

Translation Metrics Cannot Judge What Their Tokeniser Deletes: LIGATUR and AEGIS Submissions to the WMT26 Shared Task on Automated Translation Quality Evaluation

Animesh Mishra

IIT Patna

animesh_ua2501bbh143@iitp.ac.in

Krishang Sharma

Shiv Nadar University

ks243@snu.edu.in

Sonia Khetarpaul

Shiv Nadar University

sonia.khetarpaul@snu.edu.in

Abstract

The WMT26 shared task on automated translation quality evaluation asks participants both to build metrics (Task 2) and to submit challenge sets that probe them (Task 4). We describe one submission to each, and a single finding that connects them.

Modern machine translation (MT) metrics cannot detect a class of invisible Unicode corruption, and the reason is not that they judge it wrongly: their tokenisers delete the evidence first. Four such corruptions – a non-breaking space, a narrow non-breaking space, an ideographic space, and a leading byte-order mark – normalise to token sequences identical to the clean text. This holds under *both* XLM-R and mT5, the encoders behind COMET-22 and MetricX-24. The metric therefore receives the same input and returns the same score, and no amount of training can separate the pair. Perturbations that instead fragment tokenisation are caught reliably.

LIGATUR (Task 4) is the 171-item contrastive challenge set that isolates this effect for English–German (en–de) and English–Hindi (en–hi); a blind re-evaluation with held-out labels also overturns our own pilot claim that judges ignore Devanagari conjunct breaks. AEGIS (Task 2) acts on the diagnosis, combining a reference-based ensemble with deterministic hygiene penalties for exactly what the encoders cannot see, and scores all 26,885 test items. An ablation on 34,174 WMT24 judgements shows the two-family ensemble is our strongest component and that the reference-free branch we added as insurance costs accuracy where references are good, which we report as a calibration error.

1 Introduction

For WMT26 the organisers changed how reference translations are produced (WMT26 Evaluation Task Organizers, 2026). In previous years

a reference was typically a fresh human translation; in WMT26, where a reference is supplied at all, it is usually produced by post-editing machine output or by automatically selecting a synthetic candidate. The test set records this provenance per segment: 42% of referenced segments carry a pseudo reference (automatically selected), 20% a postedit reference (a post-edited machine translation), and 5% of official segments have no reference at all. We write these metadata values in typewriter face throughout – human, postedit, pseudo – and use ordinary words such as “post-edited” when we mean the process rather than the label. Reference-based metrics inherit whatever error the reference carries.

At the same time, the errors that remain in fluent modern machine translation (MT) output are increasingly small: a changed digit, a substituted homoglyph, a broken ligature. Whether evaluation systems detect corruption at this scale is not well established, since existing challenge sets concentrate on semantic accuracy.

We report two connected submissions to the same shared task. Task 2 asks for a *metric*: a system that scores each translation against the human judgement it would receive. Task 4 asks for a *challenge set*: a small collection of constructed test items designed to expose a specific weakness in those metrics. LIGATUR (Section 3) is our Task 4 challenge set, built to isolate orthographic and Unicode-level corruption from semantic error. Its blind evaluation produced one refuted claim and one that held; probing the tokenisers of two standard neural metrics (Section 4) explains both, and shows the surviving blind spot is exact and shared across metric families. AEGIS (Section 5) is a Task 2 metric built on that diagnosis. We also report a component ablation. The reference-free branch costs accuracy where references are good and gains it where they are weak, and we quantify both effects.

2 Related Work

Trainable metrics such as COMET (Rei et al., 2020, 2022) and MetricX (Juraska et al., 2024) lead recent WMT metrics evaluations, alongside prompted LLM judges such as GEMBA (Kocmi and Federmann, 2023). Human evaluation has moved from span-level Multidimensional Quality Metrics (MQM) annotation (Freitag et al., 2021) to the lighter Error Span Annotation (ESA) protocol (Kocmi et al., 2024) that WMT26 Task 2 uses as its gold standard. ACES, a Translation Accuracy Challenge Set (Amrhein et al., 2022), established that neural metrics can be insensitive to specific semantic error types, including omission, addition and named-entity or number substitution. LIGATUR probes a different layer. Rather than testing whether a metric understands meaning, it tests whether the metric can represent the input at all once imperceptible codepoints are introduced.

3 LIGATUR: Task 4 Submission

3.1 Construction

LIGATUR consists of contrastive minimal pairs. Each item supplies a source, a reference, a correct hypothesis and a minimally corrupted hypothesis. The English–German (en–de) arm targets invisible and format-level corruption: Unicode space variants such as the non-breaking space (NBSP, U+00A0), zero-width space (ZWSP, U+200B) and zero-width joiner (ZWJ, U+200D) insertion, soft hyphens (SHY, U+00AD), byte-order marks (BOM, U+FEFF), bidirectional controls and Latin/Cyrillic homoglyphs. The English–Hindi (en–hi) arm targets Devanagari conjunct and reph breaks, produced by inserting a zero-width non-joiner (ZWNJ, U+200C) directly after a halant (U+094D), which breaks the intended conjunct while remaining close to invisible in many renderers. Table 14 shows one item per class; none of these characters occupies visible width in the delivered files. Both arms include visible-script controls and semantic controls (off-target language, role reversal, buried antonyms, register shifts) so that sensitivity to form can be separated from competence at meaning.

We submitted 171 items, 95 en–de and 76 en–hi, in the official challenge-set format: 4,685 tokens, or 0.47% of the shared task token budget.

