Journal of Creative Student Research
Volume. 2 Nomor. 6 Desember 2024
e-ISSN : 2963-4776; p-ISSN : 2963-5942, Hal. 238-258
DOI: https://doi.org/10.55606/jcsr-politama.v2i6.6153
Tersedia: https://ejurnal.politeknikpratama.ac.id/index.php/jcsr

Hybrid Subword-Character Representation for Robust Sentiment Classification on Multilingual and Code-Mixed Indonesian Text

Danang [1]*, Toni Wijanarko Adi Putra [2]
[1-2] Universitas Sains dan Teknologi Komputer
Email: danang150787@gmail.com [1], toni.wijanarko@stekom.ac.id [2]

**Penulis Korespondensi: danang150787@gmail.com*

Abstract: User-generated Indonesian text frequently exhibits code-mixing with English ("Indonglish"), informal spelling, elongation, and keyboard typos. These phenomena break subword tokeniza tion assumptions and may degrade multilingual Transformer performance in deployment. This paper studies a hybrid representation that fuses XLM-R sentence features with a character-level CharCNN branch designed to capture orthographic patterns and mitigate character noise. We evaluate (i) a standard XLM-R fine-tuning baseline, (ii) an ablation that removes the character branch (NusaX only), and (iii) the proposed hybrid model on two datasets: NusaX-Senti (12 regional languages) and Indonglish (Indonesian–English code-mixed sentiment). Beyond clean test performance, we introduce a controlled robustness protocol by injecting character-level perturbations with probability p=0.18 and measuring performance drop. Results show that the XLM-R baseline achieves the best clean Macro-F1 on both datasets, while the hybrid model substantially improves robustness on Indonglish by reducing Macro-F1 drop from 0.030 to 0.007 under noise. We analyze common error confusions and discuss when character-aware features help or harm across languages.

Keywords: sentiment analysis; code-mixing; multilingual NLP; robustness; character-level modeling; XLM-R.

1. INTRODUCTION

Sentiment classification is a core building block for monitoring customer feedback, social media analytics, and service-quality evaluation at scale. In practice, user-generated text is noisy and highly variable, and this is especially visible in the Indonesian context where informal writing dominates many public and private platforms. Real-world inputs rarely follow formal spelling conventions: users often mix Indonesian with English expressions (e.g., "overall service-nya good banget"), use elongations for emphasis (e.g., "mantaaap"), abbreviate or drop characters, and produce inconsistent forms due to mobile keyboards and autocorrect. Beyond Indonesian English mixing, multilingual settings in Indonesia may also involve regional languages and borrowed orthographic patterns, creating a persistent mismatch between training corpora and deployment text. This mismatch is not merely cosmetic: small surface-form changes can shift tokenization outcomes and ultimately affect downstream predictions, which matters for operational systems that rely on stable sentiment signals.

Modern Transformer encoders (Vaswani et al., 2017) trained with subword tokenization, such as XLM-R (Conneau et al., 2020), are a strong default for multilingual text classification. They inherit the representational strengths popularized by large-scale pretraining paradigms (Devlin et al., 2019; Liu et al., 2019) and commonly rely on subword tokenizers such as

Naskah Masuk: September 12, 2024; Revisi: September 23, 2024; Diterima: November 20, 2024; Tersedia: Desember 25, 2024; Terbit: Desember 28, 2024

SentencePiece (Kudo & Richardson, 2018). However, subword tokenization can be brittle under character-level noise: a single typo may alter token boundaries, increase fragmentation into rare pieces, and change the effective sequence composition seen by the encoder. In other words, the model may receive a substantially different input representation even when the underlying human-intended 1 meaning is unchanged. This brittleness is particularly relevant for code-mixed and informal Indonesian text, where orthographic variation is frequent and systematic.

Character-aware modeling provides a complementary view that is more directly tied to surface form and can capture patterns that subword segmentation may not handle reliably, such as repeated characters, frequent misspellings, and mixed-language orthography. Prior work has explored character- and tokenization-aware pretraining as a way to improve robustness to noisy text, for example by integrating character signals into pretrained models (El Boukkouri et al., 2020) or removing explicit tokenization via tokenization-free encoders (J. H. Clark et al., 2021; Xue et al., 2021). In addition, robustness and augmentation frameworks have highlighted how small perturbations can degrade NLP models and how controlled noise can be used to stress-test systems (Morris et al., 2020; Wei & Zou, 2019). Motivated by these findings, we examine a practical hybrid approach that preserves a strong subword Transformer baseline while injecting lightweight character-level features targeted at orthographic variation.

This work investigates a hybrid representation for robust sentiment classification that combines (i) a subword Transformer branch (fine-tuned XLM-R) and (ii) a character-level convolutional branch (CharCNN), where pooled character features are fused with Transformer features and fed into a classifier head. We focus on a concrete operational question: whether adding explicit character features improves robustness to noisy, code-mixed Indonesian text while keeping clean accuracy competitive. To answer this, we evaluate on multilingual and code-mixed sentiment benchmarks and introduce a controlled perturbation protocol that simulates common character-level noise patterns.

2. LITERATURE REVIEW

This section must contain a state-of-the-art explanation. It can be explained in several ways. First, you can discuss several related papers, both about objects, methods, and their results. From there, you can explain and emphasize gaps or differences between your research and previous research. The second way is to combine theory with related literature and explain each theory in one sub-chapter.

