

What Improves Akkadian Machine Translation, and What Only Seems To?

Francesco La Rosa
Imperial College London
fl26@ic.ac.uk

Jaehyuk Choi
University of Cambridge
jc2672@cam.ac.uk

King Shun Hanson Siu
University of Edinburgh
s2410230@ed.ac.uk

Abstract

We study machine translation of Old Assyrian Akkadian into English. Hundreds of thousands of cuneiform tablets survive, most untranslated, while the largest aligned corpus for this dialect holds roughly 6,700 sentence pairs and no pre-trained Akkadian language models exist. To find which choices matter in this setting, we compare the main ones a system builder faces. We test a byte-level (ByT5) against a subword (mT5) pretrained model family, model sizes from 300 million to 1.2 billion parameters, six preprocessing schemes drawn from Assyriological practice, and the quantity of training data, each measured with three random seeds. Two factors have reliable effects. Training data adds 17.4 points on our combined metric (the geometric mean of BLEU and chrF++) from a quarter of the corpus to all of it, and model size adds 7.3 points from 300 million to 580 million parameters, with one to two more at 1.2 billion. The remaining choices have smaller effects than a single shared fine-tuning recipe suggests. With all models tuned on one narrow learning-rate grid calibrated on the byte-level family, that family led by 17 to 21 points and every preprocessing scheme hurt the larger model. Choosing each configuration’s learning rate on validation data instead cuts that lead below two points and removes part of the apparent degradation. We recommend prioritising data quantity, then model size, and treating comparisons of model families or preprocessing as reliable only when every configuration is tuned separately.

1 Introduction

Akkadian is the oldest attested Semitic language (Huehnergard, 2011), written in cuneiform script on clay tablets across Mesopotamia from roughly 2500 to 500 BCE. Over that span the language divides into an Assyrian and a Babylonian branch, each with successive period-dialects. We study Old Assyrian, the dialect of the merchant archives of

roughly 1900 to 1700 BCE (Veenhof and Eidem, 2008). Hundreds of thousands of tablets survive, but only a fraction have been translated into any modern language, a central bottleneck for computational Assyriology (Sommerschield et al., 2023). The task poses an extreme low-resource challenge. To our knowledge the largest aligned corpus for Old Assyrian is the roughly 6,700 sentence pairs released by the Deep Past Initiative (Deep Past Initiative, 2026), which we use. The survey of machine learning for ancient languages by Sommerschield et al. (2023) records no pretrained Akkadian language model, so any neural system must either train from scratch on aligned pairs or fine-tune a multilingual model that never saw the language. The standard romanised transliteration system encodes readings through diacritics, determinatives, and Sumerographic logograms (Section 3) that have no counterpart in the languages seen during multilingual pretraining.

A builder of an Akkadian translation system must choose a model family, a model size, a preprocessing pipeline, and a data budget. Prior neural systems adopted subword tokenisation and fixed preprocessing pipelines without testing either choice (Guthertz et al., 2023; Nehme et al., 2025). We ask which of these choices improves translation quality by more than run-to-run variance. Answering that question in this setting requires care about optimisation, because fine-tuning on a few thousand pairs is acutely sensitive to hyperparameter choice, the learning rate above all (Melis et al., 2018; Dodge et al., 2019; Sennrich and Zhang, 2019). We therefore report every comparison under a single shared fine-tuning recipe, the common practice, and re-examine the largest observed effects with each configuration’s learning rate selected on validation data only. This is, to our knowledge, the first controlled comparison of byte-level and subword models for Akkadian and the first factorial study of preprocessing across model

size for any ancient language.

Contributions. We make three contributions.

1. **A reliability ranking of the design choices.**

Training data quantity has the largest effect we measured, adding 17.4 combined points from 25% to 100% of the corpus, monotone in all three seeds (Section 5.1). Model size is second, adding 7.3 points from 300 million to 580 million parameters and a further 1.3 to 2.1 at 1.2 billion (Section 5.5). No hand-crafted preprocessing intervention beat a tuned same-seed baseline by more than seed variance at any size, and the byte-versus-subword choice is worth one to two points once each family is tuned. The resulting priority order is data, then model size, then preprocessing.

2. **A measurement of how much a shared recipe distorts comparisons.**

With every model’s learning rate picked from one narrow grid calibrated on the byte-level family, the subword family trails by 17 to 21 combined points and falls below a TF-IDF retrieval baseline. With each family tuned on validation data, the gap falls below two points, and the smaller subword model overtakes its byte-level counterpart (Section 5.2). The best subword rates sit at the edge of our search range, so the true gap may be smaller still. Two 40-million-parameter models of matched architecture trained from scratch, one on bytes and one on subwords, support the same reading, with subwords winning in all three seeds.

3. **An assessment of each preprocessing intervention.**

Under the shared recipe, all six interventions hurt the 580-million-parameter model (Section 5.3). Retuning the learning rate removes the degradation for genre conditioning. Retrieval augmentation stays 0.7 points behind at every rate we tried, and dictionary glosses hurt the 1.2-billion-parameter model in all three seeds at the one rate we probed (Sections 5.4 and 5.5). We release code, configurations, and all per-run artifacts (Appendix A).

2 Related Work

Akkadian machine translation. Neural translation of Akkadian was introduced by Guthertz et al. (2023), who trained translation models from scratch on approximately 56K sentence pairs from

the Open Richly Annotated Cuneiform Corpus (ORACC), drawn primarily from Neo-Assyrian, an imperial-era dialect roughly a thousand years later than ours, and reached BLEU 37.47 with subword tokenisation. Nehme et al. (2025) applied transfer learning from a Romance-language translation model and reached BLEU 34.09 on a different ORACC split. Neither score is directly comparable to ours. Guthertz et al. trained on roughly 56K pairs and Nehme et al. on an 18K-pair ORACC subset, three to eight times more data than our single-dialect corpus, and both training sets span multiple dialects. Earlier work addressed transliteration, the conversion of cuneiform signs to readings, rather than translation (Gordin et al., 2020). In the broader ancient-language landscape, translation remains underexplored relative to restoration and attribution, where character-level models such as Ithaca already operate (Sommerschield et al., 2023; Assael et al., 2022).

Both prior systems relied on subword tokenisation and fixed preprocessing pipelines, so tokenisation and preprocessing were never examined as experimental variables, and whether philologically motivated interventions (interventions grounded in the scholarly conventions for reading these texts) such as determinative handling, logogram marking, or dictionary injection (Section 3) help neural translation remained open.

Byte-level models have since been adopted for Akkadian without such a comparison. TabletCraft (Wang, 2026) fine-tunes ByT5-base on 58K pairs, 50K of them from the same ORACC-derived corpus as Guthertz et al. (2023), and reports BLEU 49.1 on that corpus’s validation split; it includes 1.5K Deep Past pairs in training but does not evaluate on Old Assyrian. The public solution write-ups of a recent community challenge on the corpus we use also converged on ByT5-family models (Deep Past Initiative, 2026), and Old Assyrian translation is the subject of a contemporaneous preprint (Ugwumba, 2026). These systems corroborate the promise of byte-level modelling but provide no controlled comparison against subword alternatives.

