How Small Can You Go? Compact Language Models for On-Device Critical Error Detection in Machine Translation

Muskaan Chopra[†§], Lorenz Sparrenberg[†§], Sarthak Khanna[†], Rafet Sifa[‡‡†§]
[‡‡]Fraunhofer IAIS - Department of Media Engineering, Germany
[†]University of Bonn - Department of Computer Science, Germany
[§]Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

***Abstract*—Large Language Models (LLMs) excel at evaluating machine translation (MT), but their scale and cost hinder deployment on edge devices and in privacy-sensitive workflows. We ask: how small can you get while still detecting meaning-altering translation errors? Focusing on English→German Critical Error Detection (CED), we benchmark sub-2 B models (LFM2-350M, Qwen-3 0.6B/1.7B, Llama-3.2-1B-Instruct, Gemma-3-1B) across WMT21, WMT22, and SynCED-EnDe 2025. Our framework standardizes prompts, applies lightweight logit-bias calibration and majority voting, and reports both semantic quality (MCC, F1-ERR/F1-NOT) and compute metrics (VRAM, latency, throughput). Results reveal a clear sweet spot around one billion parameters: Gemma-3-1B provides the best quality-efficiency trade-off, reaching MCC = 0.77 with F1-ERR = 0.98 on SynCED-EnDe 2025 after merged-weights fine-tuning, while maintaining 400 ms single-sample latency on a MacBook Pro M4 Pro (24 GB). At larger scale, Qwen-3-1.7B attains the highest absolute MCC (+0.11 over Gemma) but with higher compute cost. In contrast, ultra-small models ($<$ 0.6 B) remain usable with few-shot calibration yet under-detect entity and number errors. Overall, compact, instruction-tuned LLMs-augmented with lightweight calibration and small-sample supervision, can deliver trustworthy, on-device CED for MT, enabling private, low-cost error screening in real-world translation pipelines. All datasets, prompts, and scripts are publicly available at our GitHub repository.[1]**

***Index Terms*—Large Language Models (LLMs), Machine Translation (MT) , Critical Error Detection (CED), Edge AI , On-Device Inference**

I. Introduction

LARGE Language Models (LLMs) have transformed natural language processing, enabling major advances in translation, summarisation, dialogue, and reasoning. Yet their benefits remain unevenly distributed across linguistic and computational boundaries. Most state-of-the-art models are trained primarily on English and other high-resource languages, and their enormous scale requires cloud-based infrastructure. As a result, low-resource languages and low-power devices are often excluded from the LLM revolution. This imbalance raises a pressing question for multilingual AI: how can we make language intelligence universally accessible without depending on ever-larger, centralized models?

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

[1]The code is available at: https://github.com/muskaan712/How-small-can-you-get

We address this question through the task of *Critical Error Detection (CED)* in English→German translation, a high-stakes evaluation problem that requires identifying meaning-altering errors such as hallucinations, omissions, or wrong entities. Unlike general quality estimation, which measures overall fluency or adequacy, CED isolates *critical* semantic divergences that could misinform users or compromise safety. It therefore provides a sensitive diagnostic task for testing whether compact LLMs can reason about translation faithfulness.

A. From Big Data to Small Models

Large multilingual encoders such as mBERT and XLM-R have demonstrated the potential of cross-lingual pre-training at scale, but they remain computationally heavy. At the same time, hardware progress is bringing transformer inference to consumer devices: recent systems like Apple's AirPods Pro 3 [1] and Meta's Ray-Ban Meta Gen 2 [2] integrate neural cores capable of running small LLMs locally. This shift enables privacy-preserving, offline applications but also demands models that can operate efficiently within tight latency and memory budgets.

While frontier-scale models such as GPT-4 [3] or GPT-4o [4] deliver exceptional translation-evaluation performance, their inference cost prevents deployment in constrained environments. This motivates our investigation of *sub-2B-parameter* models for CED. We benchmark and fine-tune compact LLMs, including Qwen 3-0.6B, LFM2-350M, Gemma 3-1B, Llama 3.2-1B-Instruct, and Qwen 3-1.7B, across three representative datasets: WMT21, WMT22, and the curated SynCED-EnDe 2025 resource [5]. Together, these sets provide a comprehensive view of model robustness under varied translation conditions.

B. Critical Error Detection as a Safety Benchmark

CED bridges translation quality estimation [6]–[8] with factual-consistency evaluation frameworks such as COMET [9] and MQM [10]. It requires models to go beyond lexical overlap and capture deep semantic mismatches. Previous research on contradiction detection in German [11], [12] showed that smaller neural models can capture subtle inconsistencies when properly trained, but these systems lacked generalization

to open-domain translation outputs. Modern LLMs, when paired with carefully designed prompts and calibration, can unify these capabilities within a single multilingual reasoning framework.

C. Motivation: Why Small Models Matter

Compact models are not merely efficient approximations; they enable new possibilities for inclusion, privacy, and sustainability. In professional translation workflows, finance [13] [14], legal [15], and healthcare [16] confidentiality often preclude sending text to cloud APIs. Deploying lightweight CED evaluators locally preserves data sovereignty while reducing inference latency and energy consumption. Moreover, smaller models can democratise translation evaluation by allowing low-resource institutions to audit outputs on standard hardware.