Multilingual Transformers

Transformer architectures (Vaswani et al., 2017) and large-scale pretraining have become the dominant approach for multilingual and cross-lingual text classification. Early work such as BERT (Devlin et al., 2019) established the effectiveness of masked language modeling, while subsequent variants focused on more robust optimization and training recipes, e.g., RoBERTa

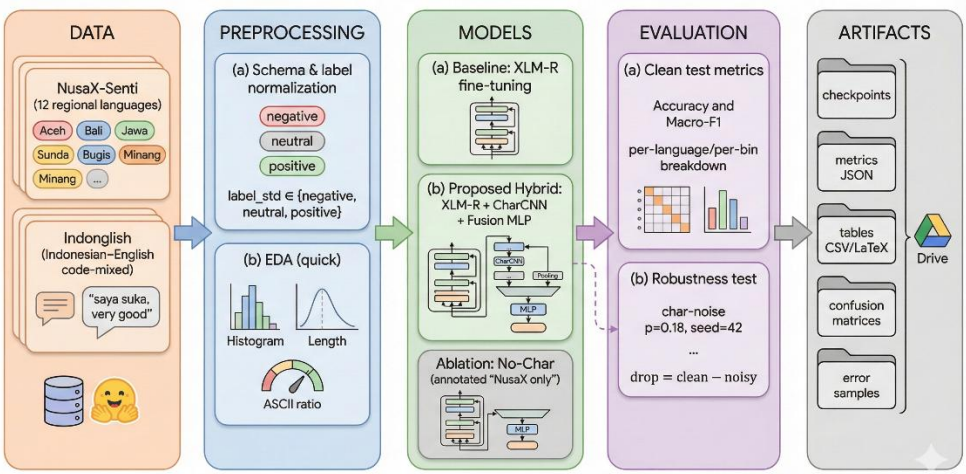

Figure 1. End-to-end experimental pipeline: datasets, preprocessing, models, evaluation, and saved artifacts.

(Liu et al., 2019). For multilingual coverage, XLM-R (Conneau et al., 2020) remains a strong baseline because it is trained on large multilingual corpora and typically transfers well across languages and domains. In Indonesian settings, this is attractive because practical deployments often involve a mixture of Indonesian, regional languages, and borrowed English expressions.

Beyond masked-LM encoders, several alternative pretraining objectives and architectures have also been explored and are frequently used as backbones in NLP systems. XLNet (Yang et al., 2019) proposes generalized autoregressive pretraining, ELECTRA (K. Clark et al., 2020) learns via replaced-token detection, and ALBERT (Lan et al., 2019) improves parameter efficiency via factorization and sharing. While these models can deliver strong clean accuracy, their robustness in user-generated settings is still influenced by the input representation pipeline, where tokenization is a critical component. Most Transformer encoders rely on subword tokenization (e.g., SentencePiece (Kudo & Richardson, 2018)), which compresses vocabulary size and improves coverage but can become brittle under character-level perturbations that shift segmentation boundaries. As a response, tokenization-free or

tokenization-minimized approaches have gained attention, including byte-level and character/byte-to-byte pretraining (J. H. Clark et al., 2021; Xue et al., 2021), which aim to reduce sensitivity to spelling variation by operating closer to the raw text.

Character-Aware and Noise-Robust Representations

Character-aware modeling introduces an explicit inductive bias toward surface form. This is useful when orthographic cues carry meaning (e.g., elongation for emphasis) or when inputs are corrupted by typing noise. One direction integrates character signals directly into pretrained models, for example CharBERT (El Boukkouri et al., 2020), while another direction removes explicit tokenization constraints via tokenization-free encoders (J. H. Clark et al., 2021; Xue et al., 2021). These approaches motivate the broader hypothesis that representations anchored at the character or byte level can be less sensitive to perturbations that destabilize subword segmentation.

A pragmatic alternative, especially when starting from a strong multilingual encoder, is to 3 attach a lightweight character module that complements the subword branch. Such a design is conceptually aligned with classic subword-aware embedding work (e.g., enriching word vectors with subword information) (Bojanowski et al., 2017), but adapted to modern Transformer backbones. In practice, a small character-CNN branch is often appealing due to simplicity, low additional compute, and clear interpretability: it explicitly targets repeated characters, missing characters, and local orthographic patterns that may not be preserved after subword segmentation. Our hybrid approach follows this pragmatic path: we keep XLM-R as the primary semantic encoder and add a compact character branch to inject robustness-relevant surface features, rather than replacing the backbone with a fully tokenization-free model.

Code-Mixed Sentiment Analysis

Code-mixed sentiment analysis is challenging because it combines multilingual phenomena (lexical borrowing, syntax mixing) with the informal style of social media. Shared tasks such as SemEval SentiMix (Patwa et al., 2020) highlight how code-mixed text amplifies lexical variation and increases the likelihood of nonstandard spelling. In Indonesian–English mixtures, additional complexity arises from informal affixation and particle attachment (e.g., "-nya") and from frequent switching within short spans, which can confuse both lexicon-based cues and purely subword-driven representations.

For Indonesian and Southeast Asian languages, the availability of benchmarks and resources has improved in recent years. NusaX (Winata et al., 2023) and SEACrowd (Winata, Cahyawijaya, et al., 2024) support broader evaluation across languages and tasks, enabling

more systematic comparisons across multilingual regimes. In this work, we focus on NusaX-Senti and Indonglish to represent two complementary regimes: (i) multilingual diversity involving regional languages, and (ii) Indonesian–English code-mixing in informal user text. This pairing allows us to test not only average clean accuracy but also robustness under realistic orthographic variation where code-mixing and noise often co-occur.

Data Augmentation and Robustness Evaluation

Text augmentation is widely used to improve generalization and mitigate overfitting, especially when labeled data is limited or noisy. EDA (Wei & Zou, 2019) popularized simple perturbation operators such as insertion and deletion, while TextAttack (Morris et al., 2020) systematized adversarial perturbations and evaluation pipelines. In robustness research, it is common to evaluate models under controlled perturbations and report performance degradation as an operational robustness signal, rather than relying on anecdotal examples. This aligns with our motivation: in real deployment, the main concern is not only peak clean accuracy, but whether performance collapses when inputs deviate from training distributions due to typos, elongations, or informal orthography.

