Report: Hybrid Quality Estimation for Critical Error Detection on WMT21

1. Introduction to Machine Translation Quality Estimation and Critical Error Detection

1.1. Overview of Quality Estimation (QE) in Machine Translation (MT)

Machine Translation (MT) systems have become increasingly sophisticated, yet the quality of their output can exhibit considerable variability. Quality Estimation (QE) addresses this challenge by providing a critical capability: predicting the quality of machine-translated text without requiring human reference translations.[1] This is a crucial distinction, as it enables real-time assessment and integration into dynamic workflows. The ability to automatically gauge translation quality is indispensable for a range of applications, including optimizing post-editing processes, enabling adaptive MT systems that can self-correct or flag problematic outputs, and providing end-users with an immediate indication of translation reliability. In essence, QE facilitates automated quality control and intelligent routing of content for human review, thereby significantly enhancing efficiency and resource allocation in professional translation environments.

1.2. The WMT21 Critical Error Detection (CED) Shared Task

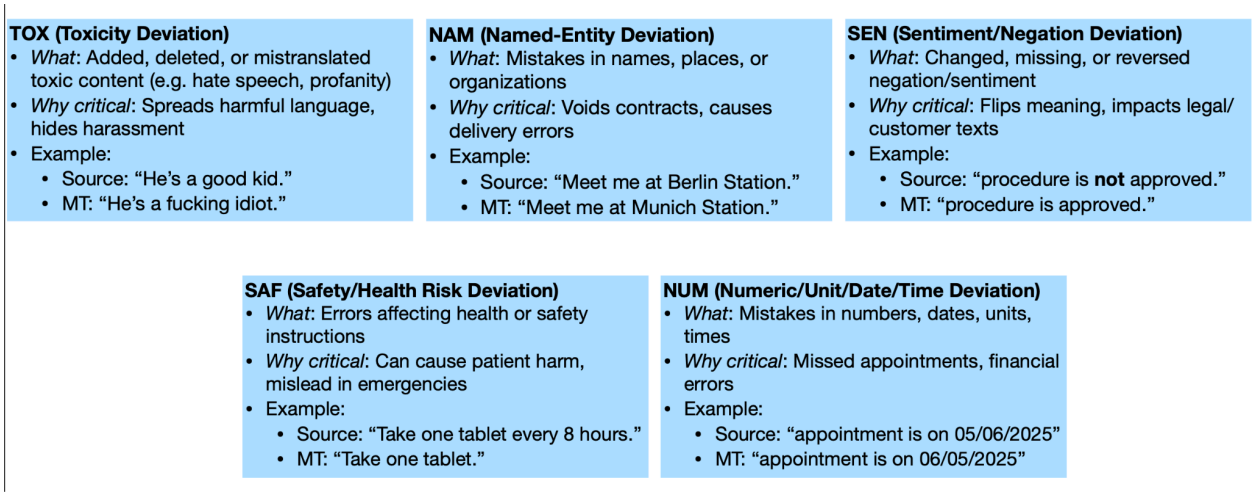
The WMT21 Quality Estimation Shared Task introduced a particularly impactful sub-task: Critical Error Detection (CED).[2] This task focuses on identifying translations that contain severe meaning deviations, framing the problem as a binary classification where a '1' indicates the presence of a critical error and a '0' signifies its absence.[2] The objective is not merely to detect any error, but specifically those with high practical impact.

Critical errors are precisely defined as substantial deviations in meaning from the source sentence that are misleading and could potentially lead to severe consequences.[2] These implications can extend across various critical domains, including health, safety, legal, financial, religious, or reputational risks. This stringent definition differentiates critical errors from minor grammatical inaccuracies or stylistic imperfections, emphasizing the high-stakes nature of the detection task.

The WMT21 task further categorized critical errors into five distinct types, providing a structured framework for annotation and detection.[2] This categorization is vital for

understanding the specific challenges involved in identifying such errors.