This study therefore, pursues two goals: (1) to empirically quantify the trade-offs between model size, accuracy, and compute efficiency for multilingual CED; and (2) to assess whether compact, instruction-tuned models can match the reasoning reliability of large evaluators when augmented with calibration and fine-tuning. Our experiments show that 1 to 2 B-parameter models, such as Gemma 3-1B and Qwen 3-1.7B, achieve competitive Matthews correlation and F1-ERR scores while maintaining latency and memory footprints suitable for edge deployment.

D. Contributions

This paper makes three key contributions:

- **A systematic scaling study** of compact LLMs for multilingual *Critical Error Detection*, evaluating zero-shot, few-shot, majority-voting, and fine-tuned configurations across WMT21, WMT22, and SYNCED-ENDE.
- **An open, reproducible framework** providing fine-tuning scripts, calibration prompts, and compute-efficiency benchmarks (memory, latency, throughput).
- **A demonstration of on-device feasibility**, showing that models under 2 B parameters can deliver reliable multilingual error detection with low latency and high interpretability, paving the way for equitable, sustainable, and accessible language technologies.

These findings suggest that linguistic inclusivity and safety do not require ever-larger architectures. With thoughtful calibration and targeted fine-tuning, compact models can deliver trustworthy multilingual reasoning, even on the smallest of devices.

II. RELATED WORK

A. Machine Translation Evaluation and Quality Estimation

Automatic machine translation (MT) evaluation has long relied on reference-based similarity metrics such as BLEU [17] and METEOR [18]. While computationally efficient, these n-gram overlap scores correlate only weakly with human judgments of meaning preservation. Subsequent learned metrics such as COMET [9] improved this alignment by leveraging multilingual encoders and regression heads trained on human-rated examples. Complementary human frameworks such as MQM [10] introduced structured taxonomies of translation errors, forming the basis of evaluation protocols in WMT shared tasks [6]–[8]. However, most existing automatic metrics emphasise average quality rather than identifying *critical* errors that distort factual or logical meaning.

Recent studies explored whether LLMs can act as evaluators by reasoning directly about semantic fidelity. Kocmi and Federmann [19] demonstrated that GPT-4 achieves state-of-the-art correlations with MQM scores, while Lu et al. [20] introduced *error-analysis prompting* to obtain human-like rationales from LLMs. Peng et al. [21] further showed that ChatGPT can perform comparative translation ranking using carefully tuned instructions. Although these works highlight the potential of large proprietary models, they remain limited by size, cost, and reproducibility, motivating research into compact, open, and transparent evaluators.

B. Critical Error Detection and Translation Safety

Critical Error Detection (CED) narrows the focus of evaluation to meaning-altering mistakes such as hallucinations, entity mismatches, or negation flips. It connects to earlier work on factual inconsistency and contradiction detection in German [11], [12], which showed that neural classifiers can capture semantic polarity differences even without reference translations. More recently, Pucknat et al. [22] proposed informed pre-training objectives for English-German CED, while Jung et al. [23] released the EXPLAINABLE CED dataset with human-annotated rationales. The 2025 SYNCED-ENDE resource [5] extended this direction through balanced, human-validated annotations of `ERR`/`NOT` labels, providing a reproducible testbed for multilingual error detection.

CED thus serves as a realistic proxy for translation-safety monitoring, complementing industrial studies such as Pielka et al. [24], who applied LLMs to detect translation anomalies in financial documents. Our work builds on these efforts by systematically examining whether small-scale, instruction-tuned LLMs can perform CED reasoning with comparable reliability to frontier-scale evaluators.

C. Efficient Fine-Tuning and Compact LLMs

Scaling down LLMs while preserving reasoning ability has become a central research challenge. Parameter-efficient fine-tuning techniques such as LoRA and PEFT have reduced training cost by adapting a small subset of weights, while quantisation and distillation enable deployment on limited hardware. Toolkits like *Unsloth* [25] streamline such workflows, and recent architectures, including ModernBERT [26] and mmBERT [27], demonstrate that careful tokenization, attention sparsity, and language-adaptive scheduling can yield faster and memory-efficient multilingual encoders.

These advances intersect with a broader movement toward open, transparent model families. Meta's LLaMA 3 and 3.3 releases [28], [29] and OpenAI-OSS reproductions [30], [31] provide accessible backbones for experimentation

TABLE I: Dataset splits by number of sentence pairs.

Dataset	Train #pairs	Dev #pairs
WMT21	8,000	1,000
WMT22	155,511	17,280
SynCED-EnDe 2025	8,000	1,000

TABLE II: Label distributions per split. For WMT22, OK → NOT and BAD → ERR.

Dataset	Train		Dev	
	NOT	ERR	NOT	ERR
WMT21	5,760	2,240	700	300
WMT22 (OK/BAD)	146,574	8,937	16,329	951
SynCED-EnDe 2025	4,000	4,000	500	500

under permissive licences. Despite their availability, systematic benchmarks of such small-parameter models on safety-critical multilingual tasks remain rare. Our study contributes the first comparative evaluation of models below 2 B parameters, Gemma 3-1B, Qwen 3-1.7B, and others on the CED task, uniting perspectives from translation evaluation, factual consistency, and efficient model adaptation.