In addition to character-level perturbations, robustness can also be influenced by modeling and tooling choices. Sequence-to-sequence pretraining (e.g., T5 (Raffel et al., 2020) and BART (Lewis et al., 2020)) has been used in related contexts for normalization, rewriting, or generating augmented variants, although such approaches may add computational overhead and introduce semantic drift if not carefully controlled. For our setting, we adopt targeted character-level operators (e.g., swap, elongation, deletion, and substitution) to keep perturbations local and interpretable, and we quantify robustness via clean-to-noisy performance drops. Finally, to ensure that robustness comparisons are reproducible, we align our implementation with widely used open-source tooling: the Transformers library (Wolf et al., 2020), dataset utilities (Lhoest et al., 2021), scalable training helpers (Gugger et al., 2022), and the PyTorch framework (Paszke et al., 2019). Optimization details such as AdamW (Loshchilov & Hutter, 2019) are also standard in fine-tuning regimes and help make results comparable across studies.

3. PROPOSED METHOD

Datasets

We evaluate on two datasets that represent complementary real-world challenges for Indone sian sentiment classification: multilingual diversity across regional languages and

informal Indonesian–English code-mixing. These settings are common in practical monitoring pipelines where sentiment signals must remain reliable even when users switch languages, use informal orthography, or mix borrowed expressions.

priors. NusaX-Senti is a multilingual sentiment benchmark derived from the NusaX suite (Winata et al., 2023). It covers 12 languages (Indonesian, English, and 10 Indonesian local languages), which makes it suitable for testing whether a single multilingual encoder can generalize across typologically diverse languages and domain shifts. Because regional-language content can differ in spelling conventions and word formation, NusaX-Senti provides a natural stress test for models that rely heavily on subword segmentation and pretrained lexical priors.

Indonglish is an Indonesian–English code-mixed sentiment dataset available within SEACrowd resources (Winata, Cahyawijaya, et al., 2024). Unlike purely multilingual benchmarks where each example tends to stay within a language, Indonglish features intra-sentence switching and mixed orthography that is typical of informal reviews and social media. This dataset is particularly relevant for robustness because code-mixed text often co-occurs with spelling variation, informal affixation (e.g., "-nya"), and user-typed noise.

For comparability across datasets, we map all labels into a unified 3-class label set: Y = {negative,neutral,positive}. This unification reflects practical sentiment monitoring scenarios that prioritize stable, interpretable categories. Fig. 2 provides illustrative examples of the two datasets after label standardization; these examples are shown for communication and are not direct quotes from the underlying corpora.

PANEL A: NusaX-Senti (12 languages; multilingual)

Lang	Example text (illustrative)	Label
ID	'Pelayanannya cepat, tapi tempatnya penuh.'	NEU
EN	'Great taste, would order again.'	POS
JV	'Regane larang, nanging kualitas apik.'	NEU
SU	'Teu puas, lila pisan nungguan.'	NEG
MIN	'Raso enak bana, pasti bali lai.'	POS
MAD	'Kelleh, tapi e toles.'	NEU

Illustrative examples (not direct quotes)

PANEL B: Indonglish (Indonesian–English code-mixed)

• 'overall service-nya good banget, recommended.'	POS
• 'the staff ramah, but the wait was too long.'	NEU
• 'price-nya oke, but kualitasnya mengecewakan.'	NEG
• 'tempatnya cozy, aku suka vibes-nya.'	POS
• 'packing-nya rapi, but rasanya biasa aja.'	NEU
• 'chat admin slow, jadi aku cancel.'	NEG

Illustrative examples (not direct quotes)

Figure 2. Dataset Examples Used In this study.

Preprocessing and Normalization

Given raw text x, our preprocessing is intentionally light-weight so that sentiment-bearing cues are preserved. Over-normalization can remove informative signals such as elongation 5 (e.g., repeated characters used for emphasis) or informal spellings that correlate with polarity. Therefore, we restrict preprocessing to three core steps: (1) whitespace cleanup (trimming and collapsing repeated spaces), (2) label standardization into the unified set Y, and (3) character encoding required by the CharCNN branch. We keep punctuation and casing as-is when they appear naturally in the data because they can serve as sentiment markers (e.g., repeated exclamation marks).

In addition to these transformations, we compute descriptive statistics to characterize each dataset and to support reproducible reporting. We track length distributions at both the token/subword level and the character level, since the two branches in our model consume different representations. We also compute an ASCII ratio as part of exploratory analysis, which serves as a coarse proxy for the presence of English segments and mixed-language orthography in Indonesian text. These analyses are saved as artifacts within the experiment pipeline (Fig. 1), enabling traceability from raw inputs to standardized representations and reported results**.**

Baseline Model: XLM-R Fine-Tuning

Our baseline follows the standard fine-tuning recipe for multilingual Transformers, using XLM R (Conneau et al., 2020). The encoder is based on the Transformer architecture (Vaswani et al., 2017) and inherits the general effectiveness of pretrained language models established by BERT-style masked language modeling (Devlin et al., 2019) and robust optimization strategies such as RoBERTa (Liu et al., 2019). Inputs are converted into subword token ids using a SentencePiece-style tokenizer (Kudo & Richardson, 2018), which provides broad coverage while keeping vocabulary size manageable.

