Vol. 11, No. 2, Oktober 2026, Page. 468-490
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

# Position and Language as Diagnostic Variables: A Controlled Invariance Test of Sentiment Classifiers on Indonesian-English Code-Switched Reviews

**Kennedy Lie[*1], Handri Santoso[2]**

[1,2]Information Technology Department, Faculty of Information Technology, Pradita University
Email: [1]kennedy@student.pradita.ac.id, [2]handri.santoso@pradita.ac.id

**Abstract**

Sentiment classifiers are usually judged by accuracy or F1-score, which cannot show whether a model reads the sentiment cue or relies on the surrounding text. Code-switched text, which mixes languages within a sentence, makes this harder to judge. This study proposes a controlled behavioural invariance test in which one sentiment cue is moved across three positions (beginning, middle, end) and written in two languages (Indonesian, English) in 272 reviews, giving 1,632 minimal-pair variants. Logistic Regression, Linear SVC, BERT, mBERT, XLM-RoBERTa and IndoBERT were compared using Prediction Consistency Rate, Prediction Flip Rate and McNemar tests with Holm correction, alongside standard metrics. The two TF-IDF models were largely insensitive to cue position and language but still flipped 5.1% and 1.5% of predictions, and deleting the frequent token “aplikasi” changed their output margins nearly as much as deleting the sentiment cue. BERT and mBERT made more errors on Indonesian than on English cues (adjusted p = 0.014 and 0.031). XLM-RoBERTa was unaffected by the token on the reported split and, over five repeated splits, had the highest mean accuracy and PCR and the lowest PFR. On this dataset, strong standard metrics did not guarantee consistent predictions when the cue was moved or translated.

**Keywords:** sentiment analysis, code-switched data, behavioral invariance testing, deep learning transformers, multilingual transformers

***This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)***

## 1. INTRODUCTION

Sentiment analysis classifies opinions, attitudes and emotions in any form of digital text [1]. It has been widely applied to monolingual social media data and tweet-based classification tasks. In code-switched texts, however, it is harder to trust as speakers incorporate two or more languages within or between sentences [2]. This is commonly seen in modern digital communications like informal online interactions, online reviews, and social media posts [3]. Code-switching is a challenge because a single sentence can contain vocabulary, grammar, and meaning from more than one language [4]. Take this example, "*Tempatnya sangat comfortable dan suasananya very modern*". It combined both Indonesian context with English sentiment words. The sentiment is positive but the sentence used two languages, so a classifier has to handle sentiment words from two languages within one sentence.

Most research on code-switched natural language processing (NLP) has moved from feature engineering and language identification toward contextual embedding and transformer-based classification, aimed to improve raw classification performance [4], [5]. Winata et al. show pretrained multilingual transformer models do not automatically guarantee high performance even with embeddings [6]. Shared tasks and benchmarks such as GLUECoS and SentiMix 2020 have shown multilingual transformer models are best suited for code-switching texts [7], [8]. Chanda et al. found that adding language identification improved classification performance on code-switched comments [9]. Yulianti et al., Astuti et al., and Hidayatullah et al. all agree Indonesian-English code-switching is an active research problem where model performance depended on the dataset design [3], [10], [11]. But they all only evaluate on naturally occurring instances.

Fewer studies ask whether specific, sentence-internal properties of code-switched text can expose how a sentiment classifier actually works, rather than treating the mixed-language input as a whole. Code-switching is not only about two languages or more mixed in a sentence. Poplack and Sankoff and Poplack showed intra-sentential switching follows grammatical constraints rather than occurring at random [2], [12]. Myers-Scotton's Matrix Language Frame split the languages by role, one that supplied

the sentence's grammatical frame and the language that supplied the embedded elements [13]. These studies did not test classifiers but informed that position of a switched element is not random but has meaning.