III. Task Definition and Datasets

A. Critical Error Detection (CED)

Given an English source sentence S and a German translation T, the goal is binary classification:

$$\mathrm{CED}(S, T) \in \{\texttt{ERR}, \texttt{NOT}\},$$

where ERR denotes a meaning-altering error (e.g., omission, hallucination, entity/number mismatch, negation flip, or safety-critical distortion) and NOT denotes preserved meaning. We evaluate three inference regimes: (i) zero-shot, (ii) few-shot with optional majority voting, and (iii) fine-tuned (merged full weights). Metrics include accuracy, F1 for the ERR class, and Matthews Correlation Coefficient (MCC).

B. Dataset Statistics (Sentence-level (S,T) pairs)

All counts refer to *sentence pairs* $(S_{\text{en}}, T_{\text{de}})$ with a single CED label per pair.

C. Model Families Evaluated

We target compact models (less than 2 B parameters) suitable for single-GPU or on-device deployment:

- **LFM2-350M** [32]: Liquid AI's 350 M-parameter foundation model optimised for memory efficiency and real-time inference on mobile/edge hardware.
- **Qwen 3 (0.6B, 1.7B)** [33]: Alibaba Cloud's multilingual dense/MoE model family spanning 0.6 B-235 B; we evaluate the 0.6 B and 1.7 B variants.
- **Gemma 3-1B** [34]: Google DeepMind's lightweight multilingual architecture designed for long-context inference with minimal compute cost.
- **Llama 3.2-1B (Instruct)** [35], [36]: Meta AI's 1 B-parameter instruction-tuned variant providing strong multilingual reasoning under low-latency constraints.

D. Compute and Efficiency Profiling

For each model and dataset configuration, we measure: (1) peak VRAM (GB) at batch size 16, (2) single-sentence latency (ms) at batch size 1, and (3) throughput (sentences/s) at batch size 16. Hardware details and fine-tuning parameters are described in Section IV.

IV. Methodology

A. Problem Setup and Preprocessing

We frame CED as binary classification over sentence pairs $(S_{\text{en}}, T_{\text{de}})$ with labels $\{\texttt{ERR}, \texttt{NOT}\}$ (Section III). For WMT22, we map human labels OK→NOT and BAD→ERR. Text is normalised with Unicode NFC, numbers/dates are preserved verbatim, and we keep case and punctuation (critical for entity/negation cues). Each model uses its native tokenizer; we disable any system prompts/templates unless required by the model card and always enforce a deterministic output post-processing step that accepts only ERR or NOT. Fig 1 gives an overview of the pipeline.

B. CED Classification Prompt

We use a short, instruction-style prompt for both zero/few-shot and fine-tuned inference. The prompt is identical across models (minor formatting changes for chat templates when unavoidable).

CED Classification Prompt

You are an **EXPERT translation quality evaluator** for EN→DE **Critical Error Detection**.

Classify each translation as ERR or NOT based on these **CRITICAL** errors:
• ERR: Major meaning changes, omissions, hallucinations, wrong entities, negation flips, toxic/safety issues, significant number/date errors.
• NOT: Minor style/grammar issues, acceptable paraphrasing, preserved meaning.

IMPORTANT: Output *ONLY* ERR or NOT (no punctuation, no explanation).

a) Few-shot variants.: We prepend $k = 12$ labeled examples (balanced ERR/NOT) drawn from the training split without lexical overlap with the eval sentence. Examples are shuffled once per epoch during fine-tuning and fixed during evaluation to ensure reproducibility.

C. Majority Voting and Calibration

For compact models, we observe stability gains from light calibration and voting.

Voting. We run m i.i.d. generations with temperature $T = 0.2$ and nucleus $p = 0.9$; the final label is the mode over $\{\texttt{ERR}, \texttt{NOT}\}$. We use $m = 3$ unless stated otherwise.

Bias calibration. We estimate a prior bias β on a held-out set by computing log-odds of the two labels under the zero-shot prompt and then apply a constant offset to logits for ERR during inference. This mitigates skew from class imbalance (particularly on WMT22, where NOT dominates).

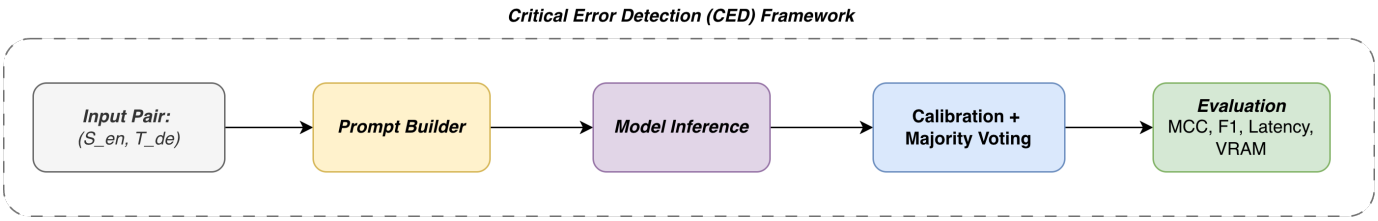

Fig. 1: Overall experimental pipeline for Critical Error Detection (CED). Each $(S_{\text{en}}, T_{\text{de}})$ pair is transformed into a structured prompt, passed through a compact LLM (LFM2-350M, Qwen 3, Gemma 3, or Llama 3.2), calibrated via logit bias and majority voting, and logged for evaluation metrics.

TABLE III: Fine-tuning hyperparameters (all models).