Character-level translation. Subword segmentation via byte-pair encoding (BPE) (Sennrich et al., 2016) remains the dominant tokenisation strategy in neural machine translation, but character-level and byte-level models have shown advantages for morphologically rich, orthographically diverse, and low-resource languages (Lee et al., 2017; Cherry

et al., 2018; Ataman and Federico, 2018). Edman et al. (2024) compared ByT5 and mT5 across multiple language pairs and attributed ByT5’s gains to better handling of rare words and orthographic variation, exactly the phenomena our preprocessing interventions target. The high-resource evidence is more mixed (Libovický et al., 2022; Mielke et al., 2021). Comparing pretrained checkpoints also confounds tokenisation with vocabulary fit (Rust et al., 2021; Bostrom and Durrett, 2020), a caveat that applies to our mT5–ByT5 comparison, and our from-scratch contrast (Appendix J) addresses it directly. CANINE (Clark et al., 2022) shows that tokenisation-free encoders extend beyond the ByT5 family.

3 Dataset and Task

Task. We address sentence-level translation from romanised Akkadian transliterations, scholarly transcriptions of cuneiform signs into Latin characters (Huehnergard, 2011), to modern English. Several notation features are relevant to system design.

- **Diacritics.** Because multiple cuneiform signs can map to the same Latin syllable, scholars add accents and dots (s vs. š) to distinguish them.
- **Determinatives.** Unpronounced classifiers attached to words to mark semantic category, functioning much like writing “{city}London”; common examples mark divine names, personal names, and places.
- **Sumerographic logograms.** Sumerian words used as shorthand within Akkadian text, appearing as uppercase tokens (e.g., KÙ.BABBAR = “silver”).
- **Homophone subscripts.** Digits appended to sign readings (e.g., bi₄) that encode the identity of same-sounding signs rather than any phonological distinction (Huehnergard, 2011).

Dataset. The dataset comes from the Deep Past Initiative (DPI) Kaggle competition¹ and consists of Old Assyrian commercial texts dating to approximately 1900–1700 BCE, primarily from the Kültepe-Kaneš merchant archives (Veenhof and Eidem, 2008). The release contains 1,561 published documents, of which 1,170 yield document-

level parallel transliteration–translation pairs. Documents are segmented into sentences, yielding 6,746 pairs, of which 6,728 survive quality filtering; excluding the held-out documents (below) leaves 5,894 training sentences. The filters remove pairs with fewer than 10 translation characters and pairs whose two sides differ in length by more than a factor of ten. Both thresholds target alignment noise, since a pair that trips them is a fragment or a misalignment rather than a translation.

Dataset characteristics. Letters dominate the corpus (53.5% of sentences), followed by texts of unknown genre (23.2%) and debt notes (9.9%); Appendix C gives the full genre breakdown. We call a sentence *formulaic* if its English translation matches one of three structural templates common in Old Assyrian commercial documents. The templates are patronymic introductions (“X son of Y”), debt clauses (“X owes Y *n* shekels of silver”), and witness lists (“Witnesses: A, B, C”). We identified them by reading the training translations and encoded each as a regular expression over the English text. By this criterion, 39.3% of sentences are formulaic. The share varies strongly by genre. 88% of debt-note sentences are formulaic against 18% of letters. Averaged over the three baseline seeds, ByT5-base scores 53.6 combined points on debt-note sentences of the independent test set and 29.3 on letters, an 83% relative gap that far exceeds any preprocessing effect observed in this study.

Test sets. Results are reported on two holdout sets so that every effect can be checked for replication under a second, independently constructed split. An effect that appears on only one holdout may be an artifact of how that holdout was built, and we report these replication checks throughout. Both sets are drawn from the published documents but re-segmented and filtered separately from the training alignment, so their row counts are not a subset of the 6,728 pairs above. The *independent* set (951 rows) comes from a genre-stratified sample of 152 documents (13% of the 1,170 pair-bearing documents), of which 149 contribute rows after cleaning. The *new_test* set (661 rows, 130 documents) comes from a separate genre-stratified holdout with stricter quality filters. The two holdout lists share no documents, and their union of 282 document IDs is excluded from all training splits. Formulaic language does cross document boundaries. 3.7% of independent-test sentences (3.2% for *new_test*) have an exact source-side duplicate in

¹<https://www.kaggle.com/competitions/deep-past-initiative-machine-translation>

the training split, and 6.1% (7.3%) have a training sentence with character 5-gram Jaccard similarity of at least 0.8, concentrated in the formulaic genres (15.6% and 20.3% of debt notes respectively).

Validation split. After excluding the 282 holdout documents, the remaining corpus is split 90/10 at the document level (seed 42), preventing within-document leakage. This validation set drives early stopping and learning rate selection.

4 Methodology

4.1 Preprocessing Pipeline

All models receive source text processed through a deterministic five-step pipeline (Unicode normalisation, scribal notation removal, homophone subscript stripping, bracket normalisation, and whitespace normalisation), applied identically to every experimental condition. Appendix D gives the full specification. The interventions described in Section 4.4 modify the output of this pipeline or apply additional transformations.

4.2 Models

We frame model selection as a 2×2 factorial design crossing tokenisation strategy (subword against byte-level) with model size (300M against 580M parameters). All neural models are encoder-decoder Transformers from the T5 family, fine-tuned from pretrained multilingual checkpoints (Raffel et al., 2020). The four conditions are mT5-small, mT5-base, ByT5-small, and ByT5-base; Appendix B lists their configurations. An exploratory ByT5-large condition (1.2B parameters) extends the size axis beyond the factorial (Section 5.5).

The mT5 models use a 250K-token SentencePiece (Kudo and Richardson, 2018) vocabulary trained on mC4 (Xue et al., 2021). The ByT5 models operate directly on UTF-8 bytes, 256 byte values plus special tokens, a 384-entry embedding table in the released checkpoints (Xue et al., 2022). Because the SentencePiece vocabulary was learned from modern web text, it has almost no merges for diacriticked transliteration syllables, so mT5 breaks each source word into many small pieces. On our normalised training sources it emits 4.95 pieces per whitespace-separated word, 53% of them single characters, against 1.91 pieces per word on the English targets, and this fragmentation discards the syllabic boundaries that carry meaning in the transliteration system. ByT5’s sequences are longer still, at 8.55 bytes per word, but its segmentation

	ByT5 base	ByT5 small	mT5 base	mT5 small
Learning rate	7e-5	1e-4	5e-5	5e-5
Batch size	8	16	8	16
Grad. accum. steps	4	2	4	2
Epochs	20	20	20	20
Warmup steps	200	200	200	200
ES patience	6	6	6	6

Table 1: Per-model baseline hyperparameters; effective batch size is 32 throughout. Learning rates are the first-round validation selections (Section 4.3); the widened second-round selections are reported in Section 5.2.

is uniform by design rather than an artifact of a mismatched vocabulary.

As a non-neural reference, we include a **TF-IDF retrieval baseline**. It vectorises character 3–5 grams, retrieves the most similar training source by cosine similarity, and returns that source’s training translation.

4.3 Training Protocol

All neural models are fine-tuned from their pre-trained checkpoints with Adafactor (Shazeer and Stern, 2018), linear warmup followed by cosine decay (Loshchilov and Hutter, 2017), gradient clipping at max norm 1.0, BF16 mixed precision, label smoothing of 0.1, weight decay of 0.01, and early stopping on the validation combined score. Table 1 lists the per-model settings. All four baselines use an effective batch size of 32. Label smoothing, weight decay, and warmup were not tuned. We fixed label smoothing following Szegedy et al. (2016). Appendix E gives the remaining detail, including sequence-length limits, the retrieval-condition adjustments, and inference settings.

