters and the thirteenth is at chance.

2019): the corruption of a LLM translation leaves fluent, high-probability, model-typical Czech. The damage the rewrite does is semantic — it is invisible to a fluency signal — while the fluency gain from being model-generated is large and directly visible. The two effects work in opposite directions and they very nearly cancel.

6 Discussion

So, how many raters? In our experiments, the best results were achieved with well chosen ensembles of two or three: dolphin+gptoss already reach 0.887 and a third (llama) the optimum 0.890, past which the best achievable score only declines,

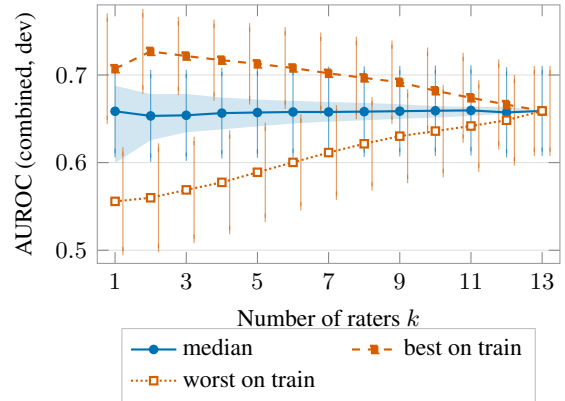


Figure 4: Experiment 2: cross-validated combined dev AUROC versus ensemble size k with human references as the positive class, plotted exactly as Figure 2. Note the y range: the best ensemble of any size (0.727) does not reach the *worst* singleton rater of Experiment 1 (0.764).

with the full pool (0.848) ending up *worse* than the best singleton rater (0.876). If the raters cannot be chosen intentionally — i.e. we don’t know how good they are — then more raters raise the floor and protect against a bad pick, but they are unlikely to beat a smaller ensemble of better raters.

Composition beats count. The size-matched marginal values in Table 3 suggest an explanation why: rater quality for ensembling is not equivalent to the rater’s singleton quality. gptoss is a mid-table singleton on machine translations (0.848, fifth of thirteen) but the most valuable team member, because its perplexity errs differently from the other models’ — its rank correlation with every other rater is at most .39, against .69–.90 for any

k	Best ensemble (train)	Train	Dev	sd
1	gptoss	0.704	0.707	.058
2	+ dolphin	0.725	0.727	.043
3	+ llama	0.720	0.722	.040
4	+ tower	0.714	0.717	.041
5	+ eu	0.708	0.713	.042
6	+ commandr	0.704	0.708	.042
13	all thirteen raters	0.650	0.659	.047

Table 7: Experiment 2: best ensemble per size on the combined task, selected on the training splits, with the standard deviation of its ten per-fold dev means. Every row extends the row above by one rater, and the chain is exactly what greedy forward selection picks — including the optimum at $k=2$.

other pair (Table 5). This is the classical ensemble-diversity argument (Dietterich, 2000) played out over frozen LLMs: what matters is decorrelation of errors, which singleton AUROC does not reveal but the correlation matrix of Table 5 makes directly visible.

Fluency vs. adequacy, revisited. The class-level results refine the standard observation that perplexity measures fluency: for fluency violations (morphology) LLM raters broadly agree, and a single complementary pair captures nearly all the signal (0.967); for adequacy violations (semantics) the signal is for all raters harder to pick up than fluency.

Correctness vs authorship Taken together, the two experiments suggest that a perplexity ensemble is in addition to *fluency* and *adequacy* a detector of *model-typicality* to a degree that can outweigh the *adequacy* signal. Experiment 1’s positives and negatives are both LLM output, differing only in the damage applied afterwards, and there the perplexity judging does well. Experiment 2 breaks that symmetry by using human references and we see that when comparing human text against deliberately wrong machine text, the ensembles prefer the wrong machine text.

7 Conclusion

We exhaustively evaluated all 8k ensembles of 13 open-weight LLMs used as perplexity raters of English to Czech translation, twice, under 10-fold cross-validation over the 50 source sentences. With LLM-based translations as the positive class, a single good rater separates good from degraded ones with held-out AUROC 0.876, a well-chosen pair reaches 0.887, and the optimum of 0.890 needs

Rater	Good		Damaged			
	MT	Human	CM	CH	BM	BH
gemmabig	16.3	55.8	23.9	58.1	100.8	202.4
ayaexp	10.7	34.4	15.8	42.5	53.1	136.0
commandr	5.1	10.3	9.0	12.7	18.4	32.0
eu	4.4	7.4	6.7	9.4	15.5	23.5
qwen	5.4	13.4	11.8	20.2	23.1	47.2
mistral7b	3.2	6.1	5.3	7.2	9.3	16.2
llama	4.5	8.2	9.0	10.3	17.8	29.1
dolphin	4.4	8.2	9.1	10.8	18.4	30.4
darwin	4.1	7.2	5.8	7.9	13.5	21.1
ibm	9.7	21.5	19.0	25.1	28.3	44.4
gptoss	21.6	38.6	39.9	41.1	85.4	120.6
aya	6.3	13.6	9.7	14.3	20.0	35.0
tower	6.4	10.6	12.7	14.4	24.5	38.1
geo. mean	6.6	13.8	11.5	16.7	25.3	43.7