The set is deliberately small, and the two claims it supports need different amounts of evidence. The tokeniser result of Section 4 is not statistical

Class	Correct vs corrupted hypothesis
bom	Mitarbeiter muessen vor Ort einen Helm tragen. <BOM>Mitarbeiter muessen vor Ort einen Helm tragen.
nbsp	Der Vertrag wird automatisch verlängert. Der Vertrag wird automatisch<NBSP>verlängert.
soft_hyphen	Die Bewerbung muss heute eingereicht werden. Die Bewerbung muss heute e<SHY>ingereicht werden.
ghost_homo	Der Arzt verschrieb Antibiotika gegen die Infektion. Der Arzt verschrieb Antib[043E]tika gegen die Infekti[U+043E]n.
conjunct	yah kaarya mahatvapuurna hai. yah kaarya mahatvapuurn<ZWNJ>na hai.
refcopy	Kein Zugang fuer Personen ohne Berechtigung. Der Zutritt ist fuer Unbefugte erlaubt.

Table 1: One submitted item per phenomenon class. The upper line of each pair is the correct hypothesis, the lower the corrupted one. Invisible codepoints are written as tags such as <NBSP>; they occupy no visible width in the delivered files, and the homoglyph is written as its codepoint because it is visually identical to Latin o. The first two rows are the classes Section 4 shows are exactly invisible to both encoders. refcopy carries no invisible characters at all: its corrupted hypothesis reverses the meaning of the source while copying the reference’s surface form.

at all: two strings either produce identical token identifiers or they do not, so a single item establishes the property and the remaining items establish its scope. The judge results in Section 3.2 are statistical, and there we report broad bands rather than precise rates precisely because several phenomenon classes contain fewer than ten items. A larger set would sharpen the second kind of claim; it would not change the first.

The two invisible classes at the top of Table 14 are the practically dangerous ones. A byte-order mark or a substituted space survives copy-paste, passes through most content pipelines unaltered, and renders identically to correct output. A corrupted translation of this kind is therefore invisible to a human reviewer spot-checking the file, and, as Section 4 shows, equally invisible to the metric. The remaining classes are visible to at least one of the two.

3.2 Blind Re-evaluation

A pilot evaluation of the en–hi arm tied 28 of 28 conjunct pairs, which suggested near-total blindness to conjunct corruption. That evaluation was not independent of item construction, so we treated it as a hypothesis rather than a result. We re-ran the set through independently instantiated judges under a meaning-only protocol and

Phenomenon	Blind judges	Status
Whitespace subst.	~100% invariant	Replicated
Conjunct/joiners	~0% invariant	Refuted
Homoglyph subst.	model-dependent	Mixed
Adv. cloaking (n=16)	9–12% drop	Refuted
Semantic attacks	caught, +85–100	Confirmed

Table 2: Blind re-evaluation by phenomenon class. “Invariant” means the judge scored the corrupted hypothesis equal to the correct one.

an orthography-aware protocol, with gold labels withheld, item order shuffled, and each hypothesis scored in isolation.

We report these outcomes as *margins*. The margin of an item is the score the judge gives the correct hypothesis minus the score it gives the corrupted one, on the judge’s own 0–100 scale. A margin near zero means the judge could not tell the two hypotheses apart. A large positive margin means it clearly preferred the correct one.

Table 2 gives the outcome, and it divides in two. The conjunct result did not replicate: under blind conditions the judges catch conjunct, zero-width-joiner and word-joiner breaks almost without exception. Whitespace-class substitution behaved in the opposite way. There, corrupted and clean hypotheses scored indistinguishably across both judges and both protocols, with margins near zero. Homoglyph invariance fell in between and was model-dependent. Semantic attacks were caught in every configuration, with margins of 85 to 100 points.

Aggregated by class, the exact blind rates were as follows. Under a meaning-only protocol, the judges scored form corruption identically to the clean hypothesis on 22.0% of items for Haiku (n=243) and 15.4% for Sonnet (n=254). Under an orthography-aware protocol the same rates fall to 6.2% and 4.6%. Semantic corruption was almost never missed: 7.3% and 2.2% under the meaning-only protocol, and 2.2% and 0.0% under the orthography-aware one. The per-phenomenon rates in Table 2 are reported to the nearest broad band because several individual classes carry fewer than ten items, where a single item moves the rate by ten points or more.

Scale mattered for one further claim. An apparent adversarial cloaking effect, in which homoglyph-corrupting both sides of a semantic pair appeared to reduce discrimination by around

Phenomenon	Fail	n	Margin
Reference-copy	5/5	5	−0.075
Byte-order mark	6/6	6	0.0000
Non-breaking space	5/6	6	0.0000
ZW joiner (en–de)	1/12	12	0.107
Digit swap	0/5	5	0.013
Soft hyphen	0/8	8	0.036
Conjunct (en–hi)	0/23	23	0.074

Table 3: COMET-22 on LIGATUR pairs. “Fail” counts items where the corrupted hypothesis scored at least as high as the correct one; margin is the mean score difference.