Learning rate protocol. Learning rates were selected in two rounds, and every selection used validation data only. In the first round, each of the four factorial models picked its rate from the grid {5e-5, 7e-5, 1e-4}, and every preprocessing intervention inherited the rate of its size-matched baseline. This grid was calibrated on the byte-level models. Both mT5 models picked the boundary value of this grid. A search that ends at a boundary warns that the best rate may lie outside the searched range, here a range built for the other model family. In the second round we therefore searched further, per configuration. mT5 searched {3e-4, 1e-3} at both sizes. The two interventions we retuned, genre conditioning and retrieval augmentation, searched {5e-5, 1e-4} at 580M against their inherited 7e-5.

The 580M baseline was retrained at $1e-4$, and a ByT5-large probe searched $\{5e-5, 7e-5\}$. Section 5 reports results under the shared first-round recipe, then shows which conclusions change under the per-configuration rates.

4.4 Preprocessing Interventions

Six interventions are tested on both ByT5-small (300M) and ByT5-base (580M), grouped by design methodology.

Philologically motivated. Three interventions derived from Assyriological conventions, designed before observing any model output. (1) **Determinative stripping.** Determinatives (Section 3) are removed from the source, testing whether the model can infer semantic class from context alone. (2) **Sumerogram tagging.** Sumerographic logograms are wrapped with explicit `<SUM> . . . </SUM>` delimiters, testing whether marking the Sumerian–Akkadian code-switching boundary helps. (3) **Dictionary gloss injection.** Up to six glosses from the Concise Dictionary of Akkadian (Black et al., 2000) are appended to each source sentence, providing candidate translations for the corpus’s many rare words.

Error-driven. Three interventions designed from manual categorisation of 150 randomly sampled ByT5-base validation predictions, which found proper name errors in 60.7% of examples and numeral/quantity errors in 34.6%, alongside the genre gap (Section 3). (4) **Genre conditioning.** Following the control-token approach introduced for multilingual neural machine translation by Johnson et al. (2017) and applied to domain control by Kobus et al. (2017), a genre control token (e.g. `genre_letter`, `genre_debt_note`) is prepended to each source sentence. (5) **Numeral normalisation.** Decimal and fractional tokens are annotated with canonical numeric tags, converting ambiguous number notation to explicit fraction representations. (6) **Retrieval augmentation.** The most similar training example (TF-IDF character 3–5 grams, cosine similarity, self-exclusion) is prepended as context with delimiter tokens (Khan-delwal et al., 2021; Cai et al., 2021; Zhang et al., 2023), truncated to 160 source and 220 translation bytes. Maximum source length rises to 768 tokens. Two repair variants, *compact* and *source-only*, are described with the format ablation in Appendix F.

4.5 Data Conditions

To assess training data quantity, ByT5-base is trained on document-level subsets containing 25%, 50%, 75%, and 100% of the eligible non-holdout gold-corpus documents (subset selection seed 42; three training seeds each). An additional condition combines the full gold corpus with an Old Assyrian-focused external alignment mix of about 29,000 sentence pairs spanning multiple Akkadian dialects, testing whether data from other dialects substitutes for in-domain data (configuration in Appendix E).

4.6 Evaluation

We report corpus-level BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017) (`word_order=2`) computed with sacrebleu (Post, 2018), summarised as $\text{combined} = \sqrt{\text{BLEU} \times \text{chrF++}}$. The two metrics err in different directions. BLEU counts exact word n-gram matches, so with a single reference it gives no credit for a nearly correct word. chrF++ counts character n-grams, so it gives partial credit for near matches, which matters for the transliterated proper names that fill these texts. A system scores well on the geometric mean only by doing well on both.

Both ByT5 baselines and all six interventions are run with seeds {42, 52, 62} at both model sizes, as is the external data condition. For each intervention i and model size, the change relative to baseline is computed against the same-seed baseline run and averaged over the three seeds ($\bar{\Delta}_{\text{small},i}$ and $\bar{\Delta}_{\text{base},i}$), which also yields a per-intervention variance estimate. We test the capacity $\times$ preprocessing interaction with a one-sided sign test (Conover, 1999) over the six seed-averaged differences $\bar{\Delta}_{\text{small},i} - \bar{\Delta}_{\text{base},i}$: under the null that capacity favours neither direction, six same-direction signs have one-sided probability $(1/2)^6 \approx 0.016$ (two-sided: 0.031). The direction was fixed before the multi-seed matrix ran, from a single-seed pilot under the same recipe, so the one-sided test was specified in advance of the data it is applied to. Appendix L details which conditions come from which software and hardware environment and why scores are comparable across them.

5 Experiments

5.1 Data Scaling

Figure 1 presents ByT5-base performance as a function of training data quantity, with document-level

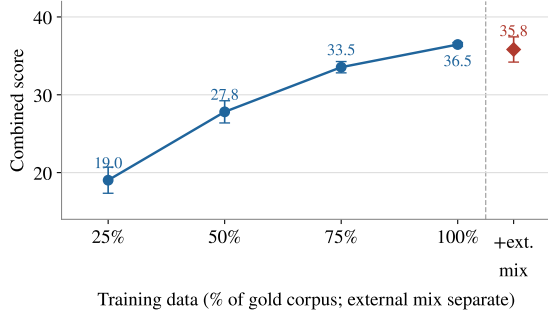


Figure 1: Data scaling curve for ByT5-base (independent test). Every point is the mean over seeds {42, 52, 62} with a $\pm\sigma$ error bar; the external-mix point is offset from the percentage axis.

subsets of 25%, 50%, and 75% each trained under three seeds in the same environment as the ByT5-base baseline of Section 5.2. Seed-averaged combined scores rise from 19.03 (25%) through 27.81 (50%) and 33.55 (75%) to 36.46 (100%), and the ordering is monotone within every individual seed. The curve replicates on new_test, rising from 19.30 through 28.37 and 33.99 to 36.98, again monotone within every seed. Growth is sub-linear, consistent with established data scaling relationships for machine translation (Bansal et al., 2022). The 50% subset reaches 76% of the full-data score, and the jump from 75% to 100% adds 2.9 combined points. The gains flatten at the top end, but they do not disappear. The full-range gain of +17.4 points is larger than any model-choice or preprocessing effect that survives retuning elsewhere in this study (paired-bootstrap 95% interval for the 100% versus 25% gap at seed 42: [+15.8, +18.2]; Appendix I). Because every run trains for the same 20 epochs, the update budget grows with the data. The 25% subset takes 174 mini-batches per epoch against 662 for the full corpus, and early stopping never triggered, so the best validation checkpoint fell between epochs 16 and 20 in every scaling run. The measured gain is therefore the effect of more data at a fixed epoch budget, which combines more examples with more optimiser updates, and we did not run a step-matched control to separate the two.