Naturally occurring data has real limits for evaluating models. That is part of why NLP research has been moving toward behavioral and diagnostic probing instead. Ribeiro et al.'s CheckList framework tests specific linguistic capabilities by perturbing the input in targeted ways. It changes a sentence without altering its label then checks whether the prediction stays the same [14]. HANS (Heuristic Analysis for NLI Systems) uses a similar idea to test whether models rely on syntactic heuristics in monolingual pipelines, but this kind of testing has rarely been applied to intra-sentential code-switching [15]. Clark et al. offer a plausible mechanism for this kind of heuristic reliance, finding that BERT's attention heads often default to a dominant, low-information background token rather than the actual sentiment cue [16]. An invariance-testing protocol was applied to determine whether structural changes flip the model's prediction, exposing either shortcut reliance or syntactic instability.

To our knowledge, the work cited above does not test whether a sentiment classifier keeps its prediction when the same sentiment cue is moved within an Indonesian-English code-switched sentence or expressed in the other language. If switching points are bound by grammar, manipulating their position lets us test whether a transformer model actually parses the sentence or just exploits localized vocabulary shortcuts to predict the sentiment label. Transformer-based models also rely on positional encoding to track word order [17]. Two code-switched sentences can share the same words and sentiment label, and still be treated differently when the sentiment-bearing expression changes position. Language is the second variable. The sentiment in Indonesian-English code-switched text can be in Indonesian or English, or both. Examples such as "*The place was nyaman*" and "*Tempatnya comfortable*" share the same positive sentiment but appear in different languages. A multilingual model may process these sentiment words differently. Its vocabulary coverage, pretraining data, or embedding quality can all differ when processing words in different languages [18], [19]. One reason is because English is a high-resource language while Indonesian and most Indonesian local languages remain low-resource, despite Indonesia's population being among the largest in the world [20], [21]. Both factors raised a direct question: "Can a model still classify a statement correctly when the sentiment cue is repositioned to the beginning, middle, or end of the sentence or when it is expressed in a different language? Or will the model fall back on a familiar template and lean on the surrounding background text instead?".

The present study treats Indonesian-English code-switched sentiment analysis as a controlled behavioral invariance probe to address this question. We manipulate two variables: the position of the sentiment cue (beginning, middle, end) and the language carrying the sentiment (Indonesian or English), building a controlled experimental framework in which code-switched variants are constructed from a base dataset. These variants are not just simple additions to the data. We treat them as controlled properties designed to test whether the classifier actually process changes to the statement or rely on dominant background tokens. Each variant is checked for grammatical acceptability and sentiment label consistency before evaluation. Lastly, we introduce invariance-based behavioral metrics to support this design. These help check how consistent a model's predictions are across variations of the same base review, which lets us compare how sensitive each model is to the position and the language of the sentiment cue.

This study makes three contributions. First, it provides a controlled invariance test in which one sentiment cue is held fixed and moved across three positions and two languages within the same Indonesian-English review, with Prediction Consistency Rate and Prediction Flip Rate as measures of how stable a prediction is across the six variants. Second, it reports how six classifiers, two TF-IDF models and four transformers, respond to cue position and language, with paired tests corrected within each model. Third, it adds intervention tests on one frequent background token to check whether the observed errors are consistent with reliance on that token.

### 1.1. Related Work

Code-switched NLP has mostly been evaluated with benchmarks that report aggregate scores on held-out test sets. LinCE collects code-switching datasets for tasks such as language identification, part-of-speech tagging, named-entity recognition and sentiment analysis in one benchmark [22], and

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

GLUECoS does so across several tasks [7]. Work on Indonesian follows the same pattern. IndoNLU provides a benchmark for Indonesian understanding tasks, including sentiment analysis [23], IndoBERTweet is a language model pretrained on Indonesian Twitter [24], and NusaX offers a sentiment dataset that is parallel across Indonesian local languages [21]. These resources let models be compared by accuracy or F1. NusaX makes languages comparable because each sentence is translated as a whole, but none of them holds one sentiment cue fixed and moves it inside a code-switched sentence, so they cannot say whether a prediction depends on where the cue is or which language carries it.