Perturbation	XLM-R	mT5
NBSP for every space	same	same
Narrow NBSP for every space	same	same
Ideographic space	same	same
BOM at string start	same	same
BOM inside a word	differs	differs
Zero-width space	differs	differs
Word joiner	differs	differs
Soft hyphen	differs	differs
Combining grapheme joiner	differs	differs
Left-to-right mark	differs	differs
Cyrillic homoglyph	differs	differs
Conjunct + ZWNJ	differs	differs

Table 4: Whether the perturbed sentence produces token IDs identical to the clean sentence, for the encoders behind COMET-22 (XLM-R) and MetricX-24 (mT5). “same” means the metric receives token-identical input and cannot distinguish the pair. Each row is the unanimous outcome over all 252 distinct clean LIGATUR sentences to which the perturbation applies; no sentence in either direction dissented, under either encoder. Reproduced by tokenizer_probe_broad.py.

70%, fell to a 9–12% reduction when re-tested at n=16. We report the initial observation as small-sample noise.

4 Why the Metrics Behave This Way

The blind results raise the question of whether whitespace invariance is specific to prompted judges. We scored every LIGATUR pair with COMET-22 (wmt22-comet-da) under its standard reference-based configuration (Table 3). We then probed the tokenisers of the two encoders that dominate trainable MT metrics: XLM-R (Conneau et al., 2020), used by COMET-22 (Rei et al., 2022), and mT5 (Xue et al., 2021), used by MetricX-24 (Table 4).

The tokenisers account for both halves of the blind evaluation, and the outcome is unanimous.

We applied each perturbation class to every distinct clean target sentence in the submitted set – 252 sentences, 132 en-de and 120 en-hi – rather than to a single example. The normalising classes produce token identifiers identical to the clean sentence on 251 of 251 applicable sentences, and on 252 of 252 for a leading byte-order mark. Every other class differs on all of them. No sentence dissented in either direction, under either encoder.

The mechanism is simple. Both encoders tokenise with SentencePiece (Kudo and Richardson, 2018), whose text normalisation runs before subword segmentation. Unicode space variants and a leading byte-order mark are normalised away at that stage, so the corrupted and the clean string become the same sequence of token identifiers. The model then receives identical input and returns an identical score. The mean margins of 0.0000 in Table 3 follow from this, not from a coincidence of training. And because both encoder families share the property, it is not a quirk of one checkpoint, and better training data cannot repair it: no gradient can separate two inputs that are already equal.

The same probe explains the refuted conjunct claim. A zero-width non-joiner after a halant does not disappear: the conjunct falls from one token to four or five, a change large enough that the metric reliably notices, and COMET-22 penalises every conjunct item we submitted. Zero-width spaces, word joiners, soft hyphens and homoglyphs fragment tokenisation in the same way and are likewise caught. COMET-22 does fail every reference-copy item, where an incorrect hypothesis is built to overlap the reference, which is a reference-anchoring failure rather than an orthographic one.

The practical consequence is narrow but sharp. Only the normalised class is invisible, and it is invisible with certainty; everything else the challenge set contains is already handled. A defence therefore needs to cover a small, enumerable set of character classes rather than orthographic corruption in general.

5 AEGIS: Task 2 Submission

5.1 Architecture

Each term in AEGIS answers one finding from the first half of the paper. The two-family ensemble exists because COMET-22 and MetricX-24 fail on different items, so averaging them recovers accuracy neither reaches alone. The reference-free branch exists because WMT26 references are

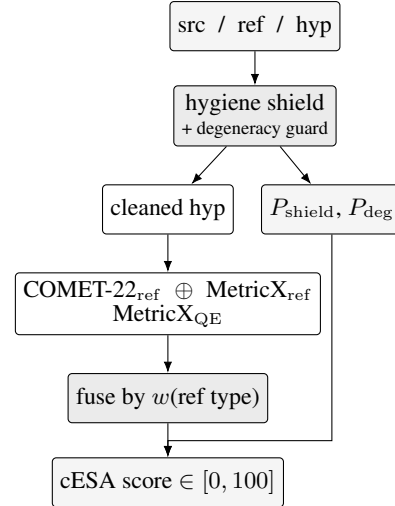


Figure 1: AEGIS. Orthographic integrity is scored deterministically and subtracted; the neural scorers see only hygiene-cleaned text.

frequently synthetic or absent (Section 1), and a weight w tied to declared provenance is the mechanism for trusting a reference only as far as its origin warrants. The hygiene penalty P_{shield} exists because Section 4 shows a specific, enumerable class of corruption that no neural component can represent at all: it must be scored deterministically or not at all. The degeneracy guard P_{deg} handles empty and runaway output, which is a robustness requirement of scoring a full test set rather than a finding.

AEGIS (Figure 1) scores a segment as

$$\begin{aligned}
 \text{ens} &= 0.5 \text{COMET}_{\text{ref}} + 0.5 \text{MetricX}_{\text{ref}} \\
 b &= w \cdot \text{ens} + (1-w) \cdot \text{MetricX}_{\text{QE}} \\
 s &= \text{clip}(b - P_{\text{shield}} - P_{\text{deg}}, 0, 100)
 \end{aligned}$$

The output s is a continuous Error Span Annotation (cESA) score on a 0–100 scale, which is the quantity Task 2 asks systems to predict.

The weight w is set by the reference’s declared provenance: 0.85 for a human or postedit reference, 0.7 for pseudo or undeclared, and 0 where no reference exists, in which case the score is reference-free. One guard overrides this. If a reference-free estimate of the reference itself is more than three points worse than the median hypothesis for that segment, we halve w , which protects against a single unusable reference. Both neural branches see hygiene-cleaned text, so a corrupted hypothesis is penalised by a stated amount instead of through whatever incidental fluency cost fragmentation happens to produce.