Adding the external mix at 580M does not replicate across seeds. The gain is +1.11 on seed 42 against the same-environment, same-seed baseline, but -1.37 and -1.70 on seeds 52 and 62 ($\sigma = 1.62$; Appendix G), consistent with the mix spanning dialects beyond Old Assyrian. A single ByT5-small run gains +4.24 from the same mix,

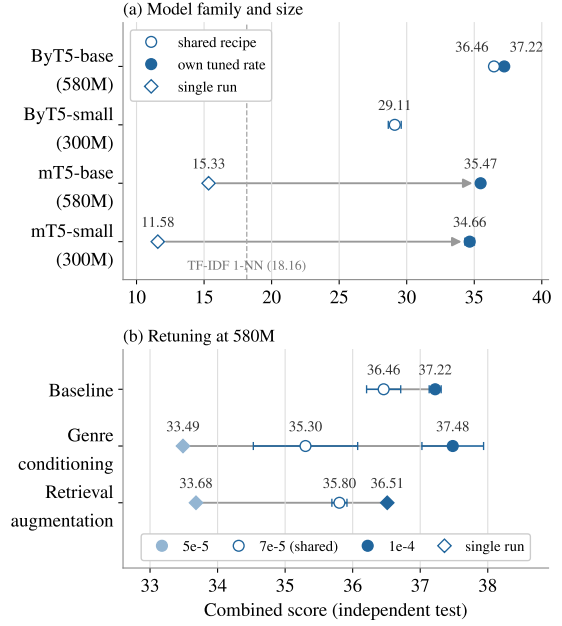


Figure 2: Learning-rate tuning changes both comparisons. (a) Each configuration under the shared recipe (open markers) and at its own tuned rate (filled markers). (b) The 580M retuning probe of Section 5.4. Circles are means over seeds {42, 52, 62} with $\pm\sigma$ bars; diamonds are single runs.

unreplicated across seeds. In our runs, additional data helped consistently only when it came from the same dialect.

5.2 Model Family, Model Size, and the Learning Rate

Figure 2a presents the 2×2 factorial results alongside the TF-IDF baseline, under both tuning protocols (per-metric scores in Appendix H). Under the shared recipe, the difference between families is large. Byte-level models beat subword models by +17.5 combined points at 300M parameters and +21.1 at 580M, and the TF-IDF baseline (18.16) surpasses both mT5 sizes, which would ordinarily indicate models badly mismatched to the task. The vocabulary evidence of Section 4.2 offers a ready explanation, and prior work associates byte-level models with exactly such orthographically irregular input (Edman et al., 2024).

Per-configuration tuning removes most of this difference (Figure 2a). With learning rates selected on three-seed mean validation combined from the widened grid, both mT5 models land at 1e-3 and recover more than 20 combined points. mT5-base reaches 35.47 ± 0.18 against the ByT5-base mean of 36.46 ± 0.25 , and the same ordering holds on

new_test (36.33 against 36.98). The shared-recipe ByT5 advantage shrinks to between one and two points, although base-size parity is not established, and against the 580M baseline returned to $1e-4$ (Section 5.4) the gap is 1.75. The selected mT5 rates and ByT5-small’s shared-recipe rate all sit at their grid edges, so a wider search could still improve those models. Paired bootstrap over test sentences puts the retuned gap’s 95% interval above zero in every seed, while the gap against the shared-recipe reference is within sentence-level noise (Appendix I).

We read the 17-to-21-point family gap as a property of the shared recipe. The vocabulary mismatch and the fertility difference are real, but in our runs their measurable cost is a shifted learning-rate optimum. The practical difference on this corpus is that ByT5 tolerated the shared grid and mT5 did not. At the shared-recipe rate mT5 ignores the source and emits a high-frequency formula, an optimisation failure rather than a representation limit (examples in Appendix K).

A from-scratch control removes pretraining entirely. We train T5 models of matched architecture (four encoder and four decoder layers, hidden size 512, roughly 40M parameters) on our corpus alone, one on UTF-8 bytes and one with a 4,000-piece SentencePiece vocabulary learned from the training data, under an identical schedule with three seeds each (Appendix J). The subword model wins in every seed, reaching 18.74 ± 0.60 combined against 13.21 ± 1.06 for bytes. Together with the tuned mT5 results, this argues against a general byte-level advantage in our experiments.

5.3 Preprocessing Intervention Effects

Table 2 presents all six preprocessing interventions applied to both ByT5-small and ByT5-base, with each condition run under three seeds. All six satisfy $\bar{\Delta}_{\text{small},i} > \bar{\Delta}_{\text{base},i}$: seed-averaged, the small model benefits more (or is hurt less) than the base model in every case (Figure 3). Applying the sign test defined in Section 4.6, six unanimous signs yield $p = (1/2)^6 \approx 0.016$. The six differences are not fully independent, because at each size every intervention is compared against the same three baseline runs, so we read the test as evidence about this intervention set rather than about preprocessing in general (see Limitations). Robustness checks that drop this independence assumption, per-seed signs and leave-one-seed-out means, give the same picture (Appendix I).

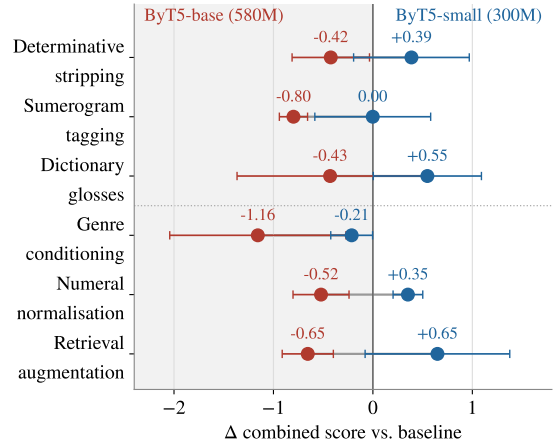


Figure 3: Capacity $\times$ preprocessing interaction on the independent test set. Each row is the seed-averaged delta against the same-size baseline, with $\pm\sigma$ bars. The dotted rule separates the philological (top) from the error-driven (bottom) interventions.

All results in this subsection are measured under the shared recipe. Section 5.4 tests how far they survive retuning. The mean seed-averaged delta across interventions is $+0.29$ for the small model and -0.66 for the base model (Table 2). The base model degrades under all six interventions; the small model improves under four and is essentially unchanged under the other two. Both intervention families show the same pattern, so the interaction does not depend on how the intervention was designed. Per-intervention standard deviations range from 0.07 to 0.77 combined points and rival several of the mean deltas, so individual effect sizes should be read directionally; the sign test, which uses direction alone, is the evidence we rely on. The interaction holds on BLEU and chrF++ separately (Appendix I). On new_test the base model’s mean delta is -0.63 , close to the -0.66 on the independent set, and the direction replicates for five of six interventions; the small model’s average gain drops to $+0.06$ and does not replicate.

Seed-averaged deltas also conceal spread. Dictionary glosses at 580M average -0.43 from per-seed deltas of $+0.19$, $+0.03$, and -1.50 , so the mean direction rests on one seed. A single-seed ablation of retrieval input formats appears in Appendix F.

5.4 Retuning the Harmful Interventions

If an intervention changes the input distribution, it can also change the optimal learning rate, and a fixed recipe would then register the mismatch as a

Intervention	ByT5-small (300M)		ByT5-base (580M)		new_test $\bar{\Delta}$	
	Combined	$\bar{\Delta}$	Combined	$\bar{\Delta}$	Small	Base
Baseline	29.11 $\pm$ 0.48	—	36.46 $\pm$ 0.25	—	—	—
1. Strip determinatives	29.50 $\pm$ 0.12	+0.39	36.04 $\pm$ 0.32	−0.42	+0.36	−0.49
2. Tag Sumerograms	29.11 $\pm$ 0.45	−0.00	35.66 $\pm$ 0.23	−0.80	−0.06	−0.40
3. Dictionary glosses	29.66 $\pm$ 0.07	+0.55	36.03 $\pm$ 0.73	−0.43	+0.56	−0.40
4. Genre conditioning	28.90 $\pm$ 0.27	−0.21	35.30 $\pm$ 0.77	−1.16	−0.19	−0.80
5. Numeral normalisation	29.47 $\pm$ 0.38	+0.35	35.94 $\pm$ 0.32	−0.52	+0.12	+0.14
6. Retrieval augmentation	29.76 $\pm$ 0.27	+0.65	35.80 $\pm$ 0.11	−0.65	−0.43	−1.81
Mean		+0.29		−0.66		