Let x denote the raw input string and Tok(·) denote the subword tokenizer. We obtain token ids t = Tok(x) and contextual representations H ∈ RL×d, where L is the subword sequence length and d is the hidden dimension. We use a pooled feature vector (e.g., the [CLS] embedding) h ∈Rd and train a linear classifier for 3-class prediction:

$$\hat{y} = softmax(Wh + b),\ W \in \mathbb{R}^{|y| \times d} \quad (1)$$

This baseline is intentionally strong and widely used in multilingual sentiment classification. At the same time, because the representation depends on subword segmentation,

it can be sensitive to character-level noise that shifts token boundaries and increases fragmentation into rare pieces.

Hybrid Representation: XLM-R + CharCNN

The proposed model augments the baseline with a character-level branch designed to capture surface-form signals that may be unstable under subword tokenization. The motivation is pragmatic: rather than replacing the multilingual backbone, we keep XLM-R as the primary semantic encoder and add a lightweight character module that explicitly targets orthographic variation. This complements subword modeling, which already benefits from subword-aware representations (Bojanowski et al., 2017).

An overview of the architecture is shown in Fig. 3. The model consists of two parallel branches whose outputs are fused before classification. The Transformer branch provides contextual semantic features robust to vocabulary variation in clean settings, while the character branch provides local orthographic cues that are directly sensitive to repeated characters, missing characters, and mixed-language spelling patterns.

Character Encoding. We build a character vocabulary Vc from the training split and encode each input into a fixed-length character id sequence c = (c1,...,cT), where T is a maximum character length. A padding mask m indicates valid positions. We embed each 6 character into a dense vector to form E ∈ RT×dc, where dc is the character embedding dimension. This representation is intentionally close to the raw text, which allows the branch to respond consistently to character-level perturbations that might otherwise distort subword segmentation outcomes.

Character CNN. We apply 1D convolutions over E using multiple kernel widths k ∈ {3,4,5} to capture local character n-gram patterns at different granularities. For each kernel width, we apply a nonlinearity σ(·) followed by global max pooling:

$$z\_k = MaxPool(\sigma(Conv1D\_k(E)\ )) \quad (2)$$

and concatenate pooled outputs to obtain a fixed-size character feature vector c ∈ Rdk. This design is computationally light and easy to integrate into a Transformer fine-tuning pipeline. Compared with tokenization-free encoders such as ByT5 (Xue et al., 2021) and CANINE (J. H. Clark et al., 2021), or character-aware pretrained Transformers such as CharBERT (El Boukkouri et al., 2020), our approach keeps the backbone unchanged and isolates the benefit of adding explicit character cues.

Fusion and Classification. We fuse the Transformer pooled feature h and the character feature c by concatenation:

$$u = [h; c] \in \mathbb{R}^{d + d\_k} \quad (3)$$

and pass the fused vector through a small multi-layer perceptron (MLP) with dropout and ReLU activation before the final classification layer. The goal of fusion is not to force the model to rely on character features universally, but to make them available when subword representations become unreliable due to noise. Dropout regularization encourages complementary usage of both branches and reduces overfitting to dataset-specific shortcuts.

Training Objective

We train both baseline and hybrid models using the standard cross-entropy objective for multi class classification:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{y \in \mathcal{Y}} I[y_i = y] \log \widehat{p_{i,y}} \quad (4)$$

where N is the number of training examples and ˆpi,y denotes the predicted probability for class y. Optimization uses AdamW-style decoupled weight decay (Loshchilov & Hutter, 2019), consistent with standard Transformer fine-tuning practice. Implementation-wise, our training and evaluation stack follows widely used community tooling to support reproducibility and transparent artifact generation. We build the models and training loops in PyTorch (Paszke et al., 2019) and use the Transformers library (Wolf et al., 2020) for pretrained model loading and fine-tuning. Dataset loading, processing, and split handling follow the Hugging Face datasets ecosystem (Lhoest et al., 2021), and we use scalable training utilities when applicable (Gugger et al., 2022). These choices do not change the modeling contribution, but they ensure that training is consistent with common practice and that intermediate artifacts (metrics tables, confusion matrices, and prediction samples) can be generated systematically.

Models Compared

We compare three model variants to isolate the contribution of character-level information and to evaluate the trade-off between clean accuracy and robustness under perturbations. The

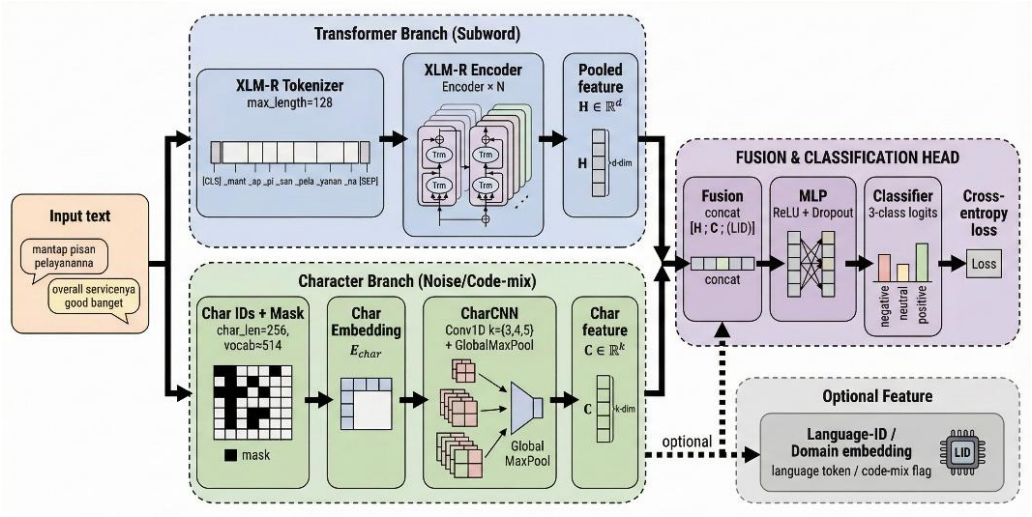

Figure 3. Hybrid subword-character architecture: XLM-R encoder branch and CharCNN branch.

comparison is intentionally structured around a strong multilingual baseline and a minimally invasive augmentation, because the practical question is whether robustness can be improved without replacing the backbone or requiring expensive retraining from scratch. All variants are trained with the same data splits, label mapping, and evaluation protocol so that observed differences can be attributed to modeling choices rather than experimental confounds.