Behavioural testing addresses this kind of question for monolingual text. CheckList tests specific capabilities with targeted perturbations, among them invariance tests in which a prediction should not change [14]. M2C carries the idea across languages by generating morphology-aware tests, which it applies to question answering in 12 typologically diverse languages [25]. Contrast sets and counterfactually augmented data use small edits as well, but the edits change the gold label, in order to probe the local decision boundary of a model [26] or to reduce its reliance on spurious features [27]. HANS, in contrast, builds test sets in which a syntactic heuristic and the correct label disagree [15]. Our test follows the invariance setting, since the label of a review does not change when the cue moves or is translated, but it works within code-switched Indonesian-English reviews and compares the prediction for six variants of the same review.

Several studies have already examined how code-mixed input degrades models. Tan and Joty generate code-mixed adversarial examples by swapping words or phrases with their translations and evaluate them on natural language inference and question answering [28]. AdvCodeMix attacks sentiment classifiers for Bengali-English and Hindi-English text with token-level perturbations ranked by importance [29]. For Indonesian, IndoRobusta replaces the words that matter most to a model with their translations in English, Sundanese, Javanese or Malay and reports the drop in accuracy on sentiment and emotion classification [30]. A recent paired study of hate-speech moderation compares the same content written in English and in Tamil-English and reports how often the decision flips, with McNemar tests [31]. These studies perturb many words at once, or choose the perturbed words by model importance or translation alignment, and they report a drop in performance. They do not keep a single cue constant in meaning and label while varying only its position and its language, so they cannot attribute a change in prediction to one of those two factors.

Finally, our analysis of why predictions change draws on the literature on shortcut learning, which describes decision rules that work on a benchmark but fail under other conditions [32]. Attention weights have been used to look for such rules in BERT [16], but whether attention explains a prediction is disputed [33], [34]. For this reason we treat attention as one piece of evidence and rely mainly on interventions on the input, namely deletion and replacement of a token with matched controls, together with Integrated Gradients attributions.

In summary, existing work covers code-switched benchmarks, label-flipping and label-preserving behavioural tests in monolingual or cross-lingual settings, and robustness of code-mixed classifiers to perturbation. The setting studied here, a fixed sentiment cue that changes only in position and language within the same Indonesian-English review, with prediction consistency measured across the six variants, is not covered by these studies, and Table 1 summarizes the differences.

Table 1. Comparison with related perturbation and behavioural-testing studies

| Study | Task and Languages | What is changed in the input | Main Measure | Holds one cue fixed and varies only its position and language | Analysis of a specific background token |
|---|---|---|---|---|---|
| CheckList [14] | Several tasks, English | Targeted template-based perturbations, including invariance tests | Failure rate per test | No | No |

| | | | | | |
|---|---|---|---|---|---|
| M2C [25] | Question answering, 12 languages | Morphology-aware test templates | Performance per language-specific test | No | No |
| Contrast sets [26] | Several tasks, mainly English | Small manual edits that change the label | Accuracy and contrast consistency | No | No |
| Counterfactually augmented data [27] | Sentiment and NLI, English | Human edits that flip the label | Accuracy of trained classifiers | No | No |
| Tan and Joty [28] | NLI and question answering, English mixed with other languages | Word- and phrase-level code-mixing by translation | Accuracy drop, attack success rate | No | No |
| AdvCodeMix [29] | Sentiment, Bengali-English and Hindi-English | Token-level perturbations ranked by importance | F1 drop | No | No |
| IndoRobusta [30] | Sentiment and emotion, Indonesian mixed with English, Sundanese, Javanese, Malay | Top-ranked words replaced by translations | Accuracy and accuracy drop | No | No |
| Paired moderation study [31] | Hate-speech moderation, English vs Tamil-English | Same content in different language views | Decision flip rate, McNemar test | No | No |
| This study | Sentiment, Indonesian-English reviews | One cue moved to three positions and written in two languages (six variants per review) | PCR, PFR, McNemar with Holm correction | Yes | Yes (deletion and replacement with controls, Integrated Gradients, attention on one token) |

## 2. METHODOLOGY

### 2.1. Research design

This study is designed to test whether sentiment classifiers process the actual sentiment cue or rely on surrounding contextual shortcuts. This is done by manually constructing variations of code-switched text instead of using naturally occurring code-switched data. Position (Beginning, Middle, End) and language of the sentiment cue (Indonesian or English) are manipulated to create the variants. Sentiment classification performance and prediction consistency are used for evaluation. Prediction consistency is measured using invariance-based behavioral metrics (Prediction Flip Rate and Prediction Consistency Rate) for structural stability and analyzed through condition-based analysis. The final dataset used in this study is based on (https://www.kaggle.com/datasets/sasyablossom/tiket-com-app-sentiment-analysis-reviews).