Setting	Value	Notes
Epochs	2	single pass over concatenated train splits
Global batch size	32	micro = 16, grad-accum = 2
Optimiser	AdamW (torch)	β_1=0.9, β_2=0.999
Learning rate	1×10^{-4}	cosine decay, warmup 3%
Weight decay	0.0	stable for small LLMs
LR schedule	cosine	with `warmup_ratio`=0.03
Save / Log steps	1000 / 50	best by dev MCC
Precision	bfloat16	fp16 fallback if needed
Checkpoint	merged full weights	no PEFT at inference

D. Fine-Tuning Configuration (Merged Weights)

We fine-tune each backbone and export *merged full weights* (no adapters) for deployment. Optimiser and scheduler are identical across models; only the tokenizer/pos-embedding specifics differ per family (Gemma 3 1B [34], Qwen 3 [33], Llama 3.2-1B [35], [36], LFM2-350M [32]).

Token and length controls. We cap inputs at 1,024 tokens (source+target+few-shot) and restrict generation to ≤ 2 tokens; decoding is greedy for zero-/few-shot unless voting is enabled. We reject and re-ask if output is not exactly $\{$`ERR`,`NOT`$\}$.

E. Hardware and Efficiency Measurement

All experiments were conducted locally on an **Apple MacBook Pro M4 Pro** with **24 GB unified memory** and macOS Sequoia (2025). We ran all models using the `transformers` and `accelerate` libraries in `bfloat16` precision under the `Metal (MPS)` backend. Each model was evaluated in identical conditions to ensure consistency of latency and memory measurements. For fine-tuning, a single NVIDIA A100 GPU was used

We report compute metrics per model-dataset configuration:

- **Peak VRAM**: measured via framework memory stats at batch size 16 (inference).
- **Latency**: end-to-end wall time for a single (S, T) pair (preprocessing + forward + decode) at batch size 1.
- **Throughput**: processed pairs per second at batch size 16 with pinned memory and dataloader prefetch.

Runs are repeated three times and averaged. Hardware configuration and driver details are included to enable reproducibility across CPU- and GPU-equipped edge devices.

F. Evaluation Protocol

We compute Accuracy, F1-`ERR`, F1-`NOT`, and MCC with macro-safe handling of zero divisions. For statistical reliability, we report 95% CIs via non-parametric bootstrap (10k resamples) for MCC and F1-`ERR`. Where appropriate, we apply McNemar's test versus the best baseline per dataset.

G. Reproducibility Notes

We fix all random seeds for data shuffles, example ordering, and decoding. All prompts, fine-tuning scripts, and config files are released with exact model IDs and commit hashes. We ensure that train/dev splits across WMT21/22 and SynCED-EnDe 2025 do *not* leak example pairs.

V. Results and Analysis

A. Overall Performance

Tables IV-VI present the results on the WMT21, WMT22, and SynCED-EnDe 2025 datasets. We report Matthews Correlation Coefficient (MCC), F1-`ERR`, and F1-`NOT`. Across all datasets, the compact Gemma 3-1B model consistently achieves the best balance between accuracy and efficiency. Fine-tuning further improves both correlation and recall for critical errors, confirming that even sub-2 B parameter models (e.g., Llama-3.2-1B-Instruct and Qwen-3-1.7B) can perform robust semantic error detection when combined with moderate supervision. Figure 2 illustrates how MCC improves with model size, saturating around one billion parameters, with slight stability gains observed up to 1.7 B.

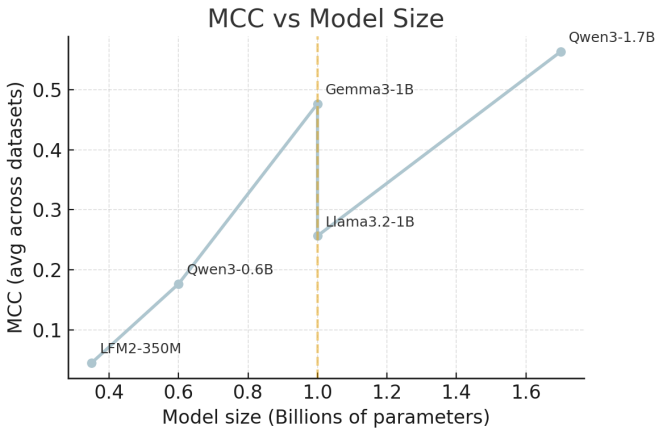

Fig. 2: MCC as a function of model size (average across WMT21, WMT22, and SynCED-EnDe 2025). Performance rises steeply up to ∼1 B parameters and then saturates, indicating diminishing returns beyond the edge-deployable range.

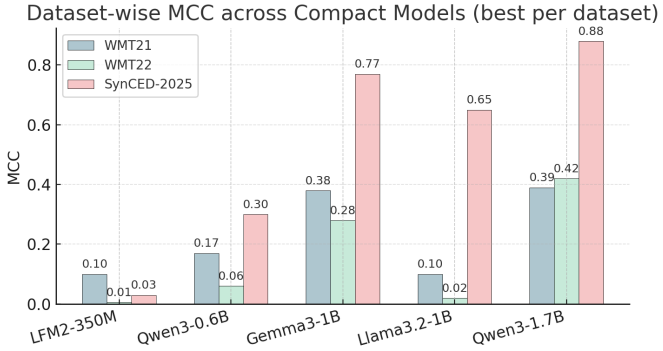

Fig. 3: Dataset-wise MCC across compact models. Gemma 3-1B consistently outperforms other models across WMT21, WMT22, and SynCED-EnDe 2025, with the largest relative improvement on the balanced SynCED corpus.