Table 2: Preprocessing intervention effects (combined score), mean $\pm$ standard deviation over seeds {42, 52, 62}. $\bar{\Delta}$ is the per-seed delta against the same-seed, same-size ByT5 baseline, averaged over seeds.

degradation. We tested this directly on genre conditioning (the largest degradation at 580M) and retrieval augmentation, retraining each at 580M with learning rates {5e-5, 1e-4} against the inherited 7e-5 (Figure 2b). At 1e-4 genre conditioning reaches a three-seed mean of 37.48 $\pm$ 0.46 and retrieval augmentation 36.51 (single run), recovering from shared-recipe scores of 35.30 and 35.80. The recovery does not show that retuned interventions beat the original baseline, because the baseline moves too. Retrained at 1e-4 with three seeds, the baseline reaches 37.22 $\pm$ 0.09, and in this environment 1e-4 is also the validation-selected rate for the baseline (mean best-validation 36.38 against 34.78 at 7e-5). The 36.46 reference is therefore itself under-tuned, and we keep it only as the fixed point that all shared-recipe comparisons were measured against.

Against the retuned baseline the two interventions separate. Genre conditioning’s nominal margin is +0.26, within cross-seed variation. Retrieval augmentation sits at −0.71, essentially its shared-recipe deficit of −0.65. Whether any intervention helps at matched tuning remains unresolved in both directions. The remaining four interventions keep their shared-recipe rates, so their measured degradations may overstate the true effect.

5.5 Extending the Size Axis to 1.2B

We label this arm exploratory because a two-value learning rate probe is not the tuning the other sizes received. ByT5-large (1.2B) trained on the full corpus reaches a three-seed mean of 38.55 combined at 5e-5, and a single run at 7e-5 reaches 39.89, the best score in this study. The three-seed mean sits +2.1 above the shared-recipe ByT5-base reference and +1.3 above the retuned 580M baseline of Section 5.4. Gains from model size therefore continue beyond 580M in this arm, at a diminishing rate

consistent with the 300M→580M gain of +7.3.

The 1.2B arm also carries the one intervention result that survives matched-rate comparison. Dictionary glosses, trained at the same 5e-5 as their baseline, are negative in all three seeds (−0.96, −0.14, −1.25; mean 37.76), with sentence-level bootstrap intervals excluding zero in two of the three (Appendix I). This degradation cannot be attributed to an inherited learning rate, though we did not sweep rates for the gloss condition itself. With the gloss pattern at smaller sizes (+0.55 at 300M, −0.43 at 580M), the direction is consistent with larger models gaining nothing from appended dictionary material, but with a single intervention and a two-value rate probe we cannot generalise beyond it.

5.6 Discussion

The learning rate is part of the experimental condition. Every configuration change we tested that altered the input distribution or the model family also moved the optimal learning rate, sometimes by an order of magnitude, and a shared recipe converted those moves into spurious quality differences of 17 to 21 combined points. This echoes [Melis et al. \(2018\)](#), [Dodge et al. \(2019\)](#), and the low-resource case study of [Sennrich and Zhang \(2019\)](#), who showed that comparisons are only meaningful when the hyperparameters suit all parties. In a low-resource setting the risk is greater, because a sweep is easy to omit.

What the ranking adds. That more data and larger models improve translation quality is expected ([Bansal et al., 2022](#); [Sennrich and Zhang, 2019](#)), and we do not claim otherwise. The contribution is to measure those two effects on one corpus, with the same seeds and metric, against the model-family and preprocessing effects, and

to show that a shared fine-tuning recipe inflates the second group by an amount comparable to the largest genuine effect.

6 Conclusion

We asked what improves machine translation of Old Assyrian Akkadian and what only seems to. Data quantity has the largest effect. Growing the training data from 25% to 100% of the corpus adds 17.4 combined points, monotone in every seed. Model size is second, adding 7.3 points from 300M to 580M and one to two more at 1.2B. The remaining effects are smaller than a shared recipe suggests. Under one shared fine-tuning recipe, the byte-level family led the subword family by 17 to 21 points and every preprocessing scheme hurt the larger model. With each configuration’s learning rate selected on validation data, the family gap falls below two points, the genre-conditioning degradation disappears, and only retrieval augmentation and dictionary glosses at 1.2B keep measurable deficits. Two findings are likely to apply beyond this corpus. In a setting this small, sweeping the learning rate for each configuration changes outcomes more than any modelling choice we tested, and a comparison is trustworthy only against a same-seed tuned baseline.

Limitations

Several results rest on single runs. The retrieval recovery at $1e-4$, the ByT5-large run at $7e-5$, the TF-IDF baseline, and the retrieval-format ablation each ran once, and we label every such number where it appears. The shared-recipe baselines and all six interventions use three random seeds at both model sizes, but effect magnitudes remain modest relative to seed variance, with per-intervention σ up to 0.77 combined points, so individual deltas should be read directionally. The sign test aggregates direction, not magnitude. The data-scaling runs share an epoch budget, so update count grows with data quantity (Section 5.1).

Within each seed, all intervention comparisons at a given model size share one baseline run. Seed-averaging attenuates but does not eliminate the resulting dependence between signs. The check on the independently constructed new_test set reduces this concern, since five of six interventions replicate the direction there. Some seed-averaged deltas rest on one dominant seed, as the gloss spread at 580M shows (+0.19, +0.03, −1.50).

The retuning is partial. We retuned the learning rate for only two of the six interventions, so the fraction of the shared-recipe interaction that is optimisation artifact is bounded below. Only the learning rate was retuned. Batch size, schedule, and other hyperparameters stayed shared throughout, so further recipe sensitivity cannot be excluded. Retrieval variants are the exception. Their longer inputs forced an effective batch size of 24 against 32 elsewhere (Appendix E), so their deficits fold in any effect of that change. The data-scaling subsets and the 1.2B arm inherit fixed rates, so those axes were not subjected to the per-configuration tuning this paper advocates. The selected mT5 rates and ByT5-small’s shared-recipe rate sit at the edge of their grids, so their optima were not bracketed.

Decoding caps every hypothesis at 256 tokens for the pretrained models. A token is a byte for ByT5 and a subword piece for mT5, so the cap binds ByT5 sooner. Only 14 independent-test references exceed the byte budget this implies, and removing them does not change the family comparison.

Formulaic overlap with the training pool (Section 3) is a further caveat. Removing the 101 independent-test rows with a training near-duplicate at character TF-IDF similarity of at least 0.90 does not reopen the family gap. The external mix was not de-duplicated against the holdout documents before training. Its publicly available component (10,677 of the roughly 29K pairs) contains near-duplicates of 1.6% of independent-test and 2.1% of new_test sentences (character 5-gram Jaccard ≥ 0.8), and the remainder derives from publication scans we cannot audit retrospectively.

The family contrast compares pretrained checkpoints that differ in vocabulary, parameter allocation, and pretraining data, so it measures the practical choice between available model families rather than tokenisation in isolation. Our experiments cover Old Assyrian commercial texts from the Kültepe-Kaneš archives. Other Akkadian varieties with broader vocabulary and more complex syntax may behave differently, and our conclusions are demonstrated for one corpus, one language pair, and one model family lineage. The postprocessing pipeline was held fixed and never ablated, and the error analysis behind interventions 4 to 6 was performed by the authors on unblinded outputs rather than through formal expert annotation. We did not test decoder-only language models, and cross-dialect transfer was probed only through the

external-mix condition; both are natural extensions.