A strict annotation guideline was established for the purpose of variant construction. Only sentences with a single dominant sentiment-bearing expression were chosen. Other sentences with multiple or repeated sentiment cues, sarcasm, irony, or complex multi-clause complaints were excluded. Failure related statements were retained as long as a single dominant cue indicated inability, malfunction, unsuccessful completion, or missing functionality (e.g. “gagal”, “tidak bisa (X)”, or “tidak tersedia”). Interrogative complaints were normalized into declarative forms. If Emojis are the only sentiment cue, they were normalized to its equivalent expression. Those who only reinforced the existing sentiment were removed.

During variant construction, the overall meaning and polarity of the statement were not changed. Only the target sentiment cue was repositioned and translated when required. Minimal adjustments to grammar or punctuations were allowed when needed to create a natural and acceptable variant. This kept position and language as the only manipulated variable. Any instability of a model's predictions can be attributed to these structural factors instead of other linguistic noise. Figure 1 illustrates the framework used for the construction of the controlled dataset. Detailed description of the dataset construction phase is explained in the following sections.

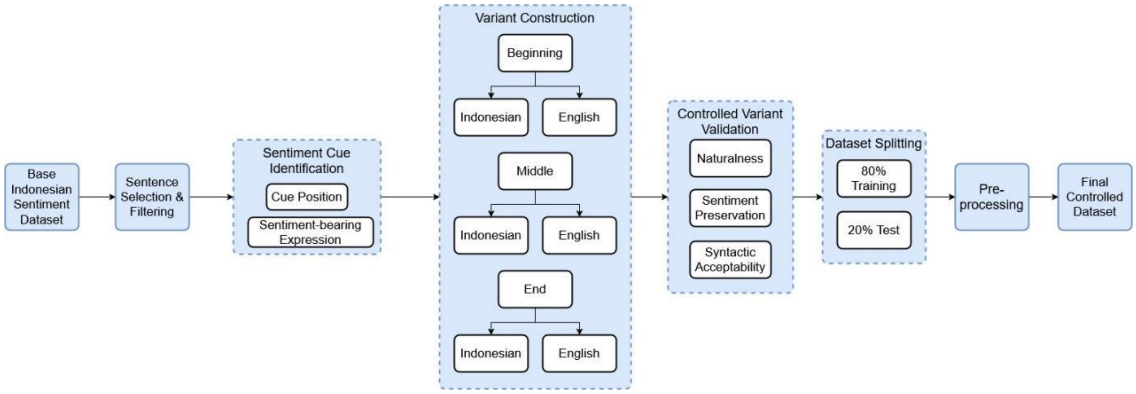



Figure 1. Controlled dataset construction framework

Table 2. Variation example

| Text | Position | Language |
|---|---|---|
| Sangat mudah pesan tiket pesawat dr sini | Beginning | Indonesian |
| pesan tiket pesawat Sangat mudah dr sini | Middle | Indonesian |
| pesan tiket pesawat dr sini Sangat mudah | End | Indonesian |
| Very easy pesan tiket pesawat dr sini | Beginning | English |
| pesan tiket pesawat Very easy dr sini | Middle | English |
| pesan tiket pesawat dr sini Very easy | End | English |

## 2.2. Dataset variant construction

This dataset consists of 72,000 user reviews of the Tiket.com app obtained from the Indonesian Google Play app store spanning 2014 to early 2026. The reviews are pre-labelled with sentiments across three spectrums: Positive, Negative and Neutral. The dataset suited the experiment design due to its volume and mix of short to medium length reviews, allowing clean single-cue variants construction.