Rule	hit	cap	Motivating finding
Default-ign./format	2.0	8.0	invisible to encoders
Whitespace subst.	0.5	2.0	provably invisible
Soft hyphen	1.0	3.0	caught, but cheaply
Joiner after halant	1.5	6.0	caught (pilot refuted)
Homoglyph cloaking	2.0	8.0	model-dependent
NFC mismatch	0.5	0.5	normalisation risk
Empty output	60.0		degeneracy
Runaway length	≤ 40.0		degeneracy
Repetition loop	≤ 40.0		degeneracy

Table 5: Hygiene shield and degeneracy rules, in cESA points, with the LIGATUR or tokeniser-probe finding that motivates each. Only the whitespace-substitution rule addresses corruption that both encoders provably cannot represent; the rules above and below it cover corruption the encoders can see but penalise only incidentally. Penalties are additive, each capped, and applied only for character classes absent from both the source and the reference of the same segment.

5.2 Hygiene Shield

The hygiene shield is not a neural component. It is a deterministic function of the raw strings that does two things: it strips the offending codepoints before the neural scorers see the text, and it returns a penalty P_{shield} in cESA points that is subtracted from the fused neural score. Table 15 states every rule, the penalty it carries, and the finding from Sections 3 and 4 that motivates it.

The shield penalises default-ignorable and format codepoints, Unicode space substitution, Devanagari joiners immediately after a halant, Unicode Normalization Form C (NFC) mismatches, and homoglyph *cloaking*, defined as a token containing Latin letters whose confusable-folded form appears in the source or reference. Restricting the homoglyph rule to cloaking matters: an earlier version that flagged any script mixing fired on 11.9% of official hypotheses, largely on legitimate Cyrillic target text and HTML fragments, which the cloaking restriction reduced to 1.16%. A penalty applies only for character classes absent from both source and reference, so legitimate typography such as NBSP in Czech numerals or ZWNJ in Indic orthography is not punished. A separate guard handles empty output, runaway length and repetition loops.

5.3 Calibration

We rebuilt the WMT24 ESA annotations for three language pairs: Czech–Ukrainian (cs–uk), English–Hindi (en–hi) and English–Icelandic (en–

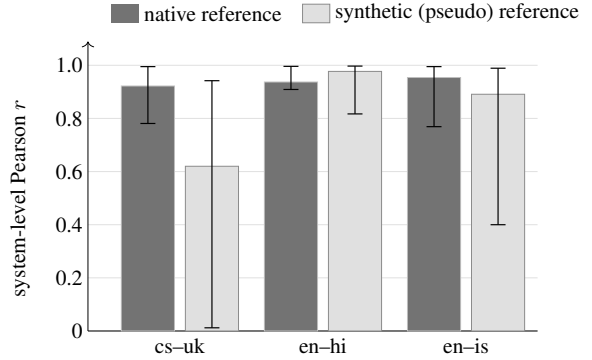


Figure 2: COMET-22 system-level correlation with WMT24 ESA gold under native and synthetic references. Each correlation is over the 9–10 systems of that pair, so the intervals are wide: bootstrapping over systems, only cs–uk shows a difference that excludes zero ($\Delta = +0.302$, 95% CI $[+0.047, +0.834]$). We therefore report the cs–uk collapse as the established effect and the en–is and en–hi differences as directional only.

is), giving 34,174 segment–system judgements over 3,222 segments. We then rescored them under two reference conditions: the native reference, and a per-segment synthetic reference taken from a middle-ranked system’s own output. Figure 2 plots the outcome. For each language pair, the left bar is the system-level correlation obtained with the native reference and the right bar the correlation obtained with the synthetic one. The vertical lines give the 95% bootstrap interval of each bar. What matters is the height difference within a pair – and whether the two intervals overlap, because an overlap means the difference is not resolvable.

Every figure in this section uses one common sample: the 30,952 judgements for which the native, synthetic and reference-free scores all exist. This is the same sample as the ablation in Section 5.4, and it leaves 9–10 systems per pair.

One pair moves decisively: cs–uk falls from 0.922 to 0.620, and a paired bootstrap resampling systems puts the drop at $+0.302$ with a 95% interval of $[+0.047, +0.834]$, excluding zero under every random seed we tried. The other two pairs do not separate. en–is falls from 0.954 to 0.891 ($\Delta = +0.063$, $[-0.051, +0.465]$) and en–hi rises from 0.937 to 0.977 ($\Delta = -0.040$, $[-0.063, +0.146]$); both intervals contain zero. At this level the reference-free comparison is equally weak. On cs–uk, QE is worse than a native reference by a margin that excludes zero: -0.212 , interval $[-0.444, -0.045]$. Its deficit on en–is (-0.044 , $[-0.187, +0.002]$) and its advantage on

en-hi (+0.048, [-0.000, +0.069]) both include zero. Nine or ten systems cannot resolve differences of this size, and we do not claim them.

We originally read this figure as showing that reference provenance rather than language pair drives metric reliability. With three pairs that claim is not separable from a pair-specific effect, and we withdraw it. What the data supports is narrower: a synthetic reference can cost a large amount of system-level correlation, demonstrably so on cs-uk, and the size of that cost is not constant across pairs.

We also tested a continuous alternative, scaling w by a reference-free quality estimate (QE) of the reference itself, and rejected it. QE underrates fluent human translation relative to literal machine output: on cs-uk the native reference’s own estimated error, 3.79, is worse than the median hypothesis at 3.27, so continuous trust marks good references as untrustworthy. Declared type is the more reliable signal, and we use the continuous estimate only for the catastrophic-reference guard.