Baseline uses a fine-tuned XLM-R encoder (Conneau et al., 2020) with a standard classifica tion head. This configuration reflects common practice for multilingual sentiment classification: a pretrained Transformer encoder (Vaswani et al., 2017) with a pooled representation (e.g., [CLS]) followed by a linear layer. It serves as a high-quality reference for clean performance and provides a realistic yardstick for deployment-like conditions.

Hybrid augments the baseline by adding a lightweight character-level branch implemented as CharCNN and fusing pooled character features with pooled Transformer features before classification. The intent is to provide an explicit surface-form channel that remains informative when subword tokenization becomes unstable. Unlike tokenization-free encoders that require a different pretraining regime (J. H. Clark et al., 2021; Xue et al., 2021) or character-aware pretrained Transformers (El Boukkouri et al., 2020), the hybrid setting keeps the pretrained backbone unchanged and focuses on a targeted modification that can be applied to existing systems with minimal disruption.

Ablation (NusaX only) removes the character features while keeping the rest of the hybrid pipeline as similar as possible. This ablation is included to avoid an uninformative comparison where improvements are explained away by additional parameters or classifier depth. By isolating the character pathway, the ablation helps answer whether robustness gains (if any) arise from the character information itself, rather than from fusion-related regularization or an expanded classifier head. We run this ablation on NusaX-Senti as a multilingual benchmark where the distributional shift across languages provides a nontrivial setting for generalization.

Hyperparameters

Our hyperparameter design follows standard fine-tuning practice for pretrained Transformers (Devlin et al., 2019; Liu et al., 2019) while explicitly controlling the parts that could otherwise confound comparisons between model variants. We set the maximum subword length to 128, 8 which is appropriate for short-form reviews and social-media-like texts and keeps computational cost manageable. When inputs exceed this length, they are truncated consistently across models to ensure comparability. Batch size is chosen to fit GPU memory, and we use the same effective batch size policy across variants (e.g., with gradient accumulation if needed) so that the hybrid model is not inadvertently advantaged by different optimization dynamics.

Regularization is applied at the classifier level via dropout, and model selection is performed using early stopping on validation Macro-F1. Early stopping is included to mitigate sensitivity to over-training, which can otherwise amplify differences between variants in a way that does not translate to stable test performance. We also keep random seeds fixed for data shuffling, noise generation, and model initialization where applicable, because robustness comparisons are particularly sensitive to small stochastic differences.

For the character branch, we fix the character sequence length to T=256 for efficiency and consistent tensor shapes. The character vocabulary is built from training data and typically results in a compact set (approximately 500 characters after normalization), covering common Latin letters, digits, punctuation, and frequently occurring symbols. We use multiple convolution kernel widths {3,4,5} with global max pooling, which provides a simple multi-scale mechanism to capture local orthographic patterns such as repeated characters, short typos, and common affix like fragments. In the fusion MLP, dropout and nonlinearity are used to encourage complementary feature usage rather than collapse onto a single branch.

Optimization uses AdamW-style decoupled weight decay (Loshchilov & Hutter, 2019), which is widely adopted in Transformer fine-tuning due to stable convergence and effective regularization. All models share the same optimization family and stopping criterion so that clean accuracy and robustness differences are attributable to representation choices rather than optimizer discrepancies.

Evaluation Metrics

We report Accuracy and Macro-F1 for 3-class sentiment classification, with Macro-F1 treated as the primary metric for comparison. Accuracy is intuitive but may hide systematic failures when class distributions are imbalanced or when a model over-predicts a dominant class. Macro-F1 mitigates this by computing the F1 score independently for each class and averaging them equally, which makes it more sensitive to minority-class performance and common confusion patterns involving the neutral class.

Formally, dataset-level Macro-F1 is computed as:

$$\text{MacroF1} = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \text{F1}(y) \tag{5}$$

In addition to aggregate metrics, we examine confusion matrices and representative error samples for interpretability. This is important because robustness evaluation can produce similar overall drops while hiding different failure modes (e.g., a model might become conservative and over predict neutral, or it might shift errors toward positive under certain perturbations). We therefore use Macro-F1 as the quantitative headline metric while treating confusion profiles as supporting evidence for qualitative analysis.

Robustness Evaluation via Character-Level Noise

Robustness is evaluated by constructing a noisy test set that simulates realistic character level variation in user-generated text. We apply perturbations with probability p=0.18, which introduces moderate corruption while keeping text largely interpretable. The perturbation process is controlled by a fixed random seed so that the noisy set is deterministic and comparable across model variants, enabling fair evaluation of clean-to-noisy degradation.