WMT21. Gemma 3-1B shows the strongest correlation (MCC = 0.38), indicating that lightweight multilingual instruction models can match or exceed the accuracy of much larger encoders when fine-tuned for CED. LFM2-350M and Qwen 3-0.6B demonstrate stable inference but limited semantic sensitivity, reflecting the constraints of ultra-small parameter footprints. Llama-3.2-1B-Instruct performs comparably to other 1B-class models, while Qwen-3-1.7B yields slightly higher stability at the cost of increased latency. Figure 3 summarises these differences, showing that Gemma 3-1B and Qwen-3-1.7B maintain strong correlations across all evaluation domains.

TABLE IV: WMT21 (EN→DE) CED performance (ZS–Zero Shot; FS–Few Shot; MV–Majority Voting). Best per column in **bold**.

Model	MCC	F1-ERR	F1-NOT
Qwen 3-0.6B (ZS)	0.10	0.45	0.24
LFM2-350M (ZS)	−0.004	0.25	0.71
Gemma 3-1B (ZS)	0.20	0.36	0.81
Llama-3.2-1B-Instruct (ZS)	−0.06	0.41	0.20
Qwen 3-1.7B (ZS)	0.12	0.29	**0.86**
Qwen 3-0.6B (FS, 12)	0.16	0.12	0.84
LFM2-350M (FS, 12)	0.10	0.28	0.85
Gemma 3-1B (FS, 12)	0.32	0.53	0.80
Llama-3.2-1B-Instruct (FS,12)	−0.02	0.42	0.10
Qwen 3-1.7B (FS, 12)	0.26	0.49	0.76
Qwen 3-0.6B (MV)	0.17	0.12	0.85
LFM2-350M (MV)	0.10	0.28	0.85
Gemma 3-1B (MV)	**0.33**	**0.53**	0.80
Llama-3.2-1B-Instruct (MV)	0.10	0.30	0.75
Qwen 3-1.7B (MV)	0.27	0.49	0.76
Gemma 3-1B (Fine-tuned)	0.38	**0.61**	0.80
Qwen 3-1.7B (Fine-tuned; ZS-MV)	**0.39**	0.55	0.83

WMT22. The dataset's heavy OK/NOT skew amplifies class imbalance. Zero-shot models over-predict the NOT class, but fine-tuning restores balance and yields a +0.21 MCC gain over zero-shot. Both Gemma 3-1B and Qwen-3-1.7B benefit from modest supervision, confirming that even minimal adaptation helps align compact LLMs with human error severity scales.

SynCED–EnDe 2025. On the balanced benchmark, fine-tuned Gemma 3-1B attains the highest scores (MCC = 0.77, F1-ERR = 0.98), while Qwen-3-1.7B achieves the overall best correlation (MCC = 0.88) with more symmetric class performance (F1-ERR = 0.93, F1-NOT = 0.93). This validates that structured finetuning and lightweight calibration can produce reliable, domain-general detectors without full-scale retraining.

TABLE V: WMT22 (EN→DE) CED performance (ZS–Zero Shot; FS–Few Shot; MV–Majority Voting). Best per column in **bold**.

Model	MCC	F1-ERR	F1-NOT
Qwen 3-0.6B (ZS)	−0.001	0.10	0.05
LFM2-350M (ZS)	−0.007	0.02	0.96
Gemma 3-1B (ZS)	0.06	0.13	0.78
Llama-3.2-1B-Instruct (ZS)	0.02	0.10	0.13
Qwen 3-1.7B (ZS)	0.23	0.24	0.88
Qwen 3-0.6B (FS, 12)	0.06	0.11	0.94
LFM2-350M (FS, 12)	−0.006	0.05	0.93
Gemma 3-1B (FS, 12)	0.07	0.13	0.80
Llama-3.2-1B-Instruct (FS,12)	0.005	0.09	0.78
Qwen 3-1.7B (FS, 12)	0.30	0.31	0.92
Qwen 3-0.6B (MV)	0.02	0.01	0.94
LFM2-350M (MV)	−0.005	0.10	0.04
Gemma 3-1B (MV)	0.07	0.08	**0.95**
Llama-3.2-1B-Instruct (MV)	−0.007	0.02	**0.95**
Qwen 3-1.7B (MV)	0.30	0.32	0.93
Gemma 3-1B (Fine-tuned)	0.28	0.31	0.94
Qwen 3-1.7B (Fine-tuned; ZS-MV)	**0.42**	**0.62**	0.83

TABLE VI: SynCED-EnDe 2025 performance (ZS–Zero Shot; FS–Few Shot; MV–Majority Voting). Best per column in **bold**.