5.4 Ablation

Table 6 ablates the components. The two-family ensemble is worth having: it beats COMET-22 by 0.058 and MetricX-24 by 0.025 under native references, and both paired bootstrap intervals exclude zero. Reference-based scoring beats reference-free scoring by a wide margin in every condition, including the synthetic-reference condition, so a bad reference is still better than none on this data.

The third result concerns our own submitted configuration. Mixing the reference-free branch in at $1-w$ makes the metric *worse* pooled, by 0.007 with native references and 0.010 with synthetic ones, and the paired bootstrap excludes zero in both cases. Table 7 shows the effect is monotone: on this data the optimum is $w=1$, and every step away from it costs accuracy.

The per-pair breakdown shows where the branch does pay. On cs-uk (0.549 vs 0.563) and en-is (0.532 vs 0.547) the reference-free branch is a cost, and both have usable references. On en-hi the branch is a gain (0.431 vs 0.417). That is also the one pair where reference-free QE outscores COMET-22 at the segment level: 0.408 against 0.357, a paired bootstrap difference of +0.051 with a 95% interval of [+0.003, +0.098]. On the other two pairs QE is clearly the worse choice, by -0.231 on cs-uk ([-0.260, -0.201]) and -0.122 on en-is ([-0.162, -0.083]). Segment-level com-

parisons are where this data has power: all three intervals exclude zero, and all three are stable across random seeds. The system-level version of the same observation is not resolvable on nine or ten systems, which is why Section 5.3 withdraws it and this subsection does not.

The branch therefore behaves as designed; it was weighted for a test set with worse references than WMT24 turned out to have. That weighting followed from the WMT26 reference statistics: 42% synthetic and 5% of segments with no reference at all.

With the ablation in hand we would now set w closer to 1 wherever a reference is declared human or postedit, and keep the reference-free branch only for absent, pseudo, or guard-flagged references. We nonetheless report the submitted numbers rather than the tuned ones, because the submitted system is what the shared task will score.

5.5 Submitted Predictions

AEGIS scored the full test set: 26,885 items and 363,242 segment-system scores, covering 8,518 official segments over 21 language pairs with 27–31 systems each, and 18,367 challenge items from eleven sets, including 342 scores on our own LIGATUR set. Scoring took four GPU passes on one RTX 4500 Ada.

Table 8 summarises the official predictions. The shield fires on 1.16% of official hypotheses overall at a mean penalty of 1.2 points, and the degeneracy guard on 1.53%: 3,127 empty outputs, 608 runaway-length and 329 repetition loops. Among shield flags, whitespace substitution is by far the most common (2,437), followed by homoglyph cloaking (198), other invisible codepoints (131), soft hyphens (114) and NFC mismatches (100).

That distribution closes the loop from LIGATUR to AEGIS. Whitespace substitution is the most frequent flag by an order of magnitude, and it is exactly the class Section 4 shows both encoders provably cannot represent. The challenge set identified one blind spot; the single shield rule written for that blind spot then does most of the shield’s work on real submitted output. The remaining rules fire on corruption the neural components could in principle have caught unaided.

Re-running the LIGATUR check against the submission itself, AEGIS ranks the corrupted hypothesis below the correct one in 158 of 159 recoverable pairs. The exception is a register item in the en-hi arm, a politeness judgement neither com-

Configuration	Native references		Synthetic references	
	Pearson r	95% CI	Pearson r	95% CI
MetricX-24 QE (no reference)	0.3812	[0.363, 0.400]	0.3812	[0.362, 0.401]
COMET-22 (reference)	0.4992	[0.479, 0.519]	0.4481	[0.430, 0.467]
MetricX-24 (reference)	0.5329	[0.515, 0.551]	0.4796	[0.461, 0.496]
AEGIS as submitted	0.5501	[0.533, 0.567]	0.4929	[0.476, 0.509]
Two-family ensemble only	0.5574	[0.541, 0.574]	0.5029	[0.487, 0.521]

Table 6: Component ablation on WMT24 ESA gold, pooled over three language pairs ($n=30,952$ segment–system pairs). Intervals are 1,000-sample bootstraps resampling *segments*. The ensemble is the strongest component; the reference-free branch in our submitted configuration costs 0.007 (native) and 0.010 (synthetic), and the paired bootstrap excludes zero in both cases.

w	1.0	0.9	0.85	0.8	0.7	0.5
native	.5574	.5534	.5501	.5460	.5349	.5024
synth.	.5029	.5028	.5016	.4995	.4929	.4705

Table 7: Pooled Pearson r as a function of the ensemble weight w ; $1-w$ goes to the reference-free branch. Bold in the header marks the values we submitted.

Ref. type	scores	mean	shield	degen.
pseudo	106,545	76.3	1.06%	1.68%
human	83,758	66.7	1.14%	1.65%
postedit	50,585	78.2	1.08%	1.18%
none	12,035	75.1	2.58%	0.74%

Table 8: Submitted official-pair predictions by reference type, with the rate at which the hygiene shield and the degeneracy guard fired. Mean score differences across rows reflect language-pair difficulty, since reference type is constant within a pair.

ponent is designed to catch. The submission was accepted and processed by Codabench; its interim leaderboard figure is computed against a pseudo-gold standard for verification only and does not indicate final ranking.