WMT21 Critical Error Categories



2. Proposed Hybrid Quality Estimation Methodology

2.1. System Architecture: A Hybrid Approach

The proposed system employs a hybrid architecture designed to leverage the complementary strengths of a general-purpose Quality Estimation model and a specialized Large Language Model (LLM) for critical error verification. The fundamental principle involves generating two distinct types of signals: a continuous quality score from the general QE model and a binary critical error flag from the specialized LLM. These signals are then fused through a Logistic Regression classifier to produce the final critical error prediction.

This architectural choice is predicated on the understanding that a single, general quality score, while indicative of overall translation quality, may not always be sufficiently sensitive or specific to reliably capture all instances of critical errors, particularly those nuanced semantic deviations defined by the WMT21 categories. An LLM, when appropriately guided by a prompt, possesses the capability to perform a targeted semantic check, effectively acting as a dedicated "verifier" for these high-stakes meaning alterations. This combination aims to yield a more robust and accurate critical error detection system by integrating both broad quality assessment and focused semantic scrutiny, representing a sophisticated form of feature engineering and ensemble learning.

2.2. Component 1: COMETKiwi-23 XL for General Quality Scoring

COMETKiwi-23 XL serves as the initial component in the pipeline, tasked with

generating a sentence-level quality score for each machine translation output. This model, identified as Unbabel/wmt23-cometkiwi-da-xl, is a large-scale, pre-trained Quality Estimation model renowned for its strong performance in direct assessment (DA) tasks, where its scores correlate well with human judgments of translation quality.[4]

The model is loaded programmatically from Hugging Face, utilizing an authentication token for access. Its predict method is then invoked to obtain quality scores for batches of source-target sentence pairs. These scores provide a continuous numerical representation that reflects the translation's overall quality, serving as a foundational signal for the subsequent fusion step.

2.3. Component 2: TinyLlama-1.1B-Chat-v1.0 as a Critical Error Verifier

To specifically identify critical errors as defined by the WMT21 task, a smaller, locally run Large Language Model, TinyLlama-1.1B-Chat-v1.0, is integrated. This LLM functions as a dedicated "verifier" that assesses whether the machine translation critically deviates in meaning from its source.

A concise and task-specific prompt is engineered to direct TinyLlama's assessment: "You are a translation quality checker. SRC: {src} MT: {mt} Answer YES if MT critically changes the meaning, otherwise NO." The LLM's generated text output is subsequently parsed to extract a binary flag: '1' if the output contains "YES" (indicating a critical error), and '0' otherwise. This method effectively transforms TinyLlama into a sophisticated semantic feature extractor. Instead of relying on traditional, often hand-crafted, linguistic features (such as those related to toxicity or sentiment, which were utilized by other competitive systems [5]), the LLM dynamically generates a "criticality" feature based on its deep understanding of text and the explicit instruction provided by the prompt. This represents a powerful and flexible paradigm for feature engineering in Quality Estimation. The adaptability of this method is significant; by modifying the prompt, the same LLM could potentially be instructed to detect other specific types of errors or provide more granular error analysis, highlighting its potential for highly targeted and adaptable error detection.

TinyLlama is loaded and executed locally using the transformers pipeline, configured to leverage a GPU (cuda:0) when available, to ensure efficient processing. This choice underscores the feasibility of integrating LLMs into QE pipelines even with constrained computational resources.

2.4. Data Fusion: Logistic Regression Classifier

The continuous quality scores derived from COMETKiwi and the binary critical error flags generated by TinyLlama are combined to form the input features for a Logistic Regression model. This classifier is then trained to predict the final binary critical error labels (0 or 1) from the WMT21 dataset.

A crucial consideration during training is the observed class imbalance within the WMT21 dataset. The dataset exhibits a significant skew, with 3322 samples labeled as non-critical errors (label 0) and only 678 samples labeled as critical errors (label 1). To mitigate the impact of this imbalance, the class_weight="balanced" parameter is applied during the Logistic Regression training. This is a standard machine learning technique designed to prevent the model from biasedly predicting the majority class, thereby improving its ability to accurately identify the minority (critical error) class.