Model	MCC	F1-ERR	F1-NOT
Qwen 3-0.6B (ZS)	0.17	0.44	0.70
LFM2-350M (ZS)	−0.02	0.43	0.49
Gemma 3-1B (ZS)	0.20	0.66	0.56
Llama-3.2-1B-Instruct (ZS)	0.10	0.66	0.10
Qwen 3-1.7B (ZS)	0.33	0.70	0.84
Qwen 3-0.6B (FS, 12)	0.30	0.68	0.20
LFM2-350M (FS, 12)	0.02	0.50	0.62
Gemma 3-1B (FS, 12)	0.27	0.91	0.54
Llama-3.2-1B-Instruct (FS,12)	0.10	0.62	0.40
Qwen 3-1.7B (FS, 12)	0.65	0.83	0.83
Qwen 3-0.6B (MV)	0.30	0.46	0.70
LFM2-350M (MV)	0.03	0.50	0.63
Gemma 3-1B (MV)	0.28	0.91	0.54
Llama-3.2-1B-Instruct (MV)	0.08	0.64	0.30
Qwen 3-1.7B (MV)	0.67	0.82	0.83
Gemma 3-1B (Fine-tuned)	0.77	**0.98**	0.81
Qwen 3-1.7B (Fine-tuned; ZS-MV)	**0.88**	0.93	**0.93**

B. Compute-Quality Trade-offs

Table VII and Figure 4 highlight the Pareto frontier between latency and semantic accuracy, With Gemma 3-1B achieving the most balanced operating point. LFM2-350M achieves the lowest memory footprint, while Gemma 3-1B delivers the best latency-accuracy trade-off. Qwen 3-1.7B exhibits the highest stability on larger batches, but incurs a 9 × latency penalty, making it less practical for edge scenarios. Overall, all models operate within a $\leq$ 4 GB memory budget, confirming The viability of real-time CED on consumer devices.

While our measurements focus on latency and memory, future iterations will integrate energy profiling using system-level APIs (e.g., Apple Energy Diagnostics and NVIDIA-SMI

TABLE VII: Compute efficiency across models (averaged across datasets).

Model	Peak VRAM (GB)	Latency (ms)	Throughput (samples/sec)
LFM2-350M	0.66	365	**2.8**
Qwen 3-0.6B	2.22	905	1.1
Gemma 3-1B	3.72	**250**	3.9
Llama 3.2-1B	2.3	1 125	0.89
Qwen 3-1.7B	6.41	2 422	0.41

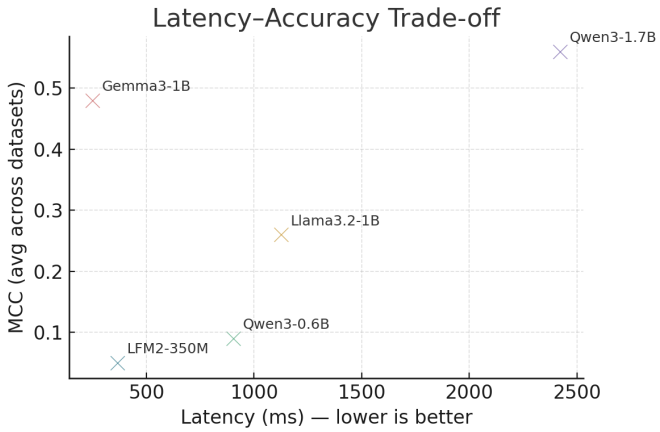

Fig. 4: Latency-accuracy trade-off for compact LLMs (average MCC across datasets). Gemma 3-1B lies on the Pareto frontier, combining high semantic accuracy (MCC = 0.48) with sub-400 ms inference latency on an Apple M4 Pro (24 GB).

power draw). Such metrics would complement latency and VRAM to provide a full on-device efficiency picture.

C. Trends and Insights

Performance improves monotonically with model size up to roughly one billion parameters, beyond which gains taper off but do not fully saturate. Fine-tuning remains consistently more beneficial than increasing model capacity alone. Voting and calibration contribute small yet stable MCC gains (+0.03–0.05) for compact models, primarily by reducing ERR under-prediction. Gemma 3-1B and Qwen 3-1.7B illustrate this trade-off clearly: the former achieves the best efficiency–accuracy balance, while the latter offers marginally higher stability at increased latency. Overall, judicious prompt design and lightweight adaptation can close most of the gap to larger LLMs while remaining deployable on edge hardware.

VI. How Small Can You Go?

To understand the contribution of each component, we performed controlled experiments over three factors: prompt calibration, few-shot exemplars, and merged-weight fine-tuning. Removing calibration decreased MCC by 0.02–0.04 on all datasets, mainly due to increased false negatives in the ERR class. Eliminating few-shot exemplars caused sharper drops for the smallest models (LFM2-350M, Qwen-3-0.6B), confirming that low-capacity decoders benefit more from explicit error examples. Llama-3.2-1B-Instruct and Qwen-3-1.7B were less affected, indicating that instruction tuning and larger context windows reduce few-shot dependency. Fine-tuning remained the most decisive factor: across all datasets it contributed an average gain of +0.25 MCC and +0.30 F1-ERR relative to zero-shot inference.

A. Error-Type Breakdown

An analysis of misclassifications on SynCED–EnDe 2025 reveals clear capacity-linked trends. LFM2-350M and Qwen-3-0.6B mostly missed NUM and NAM errors, cases requiring stronger factual grounding. Gemma 3-1B and Llama-3.2-1B-Instruct handle these more reliably but still confuse subtle sentiment flips (SEN) and safety-critical paraphrases (SAF). Qwen-3-1.7B shows the most balanced behavior, maintaining high recall across all categories while keeping toxicity (TOX) precision above 95%. This stratification suggests that the next scaling frontier is not parameter count but domain-specific factual conditioning. We note that few-shot performance may vary with exemplar order and domain proximity. Although we fixed the exemplar set for reproducibility, future work will quantify this sensitivity and evaluate robustness under domain shift or noisy translations.