References

- Yannis Assael, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2022. [Restoring and attributing ancient texts using deep neural networks](#). *Nature*, 603:280–283.
- Duygu Ataman and Marcello Federico. 2018. [Compositional representation of morphologically-rich input for neural machine translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 305–311.
- Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. 2022. [Data scaling laws in NMT: The effect of noise and architecture](#). In *International Conference on Machine Learning*, volume 162, pages 1466–1482. PMLR.
- Jeremy Black, Andrew George, and Nicholas Postgate. 2000. *A Concise Dictionary of Akkadian*. Harrassowitz Verlag, Wiesbaden.
- Kaj Bostrom and Greg Durrett. 2020. [Byte pair encoding is suboptimal for language model pretraining](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624.
- Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. [Neural machine translation with monolingual translation memory](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 7307–7318.
- Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. [Revisiting character-based neural machine translation with capacity and compression](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305.
- Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. [CANINE: Pre-training an efficient tokenization-free encoder for language representation](#). *Transactions of the Association for Computational Linguistics*, 10:73–91.
- W. J. Conover. 1999. *Practical Nonparametric Statistics*, 3rd edition. Wiley.
- Deep Past Initiative. 2026. [Deep past challenge: Translate Akkadian to English](#). Kaggle competition. Competition ended March 2026.
- Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. [Show your work: Improved reporting of experimental results](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194.
- Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024. [Are character-level translations worth the wait? Comparing ByT5 and mT5 for machine translation](#). *Transactions of the Association for Computational Linguistics*, 12:392–410.
- Shai Gordin, Gai Gutherz, Ariel Elazary, Avital Romach, Enrique Jiménez, Jonathan Berant, and Yoram Cohen. 2020. [Reading Akkadian cuneiform using natural language processing](#). *PLOS ONE*, 15(10):e0240511.
- Gai Gutherz, Shai Gordin, Luis Sáenz, Omer Levy, and Jonathan Berant. 2023. [Translating Akkadian to English with neural machine translation](#). *PNAS Nexus*, 2(5):pgad096.
- John Huehnergard. 2011. *A Grammar of Akkadian*, 3rd edition. Eisenbrauns, Winona Lake, IN.
- Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. [Google’s multilingual neural machine translation system: Enabling zero-shot translation](#). *Transactions of the Association for Computational Linguistics*, 5:339–351.
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. [Nearest neighbor machine translation](#). In *International Conference on Learning Representations*.
- Catherine Kobus, Josep Crego, and Jean Senellart. 2017. [Domain control for neural machine translation](#). In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395.
- 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.
- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. [Fully character-level neural machine translation without explicit segmentation](#). *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Jindřich Libovický, Helmut Schmid, and Alexander Fraser. 2022. [Why don’t people use character-level machine translation?](#) In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2470–2485.

- Ilya Loshchilov and Frank Hutter. 2017. [SGDR: Stochastic gradient descent with warm restarts](#). In *International Conference on Learning Representations*.
- Gábor Melis, Chris Dyer, and Phil Blunsom. 2018. [On the state of the art of evaluation in neural language models](#). In *International Conference on Learning Representations*.
- Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. [Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP](#). *arXiv preprint arXiv:2112.10508*.
- Najat Nehme, Danielle Azar, Diana Kutsalo, and Jalal Possik. 2025. [Translating Akkadian transliterations to English with transfer learning](#). In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, volume 3, pages 1048–1053.
- 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.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation*, pages 186–191.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. [How good is your tokenizer? On the monolingual performance of multilingual language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3118–3135.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.
- Rico Sennrich and Biao Zhang. 2019. [Revisiting low-resource neural machine translation: A case study](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy.
- Noam Shazeer and Mitchell Stern. 2018. [Adafactor: Adaptive learning rates with sublinear memory cost](#). In *International Conference on Machine Learning*, volume 80, pages 4596–4604. PMLR.
- Thea Sommerschildt, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando de Freitas. 2023. [Machine learning for ancient languages: A survey](#). *Computational Linguistics*, 49(3):703–747.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. [Rethinking the Inception architecture for computer vision](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826.
- Nnaemeka Kingsley Ugwumba. 2026. [Neural machine translation of Old Assyrian cuneiform business records into English](#). Research Square preprint.
- Klaas R. Veenhof and Jesper Eidem. 2008. *Mesopotamia: The Old Assyrian Period*. Number 160/5 in *Orbis Biblicus et Orientalis*. Academic Press Fribourg / Vandenhoeck & Ruprecht, Fribourg / Göttingen.
- Zhaohui Geoffrey Wang. 2026. [TabletCraft: Bridging a 4,000-year cultural gap with bidirectional Akkadian NMT and cuneiform rendering](#). In *Proceedings of the 4th Workshop on Cross-Cultural Considerations in NLP (C3NLP 2026)*, pages 132–136, San Diego, California, United States. Association for Computational Linguistics.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *arXiv preprint arXiv:1609.08144*.
- Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. [ByT5: Towards a token-free future with pre-trained byte-to-byte models](#). *Transactions of the Association for Computational Linguistics*, 10:291–306.
- 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*, pages 483–498.
- Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. [Prompting large language model for machine translation: A case study](#). In *International Conference on Machine Learning*, volume 202, pages 41092–41110. PMLR.

A Reproducibility

Every experimental condition is defined by a version-controlled configuration file. Every reported number is backed by a per-run artifact bundle that stores the metrics for both held-out test sets alongside a snapshot of the exact configuration that produced them. Source normalisation is a deterministic, rule-based pipeline (Appendix D), so the preprocessing variants and data splits can be rebuilt exactly with the build scripts in our code release. All multi-seed conditions use seeds {42, 52, 62}, with per-seed scores in Appendix G. Training and evaluation ran on a single 80GB GPU with BF16 precision and gradient checkpointing, and the full multi-seed matrix completed in under 24 hours on that device. We do not redistribute any corpus. The primary data are available from the public Deep Past Initiative Kaggle competition under the competition’s terms, and the external mix derives from a publicly available ORACC-based alignment dataset. Code, configurations, dataset-build scripts, and per-run artifacts are released under the MIT licence at <https://github.com/larosafrancesco289/akkadian-mt>.

B Model Configurations

Model	Tokenisation	Params	HuggingFace ID
mT5-small	SentencePiece	300M	google/mt5-small
mT5-base	SentencePiece	580M	google/mt5-base
ByT5-small	Byte-level	300M	google/byt5-small
ByT5-base	Byte-level	580M	google/byt5-base

Table 3: Model configurations in the 2×2 factorial design.

C Genre Distribution

Genre	Count	%	Combined
Letter	3,606	53.5	29.3
Unknown	1,562	23.2	43.5
Debt note	669	9.9	53.6
Memo	427	6.3	53.0
Note	159	2.4	40.1

Table 4: Genre distribution of the aligned corpus (6,746 pairs), with the ByT5-base combined score computed per genre on the independent test set and averaged over the three baseline seeds. Listed genres cover 95% of sentences; genres with fewer than 20 test sentences are omitted.