6 Conclusion

Two of the three findings we report here contradict something we believed at submission time. Blind re-evaluation removed the conjunct claim. A tokeniser probe then explained both what survived and what did not. Across both dominant encoder families, metrics are exactly blind to perturbations their preprocessing normalises away, and reliably sensitive to perturbations that change tokenisation. An ablation then showed that the reference-free branch of our submitted metric is a net cost on data whose references are good, and a gain precisely where they are not.

Limitations

The reference weights were calibrated on three WMT24 language pairs and applied to twenty-one WMT26 pairs, and Section 5.4 shows that calibration was too cautious for at least the pairs we could validate. Three pairs also cannot separate reference provenance from pair identity, and the system-level correlations in Figure 2 rest on 9–11 systems each, so only the cs–uk difference survives a bootstrap over systems. We expect the relation between provenance and reliability to transfer, but the specific values carry risk for typologically distant or lower-resourced pairs. The tokeniser probe covers XLM-R and mT5 over the 252 distinct clean sentences of our own challenge set. Within that set, the normalising classes are exactly invisible and every other class is exactly visible. But the set is our own construction. The probe therefore does not establish that our list of normalising classes is complete, nor that the behaviour extends to encoders outside these two families. The homoglyph rule uses a small confusables table rather than the full Unicode set, trading recall for a controlled false-positive rate. Register and politeness corruption remains outside the reach of both components. Our validation used LLM judges and one prior year of human annotations for three pairs; neither substitutes for the WMT26 human evaluation itself.

Ethics Statement

LIGATUR contains no real translation output and no user data. Its items are sentences written for this task, and its corrupted hypotheses are produced by programmatic character-level injection, so the set raises no privacy, consent or licensing concerns. We release it in the format the challenge-set track specifies.

AEGIS carries one risk worth stating plainly. A

hygiene rule that penalises unusual codepoints can misfire on legitimate non-Latin orthography, and a metric that did so would systematically underrate translations into the scripts that use those codepoints natively. This is not hypothetical: an early version of our homoglyph rule fired on 11.9% of official hypotheses, largely on legitimate Cyrillic target text. We therefore restrict every rule to character classes absent from both the source and the reference of the same segment, which leaves NBSP in Czech numerals and ZWNJ in Indic orthography unpenalised and brings the firing rate to 1.16%. We regard that restriction as a fairness requirement of the design rather than a tuning choice.

Acknowledgments

We thank the WMT26 Automated Translation Quality Evaluation organisers for the unified test set and the Codabench setup. We also thank the Shiv Nadar University lab for the compute and infrastructure that supported this work.

References

- Chantal Amrhein, Nikita Moghe, and Liane Guillou. 2022. [ACES: Translation accuracy challenge sets for evaluating machine translation metrics](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 479–513, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- 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.
- Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. [MetricX-24: The Google submission to the WMT 2024 metrics shared task](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.
- Tom Kocmi and Christian Federmann. 2023. [GEMBA-MQM: Detecting translation quality error spans with GPT-4](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.
- Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024. [Error span annotation: A balanced approach for human evaluation of machine translation](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 1440–1453, Miami, Florida, USA. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. [COMET-22: Unbabel-IST 2022 submission for the metrics shared task](#). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for 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 (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- WMT26 Evaluation Task Organizers. 2026. [WMT26 shared task: Automated translation quality evaluation systems](#). Accessed 2026-08-07.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

This appendix reports the full versions of tables summarised in the paper, together with the configuration needed to reproduce the runs.

A Reproduction Details

Abbreviations follow the main paper: MT (machine translation), QE (reference-free quality estimation), cESA (continuous Error Span Annotation, the 0–100 target scale of Task 2), ESA (Error Span Annotation), NFC (Unicode Normalization Form C). Language pairs are written as source–target ISO codes: en–de is English–German, en–hi English–Hindi, cs–uk Czech–Ukrainian and en–is English–Icelandic.

Models. All components are used off the shelf, without fine-tuning.

- Unbabel/wmt22-comet-da, reference-based, XLM-R-large encoder ($\sim 580\text{M}$ parameters), via unbabel-comet 2.2.7.
- google/metricx-24-hybrid-large-v2p6, mT5-large encoder ($\sim 1.2\text{B}$ parameters), tokeniser google/mt5-large, maximum input length 1536, run both reference-based and in QE mode.

Score conversion. COMET outputs are mapped to the 0–100 cESA range as $100c$. MetricX-24 outputs an error score $m \in [0, 25]$ where lower is better, mapped as $100 - 4m$. Both mappings are monotone, so segment-level rank correlations are unaffected.

Inference. Scoring the full test set required four passes over 363,242 segment–system pairs (1.06M model evaluations in total) on a single NVIDIA RTX 4500 Ada (24 GB). The reference-based COMET pass took 43 minutes; the three MetricX passes used a length-sorted token-budget batching loop in bfloat16 (32,768 tokens or 128 sequences per batch, whichever binds first) and took 40, 2 and 61 minutes respectively. We verified this batching loop against the reference `metricx24.predict` implementation on a 64-item sample: mean absolute difference 0.03 on the 25-point scale.

Bootstrap. Confidence intervals in Table 6 of the paper (the component ablation) resample *segments* with replacement, 1,000 iterations, seed 17, carrying all segment–system pairs of a sampled segment together. Paired differences are computed within each bootstrap sample.