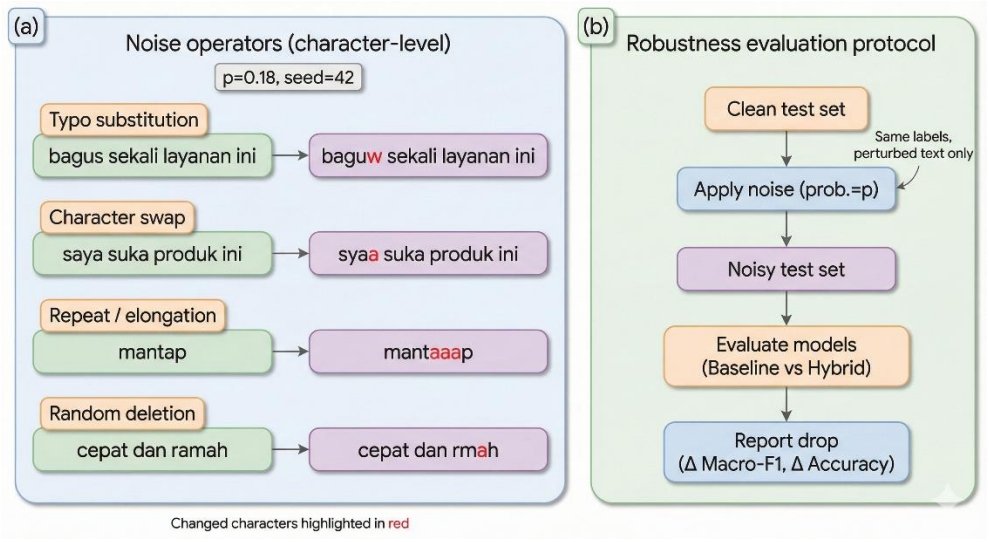

Figure 4. Character-level noise operators and robustness evaluation protocol.

The set of noise operators includes typo substitution, character swap, elongation (repetition), and random deletion (Fig. 4). These operators are motivated by common sources of noise in Indonesian and code-mixed settings: fast mobile typing, keyboard proximity errors, emphasis through character repetition, and accidental omission of characters. Crucially, labels remain unchanged; only the input surface form is perturbed. This ensures that the robustness protocol measures the model's sensitivity to orthographic variation rather than introducing semantic changes that would justify a label change.

Our choice of character-level perturbations is aligned with broader augmentation and ro bustness tooling. For example, EDA (Wei & Zou, 2019) popularizes simple perturbations that improve generalization, while TextAttack (Morris et al., 2020) offers a systematic framework for perturbation-driven evaluation. However, because our hypothesis targets tokenization brittleness and character-aware complements, we focus on perturbations that directly affect the character stream and subword segmentation outcomes.

We quantify robustness via the score drop between clean and noisy evaluation:

$$\Delta\text{MacroF1} = \text{MacroF1}_{\text{clean}} - \text{MacroF1}_{\text{noisy}} \quad (6)$$

A smaller Δ indicates better robustness. We report both clean scores and drops to avoid a misleading interpretation where a model appears "robust" simply because its clean performance is already low. Where relevant, we also inspect whether robustness gains come uniformly across classes or whether they are concentrated on particular confusion pairs.

Implementation and Reproducibility

All experiments are implemented in PyTorch (Paszke et al., 2019) and use the Hugging Face Transformers ecosystem (Wolf et al., 2020) for pretrained model loading, tokenization, and f ine-tuning. Dataset loading and preprocessing use the datasets library (Lhoest et al., 2021), which supports caching and deterministic reprocessing, while training utilities are supported by accelerate (Gugger et al., 2022).

Tabel 1. Software-engineering perspective.

Dataset	Model	Accuracy	Macro-F1
NusaX	Baseline (XLM-R)	0.821	0.810
NusaX	Ablation (No-char)	0.806	0.796
NusaX	Hybrid (XLM-R +Char CNN)	0.785	0.775
Indonglish	Baseline (XLM-R)	0.754	0.748
Indonglish	Hybrid (XLM-R +Char CNN)	0.740	0.732

4. RESULTS AND DISCUSSION

In this section, the author needs to explain the hardware and software used, dataset sources, initial data analysis, results, and results analysis/discussion. Presenting the results with pictures, graphs and tables is highly recommended. Formulas or evaluation measuring tools also need to be included here. There must be discussion/analysis, and you can't just rewrite the results in sentence form, but you need to provide an explanation of their relationship to the initial hypothesis. In addition, this section needs to discuss and elaborate on important findings.

Clean Test Performance

Table 1 summarizes clean test performance on both datasets. Overall, the XLM-R baseline achieves the strongest Macro-F1, confirming that multilingual subword Transformers remain highly competitive for standard sentiment classification evaluation (Conneau et al., 2020). This outcome is consistent with the broader success of pretrained Transformer encoders (Devlin et al., 2019; Liu et al., 2019; Vaswani et al., 2017), where clean accuracy is typically dominated by high-capacity contextual representations learned during pretraining.

On NusaX-Senti, the ablation result sits between the baseline and the hybrid model. This pattern suggests that (i) introducing the character branch does not automatically improve clean performance, and (ii) the specific fusion configuration can matter as much as the presence of character information. In multilingual settings, the baseline may already encode sufficient lexical and contextual cues through subword units, while the additional character channel can introduce an extra degree of freedom that increases the risk of overfitting or interference

between branches. On Indonglish, the hybrid underperforms the baseline on clean data by a moderate margin (0.748 vs 0.732 Macro-F1), indicating that the added character pathway does not provide an advantage when inputs are clean and tokenization is stable. However, clean performance alone is not the full story for informal and code-mixed deployment conditions, where robustness to surface-form variation is often the operational bottleneck.

Robustness Under Character Noise

Table 2 and Fig. 5 highlight the main robustness finding: the hybrid model substantially reduces performance degradation under character-level perturbations. We generate noisy test sets by applying controlled character-level operators (Fig. 4) that mimic realistic user-generated variation, following the spirit of perturbation-based robustness evaluation and augmentation toolkits (Morris et al., 2020; Wei & Zou, 2019). Under p=0.18 corruption, the Macro-F1 drop on Indonglish decreases from 0.030 (baseline) to 0.007 (hybrid). This is a large relative reduction in degradation and suggests that character-level cues help stabilize predictions when surface form is perturbed, especially in code-mixed settings where tokenization brittleness is likely to be more pronounced.