D Preprocessing Pipeline Details

All models receive source text processed through a deterministic five-step pipeline, applied identically to every experimental condition. (1) **Unicode normalisation**. NFC normalisation, character mapping for test-set compatibility, and conversion of Unicode subscript digits to ASCII. (2) **Scribal notation removal**. Editorial markers indicating scholarly confidence are stripped. (3) **Homophone subscript stripping**. Scholarly subscript digits appended to cuneiform sign readings (e.g. $bi_4 \rightarrow bi$) are removed via a regex that matches any trailing digit after a transliteration character:

$(?<=[a-zA-Z\u00C0-\u01E2B])([0-9])(?=[-\\s]|$)$

This collapses orthographic variants that encode sign identity rather than phonological distinction (Huehnergard, 2011). (4) **Bracket normalisation**. Determinative markers are standardised, damage notation is converted to structured gap tokens (`<big_gap>`, `<gap>`), and half-brackets are stripped. The gap-token conversion does nothing on the sentence-aligned release, which contains no damage notation. The step exists for raw-document inputs. (5) **Whitespace normalisation**. Multiple spaces are collapsed and edges trimmed.

E Training Protocol and Hyperparameters

The optimiser, schedule, regularisation, and per-model hyperparameters appear in Section 4.3 (Table 1); training additionally uses gradient checkpointing. The four baseline models use maximum source and target lengths of 512 and 384 tokens respectively. Retrieval variants reduce the effective batch size to 24 because their inputs are longer, and they raise the source limit to 768 tokens (full retrieval) or 640 tokens (compact and source-only retrieval). The external data condition uses 10 epochs, 400 warmup steps, and early stopping patience 4.

Inference. Final evaluation uses beam search with 8 beams and a length penalty of 1.3. Wu et al. (2016) report diminishing returns beyond beam width 4 for neural machine translation. We use 8 beams as a conservative margin given the short dataset and low inference cost. The length penalty compensates for beam search’s brevity bias, relevant here because English targets are typically longer than Akkadian source transliterations. Postprocessing consists of gap token removal, repeated phrase and n-gram deduplication, punctuation normalisation, and trailing fragment

removal. Training-time validation uses greedy decoding without postprocessing.

F Retrieval Format Ablation

Retrieval augmentation shows the largest seed-averaged small-model gain and the largest interaction (+0.65 for small, −0.65 for base). To test whether the base-model degradation is specific to the input format, we evaluate two simplified variants. The *compact* variant truncates both retrieved fields to 96 bytes, and the *source-only* variant removes the retrieved translation entirely. Table 5 reports this ablation, which uses a single seed (42). The baseline and full-retrieval rows show the corresponding single-seed scores for internal comparability. Both variants worsen performance for both model sizes, so the degradation is not confined to one retrieval format.

The small model benefits from full retrieval (+0.65 seed-averaged; +1.31 in the single-seed comparison of Table 5) but only when both retrieved source and translation are present. Removing the retrieved translation eliminates the gain, suggesting that the useful signal comes from the paired example rather than from source-side similarity alone. The base model does not benefit from retrieved context in any tested format.

Variant	ByT5-small	ByT5-base
No retrieval	28.69	36.58
Full retrieval	30.00	35.28
Compact retrieval	26.43	29.89
Source-only retrieval	26.22	26.66

Table 5: Retrieval format ablation (independent test, combined score; single seed, earlier environment). The no-retrieval row therefore differs from the seed-42 baselines of Table 6, which come from the later environment (Appendix L).

G Per-Seed Scores

Condition	Independent	New_test
<i>ByT5-small baseline</i>		
Seed 42	29.35	30.09
Seed 52	29.43	30.34
Seed 62	28.56	29.90
$\mu \pm \sigma$	29.11 ± 0.48	30.11 ± 0.22
<i>ByT5-base baseline</i>		
Seed 42	36.19	37.09
Seed 52	36.49	36.85
Seed 62	36.69	37.02
$\mu \pm \sigma$	36.46 ± 0.25	36.98 ± 0.12
<i>ByT5-base baseline (earlier environment)</i>		
Seed 42	36.58	37.19
Seed 52	36.27	36.93
Seed 62	36.57	37.29
$\mu \pm \sigma$	36.47 ± 0.18	37.14 ± 0.19
<i>ByT5-base + external data (earlier environment)</i>		
Seed 42	37.69	38.28
Seed 52	34.90	36.07
Seed 62	34.87	35.97
$\mu \pm \sigma$	35.82 ± 1.62	36.77 ± 1.31

Table 6: Multi-seed combined scores. μ and σ are sample mean and standard deviation across seeds {42, 52, 62}. The external-data deltas in Section 5.1 are computed against the earlier-environment baseline block.

Condition	Independent	New_test
mT5-small (1e-3)		
Seed 42	34.81	35.62
Seed 52	34.26	35.13
Seed 62	34.92	36.18
$\mu \pm \sigma$	34.66 ± 0.36	35.64 ± 0.53
mT5-base (1e-3)		
Seed 42	35.63	37.01
Seed 52	35.51	36.52
Seed 62	35.27	35.47
$\mu \pm \sigma$	35.47 ± 0.18	36.33 ± 0.78
mT5-base (3e-4, not selected)		
Seed 42	36.83	36.94
Seed 52	35.96	36.76
Seed 62	36.07	36.70
$\mu \pm \sigma$	36.28 ± 0.48	36.80 ± 0.13
ByT5-base baseline (1e-4)		
Seed 42	37.13	38.38
Seed 52	37.24	38.40
Seed 62	37.30	38.47
$\mu \pm \sigma$	37.22 ± 0.09	38.41 ± 0.05
Genre conditioning (1e-4)		
Seed 42	36.97	38.69
Seed 52	37.84	38.67
Seed 62	37.64	38.97
$\mu \pm \sigma$	37.48 ± 0.46	38.78 ± 0.17
ByT5-large baseline (5e-5)		
Seed 42	38.50	38.44
Seed 52	38.37	38.93
Seed 62	38.77	39.45
$\mu \pm \sigma$	38.55 ± 0.20	38.94 ± 0.50
ByT5-large + glosses (5e-5)		
Seed 42	37.54	37.55
Seed 52	38.23	39.10
Seed 62	37.52	37.86
$\mu \pm \sigma$	37.76 ± 0.40	38.17 ± 0.82

Table 7: Per-seed combined scores for the second-round multi-seed conditions. All rows come from the same environment as the multi-seed ByT5 results of Table 6.

H Per-Metric Model Scores

Table 8 reports BLEU and chrF++ alongside the combined score for the model family and size comparison of Section 5.2.

Model	BLEU	chrF++	Combined
TF-IDF 1-NN	12.20	27.02	18.16
<i>Shared recipe</i>			
mT5-small	6.93	19.32	11.58
mT5-base	9.52	24.69	15.33
ByT5-small	21.07	40.23	29.11
ByT5-base	27.51	48.32	36.46
<i>Per-configuration learning rate</i>			
mT5-small (1e-3)*	26.10	46.03	34.66
mT5-base (1e-3)*	26.73	47.08	35.47
mT5-base (3e-4)	27.50	47.87	36.28

Table 8: Model family $\times$ size results on the independent test set. ByT5 and per-configuration mT5 rows are means over seeds {42, 52, 62}; shared-recipe mT5 and TF-IDF rows are single runs. *Rate selected on validation from the widened grid {3e-4, 1e-3}.

I Interaction Robustness

The sign test in Section 4.6 treats the six seed-averaged differences as independent, which the shared baseline runs violate. Two checks that do not rely on that assumption give the same answer.

First, the interaction is stable at the level of individual seeds. Table 9 lists the per-seed interaction $\Delta_{\text{small},i} - \Delta_{\text{base},i}$ for every intervention. 17 of the 18 intervention-seed cells are positive, and the single exception (genre conditioning under seed 52) is -0.004 , indistinguishable from zero. Counting per seed, the interaction is positive for six of six interventions under seeds 42 and 62 and five of six under seed 52.