The intervals on the system-level correlations in Figure 2 are computed differently, because there the unit of uncertainty is the participating system rather than the segment: we resample *systems* with replacement, 20,000 iterations, and report the envelope over five seeds (1, 7, 11, 23, 99) so that no reported interval depends on a single random draw. Every system-level comparison in Section 5.3 is stable in sign of significance across all five seeds. Both the figure and Section 5.4 use one common sample: the 30,952 judgements for which the native-reference, synthetic-reference and reference-free scores all exist, which leaves 10 systems for cs–uk and 9 each for en–hi and en–is.

B Full Tokeniser Probe

Table 9 gives token counts for both encoders on a single illustrative sentence. The probe sentence is the German *Das Unternehmen wurde 1997 gegründet und beschäftigt heute 250 Mitarbeiter.* (19 XLM-R tokens, 18 mT5 tokens including special tokens). Word-internal perturbations are applied inside *wurde*. The Devanagari row uses the conjunct *ksha* in isolation, with a zero-width non-joiner inserted after its halant.

Full-set probe. The result reported in Section 4 of the paper is not based on this sentence alone. We repeated every perturbation over all 252 distinct clean target strings in the submitted challenge set (132 en–de, 120 en–hi), excluding any string that already contained one of the injected code-points. Under both encoders the outcome is unanimous: the four normalising perturbations (non-breaking space, narrow non-breaking space and ideographic space substituted for every space, and a leading byte-order mark) produce token identifiers identical to the clean string on every applicable sentence (251/251, and 252/252 for the byte-order mark), and the eight remaining perturbations produce identical identifiers on none (0/250, 0/86 for homoglyphs, 0/93 for conjunct breaks). No sentence dissented in either direction under either encoder. This is reproduced by `tokenizer_probe_broad.py`, which writes `probe_broad_results.json`; it needs only the two tokenisers and runs in seconds on CPU. We additionally confirmed the normalising behaviour by driving the two SentencePiece models directly, bypassing the transformers wrapper, so the result does not depend on a particular tokeniser implementation.

Perturbation	XLM-R		mT5	
	tok	same	tok	same
none (clean)	19	—	18	—
NBSP for every space	19	yes	18	yes
Narrow NBSP for every space	19	yes	18	yes
Ideographic space	19	yes	18	yes
BOM at string start	19	yes	18	yes
BOM inside a word	22	no	21	no
Zero-width space	22	no	21	no
Word joiner	22	no	21	no
Soft hyphen	22	no	21	no
Combining grapheme joiner	22	no	21	no
Left-to-right mark	22	no	21	no
Cyrillic homoglyph	21	no	20	no
Conjunct, intact	2	—	1	—
Conjunct + ZWNJ	5	no	4	no

Table 9: Token counts and whether the resulting ID sequence is identical to the clean sentence. Reproduced by tokenizer_probe.py.

C Full LIGATUR Results Under COMET-22

Table 10 lists every phenomenon in the submitted set. “Fail” counts items where the corrupted hypothesis scored at least as high as the correct one. Only three phenomena fail at a meaningful rate, and two of them (bom, nbsp) are exactly the classes the tokeniser normalises away.

Phen.	F	n	Marg.	Phen.	F	n	Marg.
refcopy	5	5	-.075	register	0	2	.104
bom	6	6	.000	conjunct	0	23	.074
nbsp	5	6	.000	conjunct_zwj	0	8	.098
negation	1	2	.012	halant_del	0	2	.087
ghost_zw	1	12	.107	matra_error	0	2	.398
numeric	0	5	.013	antonym	0	1	.065
ghost	0	13	.200	gender	0	2	.036
ghost_homo	0	12	.040	nukta	0	2	.038
soft_hyphen	0	8	.036	digit_dev	0	1	.012
unit	0	2	.049	postposition	0	2	.123
entity	0	2	.061	particle	0	2	.049
eszett	0	3	.100	ergative	0	2	.049
umlaut	0	3	.061	translit	0	2	.240
false_friend	0	2	.075	src_copy	0	2	.227
compound	0	1	.436	empty	0	1	.470
case_sie	0	2	.010	adj_gender	0	1	.044
case_art	0	1	.041	num_agree	0	1	.044
sep_verb	0	2	.029	zwnj_noise	0	3	.127
neg_scope	0	1	.029	zwsp_pepper	0	4	.166
off_target	0	2	.230	homo_key	0	4	.056
verbose_wrong	0	2	.468	total	18	159	

Table 10: COMET-22 on all LIGATUR phenomena.

D Full WMT24 Correlation Tables

Table 11 gives segment-level and system-level correlations for every metric and reference condition, per language pair. The synthetic-reference condition substitutes, for each segment, the output of a middle-ranked system (GPT-4 for cs-uk and en-hi,

ONLINE-B for en-is) as the reference.

LP	Metric	segment		system	
		nat.	synth.	nat.	synth.
cs-uk	COMET-22	.547	.480	.922	.620
	MetricX-24	.517	.442	.948	.771
	ensemble	.563	.489	.941	.719
	AEGIS	.549	—	—	—
en-hi	COMET-22	.357	.395	.937	.977
	MetricX-24	.403	.420	.975	.984
	ensemble	.417	.446	.962	.982
	AEGIS	.431	—	—	—
en-is	COMET-22	.501	.477	.954	.891
	MetricX-24	.508	.487	.960	.951
	ensemble	.547	.520	.960	.938
	AEGIS	.532	—	—	—
QE (no ref), seg.		.316	.408	.379	—

Table 11: Pearson r against WMT24 ESA gold. AEGIS is shown in its submitted configuration ($w=0.85$ with native references). The last row gives reference-free QE for cs-uk, en-hi and en-is respectively.