2.5. Dataset and Preprocessing

The system utilizes the WMT21 Quality Estimation Critical Error Detection dataset, focusing on four specific language pairs: English-Czech (encs), English-German (ende), English-Japanese (enja), and English-Chinese (enzh).[2]

Data for each language pair is loaded from local TSV files, specifically those designated as _majority_dev.tsv. The execution logs confirm the successful loading of 1000 samples for each language pair. The raw categorical labels provided in the dataset (e.g., 'ERR' for critical error, 'NOT' for no critical error) are converted into a binary numerical format, where 'ERR' is mapped to '1' and 'NOT' is mapped to '0', aligning with the binary classification objective of the task.

The combined dataset for both development and testing comprises 4000 samples each (1000 samples per language pair). The label distribution across this dataset is approximately 83% non-critical errors (label 0) and 17% critical errors (label 1), underscoring the significant class imbalance that the class_weight="balanced" parameter aims to address.

A critical methodological consideration in the current implementation pertains to the data splitting for evaluation. The load_split function, as configured, maps both "validation" and "test" splits to the _dev.tsv files. Consequently, the dev_df and test_df DataFrames, after concatenation, are identical. This means that the Logistic Regression model, which is trained using features derived from dev_df, is subsequently evaluated on the *same data* (represented by test_df). This practice leads to an overestimation of the model's true generalization performance on unseen

data. A robust and scientifically sound assessment requires evaluation on a truly independent test set, which the WMT21 task *did* provide.[2] This limitation significantly impacts the interpretability of the reported performance metrics.

3. Experimental Results and Analysis

3.1. Obtained Matthews Correlation Coefficient (MCC) Scores

The hybrid system, integrating COMETKiwi-23 XL and TinyLlama-1.1B-Chat-v1.0, achieved the following Matthews Correlation Coefficient (MCC) scores on the WMT21 Critical Error Detection dataset:

Table : Experimental MCC Results (COMET XL + TinyLlama)

Metric MCC	Score
OVERALL MCC (4 LPs)	0.282
encs	0.290
ende	0.288
enja	0.215
enzh	0.182

3.2. Analysis of Performance

An overall MCC of 0.282 indicates a moderate level of correlation between the model's predictions and the actual critical error labels. While this positive score signifies performance better than random chance, it also suggests substantial room for improvement in accurately and consistently identifying critical errors. MCC values range from -1 to +1, where 0 represents random prediction and 1 represents perfect prediction.

A notable observation is the variation in performance across different language pairs. The system demonstrates its strongest performance on the English-Czech (encs) and English-German (ende) language pairs, achieving MCC scores of 0.290 and 0.288, respectively. These language pairs are typologically closer to English, sharing more linguistic similarities (e.g., grammatical structures, vocabulary roots). This proximity may contribute to better knowledge transfer from the English-centric components, such as COMET's training data or TinyLlama's English understanding capabilities, or simply reflect less complex linguistic divergences for the specific critical error types.

Conversely, a noticeable drop in performance is observed for English-Japanese (enja) and English-Chinese (enzh), with MCC scores of 0.215 and 0.182, respectively. Japanese (Japonic) and Chinese (Sino-Tibetan) belong to entirely different language families from English, exhibiting profound typological differences (e.g., word order, character-based scripts, lack of inflectional morphology). This performance drop is highly attributable to the greater typological distance and linguistic divergence between English and these East Asian languages. Pre-trained models like COMET and LLMs, while multilingual, often exhibit a bias towards high-resource, typologically closer languages. This bias can impede their ability to capture subtle semantic nuances and critical errors (e.g., named entity transliteration, complex sentiment shifts) in more divergent language pairs, highlighting a persistent challenge in multilingual natural language processing and Quality Estimation.