Table 2. The hybrid model substantially reduces performance degradation under character-level perturbations.

Dataset	Model	Clean Acc	Clean F1	Noisy Acc	Noisy F1	Acc Drop	F1 Drop
Indonglish	Baseline (XLM-R)	0.754	0.748	0.724	0.718	0.030	0.030
Indonglish	Hybrid (XLM-R + CharCNN)	0.740	0.732	0.733	0.724	0.007	0.007
NusaX	Ablation (No-Char)	0.806	0.796	0.792	0.782	0.014	0.014
NusaX	Baseline (XLM-R)	0.821	0.810	0.804	0.792	0.017	0.019
NusaX	Hybrid (XLM-R + CharCNN)	0.785	0.775	0.773	0.763	0.012	0.012

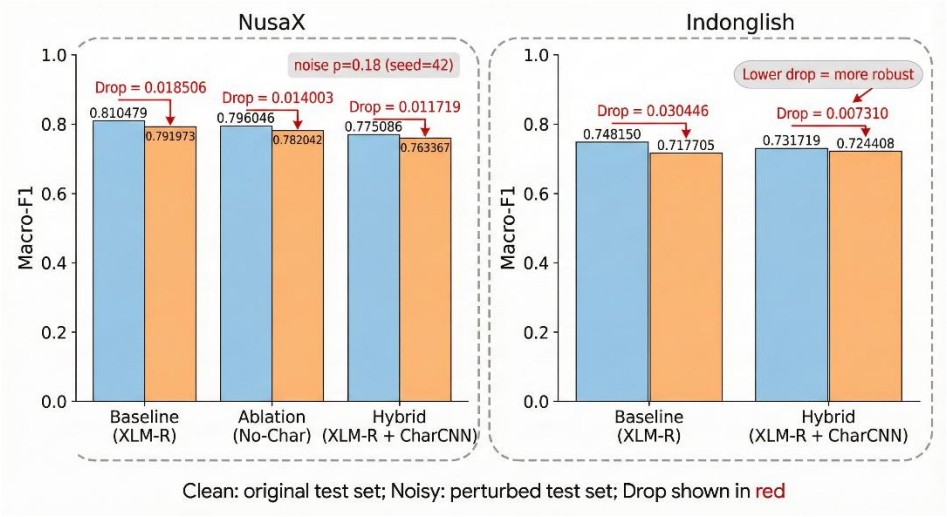

Figure 5. Robustness comparison on clean vs noisy test sets. Smaller drop indicates better.

For NusaX-Senti, robustness improvements exist but are smaller: the Macro-F1 drop reduces from 0.019 to 0.012. One plausible interpretation is that NusaX-Senti already benefits from multilingual pretraining and may contain fewer systematic code-mixed orthographic patterns than Indonglish, so the character channel provides less incremental robustness. Another explanation is that multilingual variation across languages can dominate error sources, meaning that character noise is not the only factor that drives degradation. In that scenario, a character branch may help on some languages or writing styles, but the aggregate improvement becomes diluted when averaged across diverse language distributions. Nevertheless, the direction of change is consistent across both datasets: the hybrid tends to reduce sensitivity to character perturbations, even when it does not maximize clean accuracy.

Error Patterns

To better understand the quantitative trends, we analyze confusion matrices and represen tative error samples saved by our artifact pipeline. Across both baseline and hybrid models, neutral→positive and neutral→negative confusions remain dominant. This pattern is expected in subjective review text where neutrality is often expressed implicitly or mixed with faint polarity cues, and where the same surface cues (e.g., intensifiers or hedging) can be interpreted differently across users and domains. In addition, code-mixed utterances can contain polarity-bearing English words embedded in Indonesian structure, which may push borderline neutral cases toward positive or negative depending on context. Importantly, the robustness gains from the hybrid do not imply that the neutral class becomes “solved.” Instead, the hybrid primarily reduces noise-induced errors, i.e., cases where the same semantic content flips labels due to

small character perturbations. Ambiguity in neutrality remains a separate modeling challenge. From an application perspective, this suggests that systems relying on sentiment should treat the neutral class carefully, for example by incorporating calibrated confidence, thresholding, or human-in-the-loop review for borderline predictions. Alternative formulations such as ordinal sentiment modeling or additional context (e.g., aspect-level sentiment) may also be needed to reduce inherent ambiguity beyond what robustness-focused features can provide.

Why Does Hybrid Help Robustness but Hurt Clean Accuracy?

The clean robustness trade-off observed in our results is consistent with a practical tension in representation learning. The character branch provides access to orthographic regularities (e.g., elongation, informal spellings, or consistent code-mixed affix patterns) that can be highly informative when subword tokenization becomes unstable. In noisy regimes, this acts as a stabilizer: even if subword segmentation fragments an input into rare pieces, the character channel can preserve local patterns that correlate with sentiment-bearing cues.

However, on clean test data, these same surface cues may become weakly predictive or spuriously correlated with labels, encouraging the hybrid to allocate capacity to features that do not generalize as well as the pretrained semantic representations. This can manifest as mild overfitting or interference during fusion, particularly under simple concatenation-based fusion where the model must learn when to trust character features and when to ignore them. In that sense, the hybrid may be "too eager" to exploit character signals, which improves robustness to perturbations but slightly harms clean generalization.