Second, leaving any one seed out and recomputing the seed-averaged differences from the remaining two preserves all six positive signs, so no single baseline run drives the direction. Magnitude is less evenly spread: dropping seed 62, whose baseline pair is the most extreme, leaves mean interactions of $+0.27$, $+0.49$, $+0.13$, $+0.35$, $+0.69$, and $+0.82$ (order as in Table 9), several below the per-intervention σ , so the size of the effect leans on one seed even though its sign does not.

Both checks operate at the level of training runs, where seed-to-seed variation enters; sentence-level uncertainty is quantified separately below.

Sentence-level bootstrap. Paired bootstrap resampling over the 951 independent-test sentences (Koehn, 2004) (2,000 resamples of the combined metric, seed-matched system pairs) gives 95% intervals for the study’s three headline deltas. The retuned family gap (ByT5-base at 1e-4 minus mT5-base at 1e-3) is $+1.50$ [$+0.35$, $+2.65$], $+1.72$ [$+0.55$, $+2.87$], and $+2.03$ [$+1.01$, $+3.05$]

for seeds 42, 52, and 62, excluding zero in every seed. The same mT5 run against the seed-42 shared-recipe environment-check baseline gives +0.40 [−0.74, +1.58], so against the under-tuned reference the families are statistically indistinguishable. The 100% versus 25% data gap at seed 42 is +16.97 [+15.77, +18.21]. The 1.2B gloss deltas are −0.96 [−1.73, −0.21], −0.14 [−0.91, +0.57], and −1.25 [−2.00, −0.48], excluding zero in two of three seeds.

Intervention	s42	s52	s62	Mean
Strip determinatives	+0.51	+0.03	+1.90	+0.81
Tag Sumerograms	+0.15	+0.82	+1.42	+0.80
Dictionary glosses	+0.07	+0.18	+2.68	+0.98
Genre conditioning	+0.69	−0.00	+2.14	+0.94
Numeral normalisation	+0.96	+0.41	+1.26	+0.87
Retrieval augmentation	+0.52	+1.11	+2.28	+1.30

Table 9: Per-seed interaction $\Delta_{\text{small},i} - \Delta_{\text{base},i}$ (combined score, independent test).

Table 10 reports the seed-averaged intervention deltas for BLEU and chrF++ separately on the independent test set. On each metric alone, all six interventions satisfy $\bar{\Delta}_{\text{small},i} > \bar{\Delta}_{\text{base},i}$.

Intervention	BLEU $\bar{\Delta}$		chrF++ $\bar{\Delta}$	
	Small	Base	Small	Base
Strip determinatives	+0.27	−0.43	+0.56	−0.37
Tag Sumerograms	−0.05	−0.83	+0.09	−0.66
Dictionary glosses	+0.29	−0.49	+0.96	−0.28
Genre conditioning	−0.21	−1.11	−0.19	−1.11
Numeral normalisation	+0.32	−0.47	+0.35	−0.56
Retrieval augmentation	+0.55	−0.46	+0.74	−0.93

Table 10: Seed-averaged intervention deltas per metric (independent test).

J From-Scratch Tokenisation Contrast

Both models share one T5 architecture, with four encoder and four decoder layers, hidden size 512, feed-forward size 2048, eight attention heads, and gated-GELU activations. The byte variant uses the 384-entry ByT5 vocabulary (38.0M parameters); the subword variant uses a 4,000-piece unigram SentencePiece vocabulary trained on the training-split sources and targets with full character coverage (39.8M parameters). Both train from random initialisation on the training split alone with batch size 32, learning rate 3e-4, up to 60 epochs, and early stopping on validation combined score with patience 10. Final evaluation uses beam search with 8 beams and length penalty 1.3, matching the

main protocol. Both variants share this one recipe, so the contrast carries the paper’s own caveat. It argues against an intrinsic byte advantage under a recipe chosen without reference to either variant, and it does not establish the fully tuned ranking. Table 11 reports per-seed scores.

Condition	Independent	New_test
<i>From-scratch, SentencePiece 4,000</i>		
per seed	18.77/18.13/19.33	19.04/18.25/18.04
$\mu \pm \sigma$	18.74 $\pm$ 0.60	18.44 $\pm$ 0.53
<i>From-scratch, bytes</i>		
per seed	13.76/13.88/11.99	14.51/14.99/12.18
$\mu \pm \sigma$	13.21 $\pm$ 1.06	13.89 $\pm$ 1.50

Table 11: From-scratch tokenisation contrast, combined score per seed {42, 52, 62}. The subword model is higher in every seed on both test sets.

K Translation Examples

In Table 12(a), mT5-base at the shared-recipe rate ignores the input and produces a high-frequency template, the failure mode that per-configuration tuning removes (Section 5.2). ByT5-base renders all four proper names. In (b), the base model already captures the correct quantity and modifier. The genre token degrades its output while helping the small model, which lacks that implicit knowledge.

L Environment Provenance

All multi-seed ByT5 results (the ByT5 rows of Table 8 and all of Tables 2 and 6 except the rows marked *earlier environment*) come from a single software and hardware environment, as do all second-round runs (the tuned mT5 conditions, the retuned interventions and baseline, the data-scaling grid, ByT5-large, and the from-scratch pair). Conditions peripheral to the interaction claim (first-round mT5, TF-IDF, external data, and the retrieval format variants) were run in an earlier environment. Fourteen seed-matched rerun pairs (baseline plus the six interventions, both model sizes, seed 42) measured a mean difference between the two environments of +0.04 combined points, with individual pairs ranging from −0.49 to +0.90. Individual cross-environment comparisons therefore carry an uncertainty comparable to the preprocessing effect sizes. For this reason every delta reported in this paper, covering intervention effects, data-scaling steps, and external-data deltas, is computed within a single environment. Cross-environment contrasts

(a) Shared recipe: untuned mT5 generates a template; ByT5 renders all proper names.	
Src	<i>IGI DUG-ì-lì IGI a-šur-SIPA IGI ší-lu-lu IGI (d)en-LÍL-ba-ni</i>
Ref	Witnesses: Ṭāb-ilī, Aššur-rē’ī, Šilūlu, Enlil-bāni
mT5-base@5e-5 (6.6)	If he does not pay (on time), he will add 1/2 shekels per mina per month as interest.
ByT5-base (99.1)	Witnesses: Ṭāb-ilī, Aššur-rē’ī, Šilūlu, Enlil-bāni.

(b) Capacity × preprocessing: genre token helps small, hurts base.	
Src	<i>GÍN.TA a ma-na-im i ITU.KAM ší-ib-tám ú-ša-áb 1</i>
Ref	He will add 1 shekel per mina per month as interest .
ByT5-sm. (53.1)	[...] add 1/2 shekels per mina per month.
+genre (71.0)	[...] add 1/2 shekels per mina per month as interest .
ByT5-base (85.2)	[...] add 1 shekel per mina per month as interest .
+genre (53.1)	[...] add 1/2 shekels per mina per month.

Table 12: Translation examples; all outputs are shared-recipe runs. chrF++ in parentheses. In (b), the shared prefix “If he does not pay (on time), he will” is omitted; bold marks reference-matching elements.

appear only where the gap is an order of magnitude larger than this noise, namely the first-round mT5 and TF-IDF comparisons. The tuned mT5 comparison is same-environment.