Before constructing the variations, the dataset was filtered by removing duplicate reviews and neutral-labeled reviews. Neutral instances were excluded because of their ambiguity and limited polarity information than clearly defined positive or negative expressions [35]. Sentences were filtered to only include statements with 6-12 words to reduce the likelihood of multiple sentiment-bearing expressions. The filtered dataset was manually reviewed to obtain a final subset of 281 samples (152 Positives and 129 Negatives) based on the established guidelines.

The subset is intentionally smaller than the full dataset because the purpose is not improved model performance. It is to create a dataset that can be modified using the proposed variables of position and language. Prior research on sample size in machine learning shows that adequacy is not solely based on input count. Data quality, effect size, and whether performance stabilizes as sample size increases are

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

also determining factors [36]. Only the text of each review and its sentiment label were used. Usernames, profile pictures and other account information are not part of the final dataset of variants. The identifier reviewId used in this study is a randomly generated UUID that groups the six variants of one review. A pattern-based check of the reviews for e-mail addresses, phone numbers, URLs, @-handles and numbers of eight or more digits found no matches.

The variant construction process is divided to three steps. First, the sentiment-bearing word is checked for each review. Second, the sentiment-bearing word is translated to its code-switched variant (Indonesian to English and vice versa) by the first author. Third, the variants are then constructed from the translated original statement, producing 6 variants including the source. Each review for each language has 3 variants ranging from sentiment at the beginning, middle, and end of the sentence. The constructed dataset now consisted of the original text and the code-switched text, each with different sentiment bearing word positions. Table 2 shows an example of variation made from a randomly sampled positive sentiment statement.

### 2.3. Dataset validation

Because variations in the dataset are constructed manually, validation is conducted to ensure acceptable dataset quality. External bilingual annotators were not available for this study and instead, a single reviewer manual audit by the first author was conducted. The purpose is to reduce construction bias by ensuring each variant is evaluated thoroughly using a set of criteria. The following criteria are as follows:

- **Naturalness**: a five-point scale rating (1 is highly unnatural, 5 is very natural) of how acceptable the variant is in informal code-switched text
- **Sentiment preservation**: evaluated with Yes or No, whether the variant keeps the same sentiment polarity and intensity as the source statement
- **Syntactic acceptability**: evaluated with Yes or No, whether word order and clause structure remain acceptable after repositioning of the chosen sentiment cue

Variants that did not preserve the sentiment intensity or polarity are rejected outright because it would produce a different construct than the controlled minimal pair this study requires. Variants that are not syntactically acceptable or scored a naturalness below 3 are revisable and are re-scored. If after revision the variant is still deemed unnatural, it is rejected. Variants that sound syntactically forced, the meaning of the sentence is ambiguous due to repositioning, or its original meaning was altered were rejected. Only variants which scored a naturalness of 3 and above, the sentiment intensity and polarity is preserved, and syntactically acceptable are accepted and used to train the model. An example of this audit can be found in Table 3 and the decision process can be found in Figure 2.

Table 3. Example of statement variant audit

| Variant | Naturalness | Sentiment Preserved? | Syntactically Acceptable? | Decision | Reason |
|---|---|---|---|---|---|
| sgala kebutuhan Sangat mudah tuk cari | 2 | Yes | Yes | Reject | sentence is ambiguous in meaning |
| very easy tuk cari sgala kebutuhan | 4 | Yes | Yes | Accept | understandable in casual conversations |

Vol. 11, No. 2, Oktober 2026, Page. 468-490
https://journal.uin-alauddin.ac.id/index.php/instek
p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