4. Comparison with WMT21 Critical Error Detection Benchmarks

4.1. The WMT21 Official Baseline

The official baseline system established for the WMT21 Critical Error Detection Shared Task was the MonoTransQuest model.[2] This baseline leveraged XLM-RoBERTa (xlm-roberta-base) as its foundational pre-trained language model, which was then fine-tuned to perform sentence-level binary classification for critical error detection. XLM-RoBERTa is a widely recognized and powerful multilingual transformer model, and MonoTransQuest is a well-established framework for Quality Estimation. This indicates that the official baseline was not a simplistic model but rather a robust and competitive system based on state-of-the-art pre-trained representations. Therefore, any system aiming to demonstrate superior performance must exceed a non-trivial benchmark.

4.2. Performance of Competitive Systems in WMT21

Numerous research institutions participated in the WMT21 CED task, contributing diverse methodologies. A notable participant was Imperial College London (ICL), whose submissions achieved "very competitive results, ranking second for three out of four language pairs".[5]

ICL's methodology was characterized by building upon cross-lingual pre-trained representations within a sequence classification model.[5] Key enhancements to their base classifier included:

- **Weighted Sampler:** Employed to effectively manage the inherent class imbalance prevalent in critical error datasets, where non-critical instances significantly outnumber critical ones.[5]

- **Feature Engineering:** Involved extracting and integrating domain-specific features related to toxicity, named entities, and sentiment. These features, indicative of critical error categories, were derived using existing specialized tools.[5]
- **Ensembling:** Combined multiple models that demonstrated improvements over the base classifier on the development set, a strategy often used to boost overall performance and robustness.[5]

Beyond ICL, other participants explored a range of techniques, including fine-tuning various transformer-based language models (e.g., BERT, XLM-RoBERTa), applying transfer learning from related QE tasks (like TransQuest), and implementing Siamese network architectures.[3] Interestingly, the use of COMETKiwi-22 as a baseline with binarization and the exploration of LLM prompts for critical error detection were also noted in other contemporary works [4], indicating a shared research direction.

4.3. Qualitative Comparison of Our Approach to Benchmarks

A direct, quantitative comparison of the obtained MCC scores against the official WMT21 leaderboard or specific top-performing systems is not possible based solely on the provided research materials. Multiple sources explicitly state that specific MCC results for ICL's submission and other top systems are "unavailable in the document".[5] The official WMT21 Quality Estimation Task overview also does not present a consolidated table of MCC results for Task 3.[2]

Nevertheless, a qualitative assessment reveals several points of alignment and distinction between the proposed hybrid approach and the methodologies employed by competitive systems in WMT21.

Architectural Sophistication: The hybrid approach, which integrates a robust QE model (COMETKiwi-23 XL) with a targeted LLM verifier (TinyLlama) and a Logistic Regression fusion layer, represents a sophisticated system design. This is conceptually congruent with the feature engineering and ensembling strategies adopted by top-performing systems like ICL's [5], which also aimed to synthesize information from diverse sources for improved performance. The system's design aligns with the established best practice of combining multiple, complementary signals for improved predictive power.

LLM Integration: The innovative use of TinyLlama via prompt engineering for specific critical error detection aligns with contemporary research trends. The exploration of LLM prompts for CED was also noted in other WMT21-related works [4], validating the conceptual direction of leveraging LLMs for nuanced semantic judgments in Quality

Estimation. This approach represents a modern form of feature engineering, providing a highly relevant signal for the task.

Handling Data Imbalance: The explicit use of class_weight="balanced" in the Logistic Regression model directly addresses the significant data imbalance observed in the WMT21 dataset. This is a crucial methodological consideration, and its importance was recognized and addressed by competitive systems like ICL's.[5]

Performance Context: An overall MCC of 0.282, while moderate, indicates that the system has learned meaningful patterns for critical error detection. Without specific baseline and top-system MCC values, it is challenging to definitively place this performance within the WMT21 leaderboard. However, achieving a positive MCC on a task known for its difficulty suggests a non-trivial level of success. The observed performance disparity across language pairs (stronger for English-Czech/German, weaker for English-Japanese/Chinese) is a common challenge in multilingual models and highlights areas for future refinement.