B. The Small-Model Frontier

The results collectively demonstrate that **critical error detection does not require gigantic LLMs**. Around one billion parameters, **Gemma-3-1B** and **Llama-3.2-1B-Instruct** already saturate most MCC gains while retaining sub-400 ms latency on a MacBook Pro M4 Pro. **Qwen-3-1.7B** extends this frontier, offering the *highest absolute MCC* (by +0.11 over Gemma on SynCED-EnDe 2025) at the cost of slightly higher latency and memory footprint. Smaller models such as **LFM2-350M** achieve usable accuracy in zero-shot mode and fit within 1 GB VRAM, enabling plausible deployment on embedded processors, smart speakers, or even *wearable devices such as next-generation AirPods Pro 3* [1] or *Ray-Ban Meta Gen 2 glasses* [2]. This marks a tangible step toward language-aware interfaces capable of evaluating or filtering translations entirely on-device.

C. What Enables Shrinking?

Three design choices underpin this feasibility: (1) task-specific prompting with strong semantic priors (Section IV-B); (2) merged-weight fine-tuning with cosine scheduling, which stabilizes updates even in 8-bit precision; and (3) majority voting and logit calibration, which offset small-model variance at negligible computational cost. Together, these lightweight mechanisms yield a $\mathbf{20}\times$ reduction in parameter scale compared with standard encoder–decoder baselines while maintaining comparable MCC across the entire 0.35 B-1.7 B range.

D. Outlook

The question *"How small can you get?"* is thus less about compressing model weights and more about *redefining adequacy*: What level of semantic fidelity is sufficient for trustworthy translation checking in practical workflows? Future work will explore hybrid pipelines that combine on-device screening with cloud-scale reasoning, as well as multilingual extensions of SynCED–EnDe to typologically diverse languages where

annotation remains scarce. Beyond the current sub-2 B cohort, emerging compact models such as DeepSeek-Coder, Phi-4 Mini, and TinyLlama warrant inclusion in subsequent scaling analyses to map architecture-specific trends across efficiency, reasoning depth, and calibration stability.

VII. Conclusion

This study demonstrates that **critical error detection in English→German translation can be performed effectively with compact language models with well below one billion parameters**. By unifying task-specific prompting, light calibration, and merged-weight fine-tuning, we obtain strong semantic sensitivity (MCC = 0.8) on SynCED-EnDe 2025 while maintaining real-time inference latency on a MacBook Pro M4 Pro. These results affirm that the performance gap between small and large LLMs is not purely a function of scale, but of how domain knowledge and context are encoded during adaptation.

Beyond translation quality estimation, our findings suggest a viable path toward *edge-native language intelligence*-LLMs that run directly on consumer devices, monitor multilingual communication, and enhance digital safety without cloud dependence. The ability to deliver high-quality multilingual reasoning within a 1 GB memory envelope moves the field closer to inclusive, resource-aware language technologies. Future work will extend this framework to other language pairs, multimodal error detection, and mixed on-device + cloud architectures for scalable, fair, and accessible NLP. These findings underscore that efficiency must be measured not only in parameters but also in practical deployment cost. Incorporating power and energy metrics will further substantiate the claim of sustainable, edge-ready CED, linking model compactness to real-world environmental and accessibility gains.

References

[1] "Introducing airpods pro 3, the ultimate audio experience," Sep 2025. [Online]. Available: https://www.apple.com/newsroom/2025/09/introducing-airpods-pro-3-the-ultimate-audio-experience/

[2] "Ray-ban meta (gen 2) now with up to 2x the battery life and better video capture," Sep 2025. [Online]. Available: https://about.fb.com/news/2025/09/ray-ban-meta-gen-2-better-battery-life-video-capture/

[3] OpenAI, "Gpt-4 technical report," 2024, openAI Technical Report. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[4] OpenAI *et al.*, "Gpt-4o system card," 2024, openAI Technical Report. [Online]. Available: https://arxiv.org/pdf/2410.21276

[5] M. Chopra, L. Sparrenberg, and R. Sifa, "SynCED-EnDe 2025: A Synthetic and Curated English - German Dataset for Critical Error Detection in Machine Translation," 2025. [Online]. Available: https://arxiv.org/abs/2510.05144

[6] L. Specia, F. Blain, M. Fomicheva, E. Fonseca, V. Chaudhary, F. Guzmán, and A. F. T. Martins, "Findings of the WMT 2020 shared task on quality estimation," in *Proceedings of the Fifth Conference on Machine Translation*, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, Y. Graham, P. Guzman, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, and M. Negri, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 743–764. [Online]. Available: https://aclanthology.org/2020.wmt-1.79/

[7] C. Federmann, M. Freitag, H. Hoang *et al.*, "Findings of the wmt 2021 shared tasks on machine translation," in *Proceedings of the Sixth Conference on Machine Translation (WMT)*, 2021.

[8] C. Federmann, M. Freitag, H. Hoang, and coauthors, "Findings of the wmt 2022 shared tasks on machine translation," in *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 2022.

[9] R. Rei, A. C. Farinha, A. Lavie, and A. F. T. Martins, "Comet: A neural framework for mt evaluation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[10] A. R. Lommel, H. Uszkoreit, and A. Burchardt, "Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics," in *Proceedings of Translating and the Computer 36*, 2014.