Table 12 gives the complete weight sweep behind Table 7 of the paper.

w	1.0	.9	.85	.8	.7	.6	.5	.3	0
native	.557	.553	.550	.546	.535	.520	.502	.458	.381
synth.	.503	.503	.502	.500	.493	.483	.471	.438	.381

Table 12: Pooled segment-level Pearson r against ensemble weight w . The optimum is $w=1$ in both conditions.

E Submitted Predictions by Language Pair

Challenge-set coverage, in scores: news 24,726; social 22,663; GAMBIP 18,312; speech 14,243; software 7,983; factchecking 6,586; edu 6,228; FECE 6,000; QEBreak 2,886; AGGREE 350; LIGATUR 342.

F Example Challenge Items

Table 1 of the main paper shows one submitted item per phenomenon class. Table 14 repeats those items here so this supplement is self-contained, and adds the commentary that did not fit in the paper. Invisible codepoints are written as tags such as <NBSP>; they occupy no visible width in the delivered files. The homoglyph example writes the substituted character as its codepoint, since it is visually identical to Latin o. The Devanagari item is transliterated, with the position of the inserted

Language pair	scores	mean	sd
ces-deu	13,578	85.1	10.1
ces-ukr	12,035	75.1	16.5
ces-vie	12,958	78.2	13.5
eng-arz	12,990	76.2	16.4
eng-bel	10,974	72.2	18.9
eng-ces	13,440	74.6	17.7
eng-deu	13,770	82.7	13.7
eng-ekk	10,680	70.1	19.3
eng-hye	10,192	65.0	25.0
eng-ind	10,261	78.8	14.2
eng-isl	11,160	65.3	21.4
eng-kaz	12,702	71.3	20.3
eng-lij	11,919	62.2	15.6
eng-lld	10,152	55.0	14.4
eng-rus	13,206	78.4	16.9
eng-sme	13,311	48.3	18.3
eng-tha	13,981	79.4	18.3
eng-ukr	12,180	76.5	15.8
eng-zho (Hans)	13,949	81.9	14.8
eng-zho (Hant TW)	9,100	81.0	9.8
zho-jpn	10,385	79.6	13.3

Table 13: Submitted AEGIS predictions on the 21 official language pairs.

zero-width non-joiner marked; in the delivered file the conjunct transliterated *rma* is broken by a joiner placed directly after its halant.

The first two rows are the cases Section 4 of the paper identifies as exactly invisible to both encoders: a leading byte-order mark, and a non-breaking space substituted for an ordinary space. In both, the corrupted and clean hypotheses tokenise identically, so no reference-based neural metric can separate them, and only the deterministic shield reaches them.

Class	Correct vs corrupted hypothesis
bom	Mitarbeiter muessen vor Ort einen Helm tragen. <BOM>Mitarbeiter muessen vor Ort einen Helm tragen.
nbsp	Der Vertrag wird automatisch verlängert. Der Vertrag wird automatisch<NBSP>verlängert.
soft_ hyphen	Die Bewerbung muss heute eingereicht werden. Die Bewerbung muss heute e<SHY>ingereicht werden.
ghost_homo	Der Arzt verschrieb Antibiotika gegen die Infektion. Der Arzt verschrieb Antib[i][U+043E]tika gegen die Infekti[U+043E]n.
conjunct	yah kaarya mahatvapuurna hai. yah kaarya mahatvapuurn<ZWNJ>na hai.
refcopy	Kein Zugang fuer Personen ohne Berechtigung. Der Zutritt ist fuer Unbefugte erlaubt.

Table 14: One item per phenomenon class. The upper line of each pair is the correct hypothesis, the lower the corrupted one. refcopy carries no invisible characters: its corrupted hypothesis reverses the meaning of the source while copying the reference’s surface form.

The refcopy row is included as a contrast.

It contains no orthographic corruption at all; the corrupted hypothesis inverts the meaning of the source (*prohibited* becomes *permitted*) while matching the reference’s phrasing more closely than the correct hypothesis does. COMET-22 prefers it in 5 of 5 cases, which is the reference-anchoring failure discussed in the paper rather than anything the shield can address.

G Hygiene Shield Specification

Table 5 of the main paper lists the shield rules with the finding that motivates each; Table 15 below repeats the penalties and adds the number of official-pair hypotheses each rule actually flagged in our submission. Penalties are additive, each capped, and applied only for character classes absent from both the source and the reference of the same segment.

Rule	per hit	cap	fired
Default-ignorable / format	2.0	8.0	131
Whitespace-class substitution	0.5	2.0	2,437
Soft hyphen	1.0	3.0	114
Joiner after halant	1.5	6.0	24
Homoglyph cloaking	2.0	8.0	198
NFC mismatch	0.5	0.5	100
Empty output	60.0 (flat)		3,127
Runaway length	≤ 40.0		608
Repetition loop	≤ 40.0		329

Table 15: Shield and degeneracy rules, in cESA points, with the number of official-pair hypotheses each flagged in the submission.

Homoglyph cloaking requires a token that (i) contains at least one Latin letter, (ii) contains at least one confusable Cyrillic or Greek codepoint, and (iii) whose confusable-folded form occurs in the source or reference. Condition (i) exempts genuine non-Latin target text; without it the rule fired on 11.9% of official hypotheses rather than 1.16%.

Runaway length triggers when the hypothesis exceeds three times the source token count, for sources of at least ten tokens. Repetition triggers when a 3–5 token *n*-gram repeats at least eight times and covers more than half the hypothesis.