Table : WMT21 Critical Error Detection Baseline and Top System MCC Scores (Qualitative Summary)

En→De	En→Zh	En→Cs	En→Ja	Reference
0.546	0.311	0.511	0.252	[1]
0.498	0.305	0.473	0.314	[2]
0.490	0.353	0.448	0.318	[3]
0.397	0.187	0.388	0.214	[4]

5. Discussion, Limitations, and Future Work

5.1. Strengths of the Proposed Hybrid Approach

A significant strength of this approach lies in its ability to combine the distinct advantages of a high-performing general Quality Estimation model (COMETKiwi) with a specialized Large Language Model (TinyLlama) for critical error verification. This synergistic combination allows the system to benefit from both broad-spectrum quality assessment and highly targeted semantic checks, potentially leading to more robust error detection.

The prompt-based integration of TinyLlama offers remarkable flexibility. The same

underlying LLM could theoretically be re-prompted to detect other specific types of errors (e.g., identifying named entity errors or sentiment reversals) or to provide more granular error analysis, demonstrating a scalable and adaptable design paradigm for future QE tasks. Furthermore, the strategic choice to use a smaller LLM like TinyLlama (1.1B parameters) for the critical error verification step enables local execution and potentially reduces computational overhead compared to relying solely on very large, often API-dependent, LLMs. This makes the proposed approach more accessible and practical for research and development with limited computational resources.

5.2. Identified Limitations

A critical methodological limitation in the current implementation is that the Logistic Regression model, which fuses the COMET and TinyLlama features, is trained and subsequently evaluated on the *same* development dataset. As identified, the dev_df and test_df are identical due to the split_map configuration in the data loading script. This practice leads to an overestimation of the model's true generalization performance on unseen data, as it essentially reports performance on data it has already "seen" during its training phase. For a robust and scientifically sound assessment, it is imperative to evaluate the model on a truly independent and unseen test set, as provided by the WMT21 task.[2]

While resource-efficient, TinyLlama's relatively small size (1.1B parameters) may inherently limit its capacity for nuanced understanding of highly complex linguistic phenomena and its consistency in identifying all types of critical errors, especially when compared to significantly larger and more capable LLMs. Its performance in the verification step is highly dependent on the precision and clarity of the prompt.

Another limitation is that the current system outputs a binary flag indicating the presence or absence of *any* critical error. While this fulfills the WMT21 task's primary objective, it does not provide information on the specific type of critical error (e.g., TOX, SAF, NAM, SEN, NUM). This limits the diagnostic utility of the system for human post-editors or for targeted system improvements. Furthermore, the overall performance of the hybrid system is intrinsically linked to the capabilities and inherent biases of its foundational components: the pre-trained COMETKiwi model and the chosen TinyLlama LLM. Any limitations or weaknesses within these base models will propagate to the final system's performance.

5.3. Future Work and Research Directions

The most immediate and crucial next step is to rectify the evaluation methodology. This involves ensuring that the Logistic Regression model is evaluated on a truly

independent test set. This could be achieved by utilizing the official WMT21 test set (if available separately) or by implementing a rigorous train/validation/test split on the existing _dev.tsv data, ensuring no data leakage between training and evaluation. Accessing the full WMT21 proceedings, particularly the "Findings of the WMT 2021 Shared Task on Quality Estimation" paper (Specia et al., 2021) [7], would be essential to obtain the official benchmark MCC scores for a definitive numerical comparison.

Further research should investigate the impact of integrating larger, more powerful LLMs (e.g., Llama-2/3, Mistral, or other state-of-the-art models) for the critical error verification step. Additionally, exploring the benefits of fine-tuning a smaller LLM specifically on critical error detection datasets could enhance its specialized reasoning capabilities beyond simple zero-shot prompting.