Methodologically, this also points to a limitation of the fusion strategy. Concatenation is simple and effective as a baseline, but it does not provide an explicit mechanism for conditional reliance on the character branch. Gating or attention-based fusion could allow the model to down-weight character features when inputs appear clean and to up-weight them when perturbations are present. A different direction is to avoid explicit fusion altogether by using tokenization-free or tokenization-minimized pretrained models, such as ByT5 and CANINE (J. H. Clark et al., 2021; Xue et al., 2021), which aim to reduce tokenization brittleness at the encoder level. Character-aware pretraining approaches (e.g., CharBERT) (El Boukkouri et al., 2020) are another alternative that integrates character sensitivity into the pretrained backbone rather than adding a separate branch. These approaches may improve robustness without introducing a second pathway that must be carefully controlled during fine-tuning.

Limitations

This study has several limitations that should be considered when interpreting results. First, we evaluate two datasets that represent multilingual and code-mixed regimes, but robustness trends may differ under other domains (e.g., short tweets vs long-form reviews) or other language distributions within Indonesia and Southeast Asia (Winata, Cahyawijaya, et al., 2024; Winata et al., 2023). Second, we use a single perturbation probability ($p=0.18$) and a fixed set of character-level operators. Robustness behavior can change with perturbation strength, operator composition, and whether noise targets particular character positions (e.g., vowels) versus uniformly random corruption. A broader sweep across perturbation rates would provide a more complete picture of where the hybrid begins to outperform the baseline decisively.

Third, we do not conduct large-scale hyperparameter sweeps over fusion strategies, character vocabulary design, or branch capacity. It is plausible that better-tuned fusion (e.g., gating) could reduce or eliminate the clean-performance penalty while preserving robustness gains. Finally, the hybrid design adds modest architectural complexity, and the optimal trade-off between engineering overhead and robustness improvement may depend on deployment constraints and the expected noise distribution in production data.

5. CONCLUSION

This paper studied a hybrid subword–character representation for sentiment classification on multilingual and code-mixed Indonesian text. Using two complementary benchmarks (NusaX Senti and Indonglish), we evaluated whether adding an explicit character-level branch can improve robustness to realistic orthographic variation while preserving the strong clean accuracy of multilingual Transformers. Across both datasets, a fine-tuned XLM-R baseline achieved the best clean Macro-F1, reinforcing that subword-based multilingual encoders remain highly competitive for standard evaluation. At the same time, robustness experiments reveal an important operational benefit of hybridization: under character-level perturbations, the XLM R+CharCNN model substantially reduced degradation, with the most pronounced gains on Indonglish. At $p=0.18$ noise, the Macro-F1 drop decreased from 0.030 to 0.007, indicating that character cues can stabilize predictions when surface form changes without altering semantic polarity. These results suggest a practical takeaway. If the target deployment environment is dominated by clean, well-formed text, a strong subword Transformer baseline may be sufficient. However, if the system must operate on informal, code-mixed inputs where

typos, elongation, and inconsistent spelling are common, hybrid models can provide meaningful robustness improvements even when clean accuracy gains are limited.

REFERENCES

Bojanowski, P., Grave, É., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics (TACL). https://doi.org/10.1162/tacl_a_00051

Clark, J. H., Garrette, D., Turc, I., & Wieting, J. (2021). Canine: Pre-training an efficient tokenization-free encoder for language representation. arXiv. https://doi.org/10.48550/arXiv.2103.06874

Clark, K., Luong, M.-T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. arXiv. https://doi.org/10.48550/arXiv.2003. 10555

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual represen tation learning at scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2020.acl-main.747 14

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT). https://doi.org/10.18653/v1/N19-1423

El Boukkouri, H., Ferres, J., Mamou, J., Hamdy, M., Boudoukh, G., Firooz, H., Kuan, L., & Stoyanov, V. (2020). Charbert: Character-aware pre-trained language model. arXiv. https://doi.org/10.48550/arXiv.2010.10392

Gugger, S., et al. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable. arXiv. https://doi.org/10.48550/arXiv.2205.07917

Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP). https://doi.org/10.18653/v1/D18-2012

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv. https://doi.org/10.48550/arXiv.1909.11942

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proceedings of ACL. https://doi.org/10.18653/v1/2020.acl-main.703

Lhoest, Q., Delangue, C., von Platen, P., Wolf, T., Salazar, J., et al. (2021). Datasets: A community library for natural language processing. arXiv. https://doi.org/10.48550/arXiv.2109.12209

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv. https://doi.org/10.48550/arXiv.1907.11692

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. International Con ference on Learning Representations (ICLR).

Morris, J. X., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP). https://doi.org/10.18653/v1/2020.emnlp-demos.16

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems (NeurIPS).

Patwa, P., Aguilar, G., Kar, S., Solorio, T., & Das, A. (2020). Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval). https://doi.org/10.18653/v1/2020.semeval-1.164

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), 1–67.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS). https://doi.org/10.48550/arXiv.1706.03762

Wei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP). https://doi.org/10.18653/v1/D19-1670 15

Winata, G. I., Cahyawijaya, S., et al. (2024). Seacrowd: A multilingual multimodal data hub and benchmark suite for southeast asian languages. arXiv. https://doi.org/10.48550/ arXiv.2406.10118

Winata, G. I., Cahyawijaya, S., Lin, Z., Wicaksono, A. F. A., et al. (2023). Nusax: Benchmarking machine translation for southeast asian languages. arXiv. https://doi.org/10.48550/ arXiv.2305.12267

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2020). Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP). https://doi.org/10.18653/v1/2020.emnlp demos.6

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). Byt5: Towards a token-free future with pretrained byte-to-byte models. arXiv. https://doi.org/10.48550/arXiv.2105.13626

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. arXiv. https://doi.org/10.48550/arXiv.1906.08237