Table 8: Geometric-mean perplexity of each class per rater, over the same 50 source sentences, columns sorted by the pooled value. The two undamaged classes are grouped apart from the four damaged ones. *MT* is the machine-translated *good* class and *Human* the ~ 38 human references per sentence; *CM* is the LLM’s corruption of the former, *CH* its corruption of the latter, *BM* morphological damage to the former and *BH* the same damage to the latter. Nine of the thirteen raters rank a *corrupted machine translation* as more probable than an untouched human reference.

only three raters (llama, dolphin, gptoss). The full 13-rater pool (0.848) scores below the best singleton rater and the median ensemble of any size never beats it. The cross-validation confirms the selection isn’t merely lucky: the ensemble ranking transfers from training to test items at $\rho = 0.998$, and the train-selected optimum is exactly the best ensemble found anywhere on the test set.

With human reference translations as the positive class, the approach massively worsens whenever the comparison is against machine generated sentences: the best of all ensembles reaches AUROC of 0.727, and against the LLM’s corruptions of its own translations the ordering *inverts* — all but one of the ensembles score below 0.5, with a median of 0.339. Furthermore, the examined LLMs thus do not discriminate well between diverse correct human translations and semantically corrupted LLM translations. Semantic modeling in available open-weight models is thus still unreliable.

Limitations

Our study needed a dataset with a high number of reference translations. Such a dataset was available for English→Czech translation but it also brought the main limiting factor for our study: We exper-

iment with only one language pair with a single, morphologically rich target language. Furthermore, the underlying corpus contains only 50 source sentences. This reflects in a lower stability of our estimates, e.g. during the cross-validation, the same ensemble scores 0.853 and 0.921 on two different folds in Experiment 1, 0.649 and 0.788 in Experiment 2. The fold spreads of Table 2 are ± 0.02 and ± 0.04 on the two optima — any differences smaller than those shouldn't be seen as representative. In particular, the optimum at three raters is more accurately read as an optimum around one to six raters.

The degradations are synthetic and controlled; real MT errors mix fluency and adequacy problems in ways our classes separate artificially. While the study uses 13 different models, they are (apart from one) strongly correlated with each other, a more diverse set of raters may yield different outcomes. This rater sameness issue carries over to the LLM good and LLM corrupt classes which may therefore provide less diversity than their size suggests.

Finally, the set of human references was publicly available since 2011 and could have leaked to the LLM training data. The high perplexity these translations obtain however does not support this worry.

Acknowledgement

The work has been partially supported by the grants CZ.02.01.01/00/23_020/0008518 (“Jazyková, umělá inteligence a jazykové a řečové technologie: od výzkumu k aplikacím”) and 10131474 (Czech AI Factory, CZAI) by EuroHPC JU.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

- Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. 2013. [Many czech references for 50 sentences selected from WMT11 data](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).
- Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15. Springer.
- Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. volume 8, pages 539–555.
- Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.
- Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.
- James A. Hanley and Barbara J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.
- Marcin Junczys-Dowmunt. 2018. [Dual conditional cross-entropy filtering of noisy parallel corpora](#). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.
- Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203. European Association for Machine Translation.
- Henry B. Mann and Donald R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60.
- Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997. Association for Computational Linguistics.
- Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 220–224. Association for Computational Linguistics.
- Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. 2024. Open-LLM-leaderboard: From multi-choice to open-style questions for LLMs evaluation, benchmark, and arena. *arXiv preprint arXiv:2406.07545*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th Annual Meeting of the Association for Computational Linguistics, pages 311–318. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5070–5081. International Committee on Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2685–2702. Association for Computational Linguistics.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajič. 2014. MorphoDiTa: Morphological dictionary and tagger. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 13–16. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012. European Language Resources Association.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Few-shot Translation Prompt

We used this prompt to obtain LLM-based translations:

Create a Czech translations for the given English sentence, format the answer exactly as in the examples, don't add anything else

en: "The morning was cold but sunny."

cs: "Ráno bylo chladné, ale slunečné."

en: "We waited for the train for full two hours."

cs: "Než vlak přijel, strávili jsme dvě hodiny čekáním."

en: "I have never seen such a beautiful sunset."

cs: "Tak krásný západ slunce jsem ještě neviděl"

B Semantic Corruption Prompt

Introduce semantic mistakes to the given Czech sentence, format the answer exactly as in the examples, add nothing more.

correct: "Ráno bylo chladné, ale slunečné."

wrong: "Ráno nebylo chladné, ani nepršelo."

correct: "Než vlak přijel, strávili jsme dvě hodiny čekáním."

wrong: "Než autobus přijel, strávili jsme pět minut čekáním."

correct: "Tak krásný západ slunce jsem ještě neviděl."

wrong: "Tak krásný východ slunce jsem už viděl."