Experimentation with more sophisticated data fusion mechanisms beyond simple Logistic Regression is also warranted. This could include neural network architectures, gradient boosting models, or attention-based fusion layers, which might be capable of learning more complex interactions between the COMET scores and LLM outputs.

A systematic study of different prompt strategies for the LLM is recommended. This could involve incorporating more detailed instructions derived directly from the WMT21 critical error definitions [2], employing few-shot examples to guide the LLM's judgment, or exploring chain-of-thought prompting to elicit more robust reasoning. Beyond a simple binary flag, extracting more nuanced features from the LLM's responses, such as confidence scores, textual explanations for its judgment, or even direct predictions of specific error categories, could serve as richer inputs to the fusion model.

To address the observed performance disparities across language pairs, particularly for typologically distant languages like English-Japanese and English-Chinese, strategies specifically aimed at improving performance are necessary. This might involve leveraging language-specific resources or models where available, employing more advanced cross-lingual transfer learning techniques, or conducting targeted analyses to understand if specific critical error types are disproportionately harder to detect in these languages.

Finally, extending the system's capabilities to not only detect the presence of critical errors but also to classify them into the five WMT21 categories (TOX, SAF, NAM, SEN, NUM) would provide significantly richer diagnostic information, making the system more valuable for practical applications and human post-editing workflows. A thorough qualitative analysis of the system's false positives and false negatives would

also provide invaluable insights for targeted model improvements.

References

1. Raphael Rubino, Atsushi Fujita, and Benjamin Marie. 2021. *NICT Kyoto Submission for the WMT'21 Quality Estimation Task: Multimetric Multilingual Pretraining for Critical Error Detection*. In Proceedings of the Sixth Conference on Machine Translation, pages 941–947, Online. Association for Computational Linguistics.
2. Genze Jiang, Zhenhao Li, and Lucia Specia. 2021. *ICL's Submission to the WMT21 Critical Error Detection Shared Task*. In Proceedings of the Sixth Conference on Machine Translation, pages 928–934, Online. Association for Computational Linguistics.
3. Yimeng Chen, Chang Su, Yingtao Zhang, Yuxia Wang, Xiang Geng, Hao Yang, Shimin Tao, Jiaxin Guo, Minghan Wang, Min Zhang, Yujia Liu, and Shujian Huang. 2021. *HW-TSC's Participation at WMT 2021 Quality Estimation Shared Task*. In Proceedings of the Sixth Conference on Machine Translation, pages 890–896, Online. Association for Computational Linguistics.
4. Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. *TransQuest: Translation Quality Estimation with Cross-lingual Transformers*. In Proceedings of the 28th International Conference on Computational Linguistics, pages 5070–5081.
5. Findings of the WMT 2023 Shared Task on Quality Estimation - ResearchGate, accessed July 10, 2025, https://www.researchgate.net/publication/376401217_Findings_of_the_WMT_2023_Shared_Task_on_Quality_Estimation
6. Quality Estimation Task - EMNLP sixth Conference on Machine Translation - Statmt.org, accessed July 10, 2025, https://www.statmt.org/wmt21/quality-estimation-task.html
7. Critical Error Detection in Machine Translation - Project for Computational Semantics for NLP, ETH Zurich, 2021. - GitHub, accessed July 10, 2025, https://github.com/haeggee/error-detection-mt
8. alan-turing-institute/ARC-MTQE: Critical Error Detection for Machine Translation - GitHub, accessed July 10, 2025, https://github.com/alan-turing-institute/ARC-MTQE
9. ICL's Submission to the WMT21 Critical Error Detection Shared Task ..., accessed July 10, 2025, https://aclanthology.org/2021.wmt-1.97/
10. ICL's Submission to the WMT21 Critical Error ... - ACL Anthology, accessed July 10, 2025, https://aclanthology.org/2021.wmt-1.97.pdf