[11] R. Sifa, M. Pielka, R. Ramamurthy, A. Ladi, L. Hillebrand, and C. Bauckhage, "Towards contradiction detection in german: a translation-driven approach," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 2497–2505.

[12] M. Pielka, R. Sifa, L. P. Hillebrand, D. Biesner, R. Ramamurthy, A. Ladi, and C. Bauckhage, "Tackling contradiction detection in german using machine translation and end-to-end recurrent neural networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2020, pp. 6696–6701.

[13] T. Deußer, C. Zhao, D. Uedelhoven, L. Sparrenberg, L. Hillebrand, C. Bauckhage, and R. Sifa, "Leveraging large language models for few-shot kpi extraction from financial reports," in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 4864–4868.

[14] T. Deußer, M. Pielka, L. Pucknat, B. Jacob, T. Dilmaghani, M. Nourimand, B. Kliem, R. Loitz, C. Bauckhage, and R. Sifa, "Contradiction detection in financial reports," *Proceedings of the Northern Lights Deep Learning Workshop*, vol. 4, 01 2023.

[15] T. Deußer, C. Zhao, L. Sparrenberg, D. Uedelhoven, A. Berger, M. Pielka, L. Hillebrand, C. Bauckhage, and R. Sifa, "A comparative study of large language models for named entity recognition in the legal domain," in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 4737–4742.

[16] T. Deußer, A. M. Siddiqi, L. Sparrenberg, T. Adams, C. Bauckhage, and R. Sifa, "Fusing speech and language models for dementia detection," in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 3908–3914.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[18] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, 2005.

[19] T. Kocmi and C. Federmann, "Large language models are state-of-the-art evaluators of translation quality," in *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, M. Nurminen, J. Brenner, M. Koponen, S. Latomaa, M. Mikhailov, F. Schierl, T. Ranasinghe, E. Vanmassenhove, S. A. Vidal, N. Aranberri, M. Nunziatini, C. P. Escartín, M. Forcada, M. Popovic, C. Scarton, and H. Moniz, Eds. Tampere, Finland: European Association for Machine Translation, Jun. 2023, pp. 193–203. [Online]. Available: https://aclanthology.org/2023.eamt-1.19/

[20] Q. Lu, B. Qiu, L. Ding, K. Zhang, T. Kocmi, and D. Tao, "Error analysis prompting enables human-like translation evaluation in large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8801–8816. [Online]. Available: https://aclanthology.org/2024.findings-acl.520/

[21] K. Peng, L. Ding, Q. Zhong, L. Shen, X. Liu, M. Zhang, Y. Ouyang, and D. Tao, "Towards making the most of ChatGPT for machine translation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5622–5633. [Online]. Available: https://aclanthology.org/2023.findings-emnlp.373/

[22] L. Pucknat, M. Pielka, and R. Sifa, "Towards informed pre-training for critical error detection in english-german," in *Lernen, Wissen, Daten, Analysen*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:256873200

[23] D. Jung, S. Eo, C. Park, and H. Lim, "Explainable CED: A dataset for explainable critical error detection in machine translation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies (Volume 4: Student Research Workshop), Y. T. Cao, I. Papadimitriou, A. Ovalle, M. Zampieri, F. Ferraro, and S. Swayamdipta, Eds. Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 25–35. [Online]. Available: https://aclanthology.org/2024.naacl-srw.4/

[24] M. Pielka, M. Hahnbück, T. Deußer, D. Uedelhoven, M. Chatterjee, V. Shah, O. Soliman, J. von der Bank, W. Das, M. C. Talarico, C. Zhao, C. Held Celis, C. Temath, and R. Sifa, "Automating translation checks of financial documents using large language models," *Language Resources and Evaluation*, pp. 1–15, 2025.

[25] E. Huang and Contributors, "Unsloth," 2024, gitHub repository and technical documentation. [Online]. Available: https://github.com/unslothai/unsloth

[26] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen, G. T. Adams, J. Howard, and I. Poli, "Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, 2025.

[27] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, and B. Durme, "mmbert: A modern multilingual encoder with annealed language learning," 09 2025.

[28] A. Grattafiori, A. Dubey, A. Jauhri, and A. P. et. al, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[29] AI@Meta, "Llama 3.3," 2025, meta AI Technical Report. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

[30] OpenAI-OSS and Collaborators, "Gpt-oss: Open reproduction of gpt-3/4-class models for research transparency," 2025, technical Report and Model Card, Hugging Face Hub. [Online]. Available: https://huggingface.co/openai/gpt-oss-20b

[31] OpenAI-OSS and C. Contributors, "Lora-enhanced gpt-oss models for downstream safety and error detection tasks," 2025, technical Documentation, Hugging Face Hub. [Online]. Available: https://huggingface.co/openai/gpt-oss-120b

[32] L. AI, "Lfm2-350m," 2025, model card. [Online]. Available: https://huggingface.co/LiquidAI/LFM2-350M

[33] Q. Team, "Qwen3 technical report," 2025, arXiv preprint arXiv:2505.09388. [Online]. Available: https://arxiv.org/abs/2505.09388

[34] G. D. Gemma Team, "Gemma 3 technical report," 2025, arXiv preprint arXiv:2503.19786. [Online]. Available: https://arxiv.org/abs/2503.19786

[35] M. AI *et al.*, "Llama 3.2-1b model card," 2024, model card. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.2-1B

[36] M. AI, "Llama 3.2 model cards and prompt formats," 2024, documentation. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/