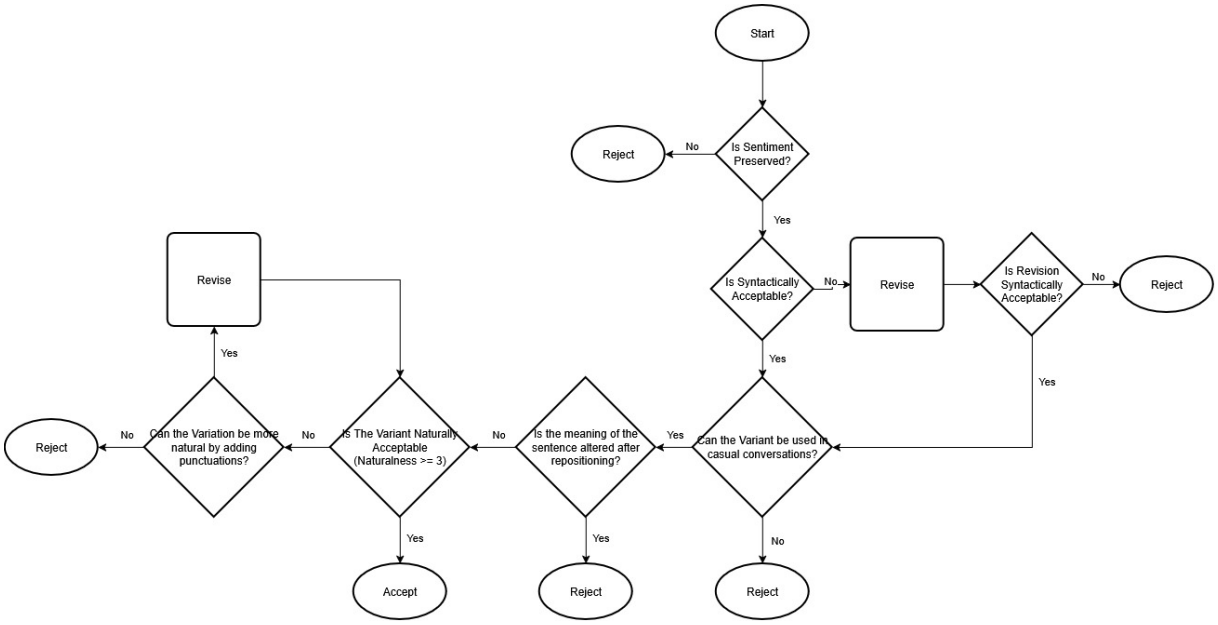



Figure 2. Variant acceptance decision flowchart

### 2.4. Dataset splitting

After validation, 272 of the 281 reviews remained, because nine reviews were removed due to the unnaturalness of their variants. The final dataset therefore contains 1,632 variants (876 Positive and 756 Negative). It was split 80:20 into training and testing using scikit-learn's '*GroupShuffleSplit*' based on reviewId so that all six variants of a review stay in the same split. The parameters used are n_splits=1 and random_state=42. Table 4 gives the full distribution of the dataset.

### 2.5. Preprocessing

All text in the final dataset were preprocessed before feature extraction. For traditional machine learning models, text was lowercased before TF-IDF vectorization. For transformer models, casing was handled by each tokenizer's default behavior. Non-sentiment-related numbers such as order IDs, booking references, tracking numbers, and invoice numbers were removed as they are unrelated to sentiment. Remaining text was cleaned using standard methods such as removal of repeated characters and punctuation (e.g., "*membantuuuuu*" to "*membantu*"), reduplication and closing-quotation-mark formatting were normalized (e.g., *lain2* or *lain"* to "*lain-lain*"), and non-ASCII characters and excess whitespace were removed.

Table 4. Distribution of review groups and variants

| Split | Label | Review groups | Variants | Beginning | Middle | End | Indonesian | English |
|---|---|---|---|---|---|---|---|---|
| Train | Negative | 101 | 606 | 202 | 202 | 202 | 303 | 303 |
| Train | Positive | 116 | 696 | 232 | 232 | 232 | 348 | 348 |
| Train | Total | 217 | 1,302 | 434 | 434 | 434 | 651 | 651 |
| Test | Negative | 25 | 150 | 50 | 50 | 50 | 75 | 75 |
| Test | Positive | 30 | 180 | 60 | 60 | 60 | 90 | 90 |
| Test | Total | 55 | 330 | 110 | 110 | 110 | 165 | 165 |
| All | Negative | 126 | 756 | 252 | 252 | 252 | 378 | 378 |
| All | Positive | 146 | 876 | 292 | 292 | 292 | 438 | 438 |
| All | Total | 272 | 1,632 | 544 | 544 | 544 | 816 | 816 |

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

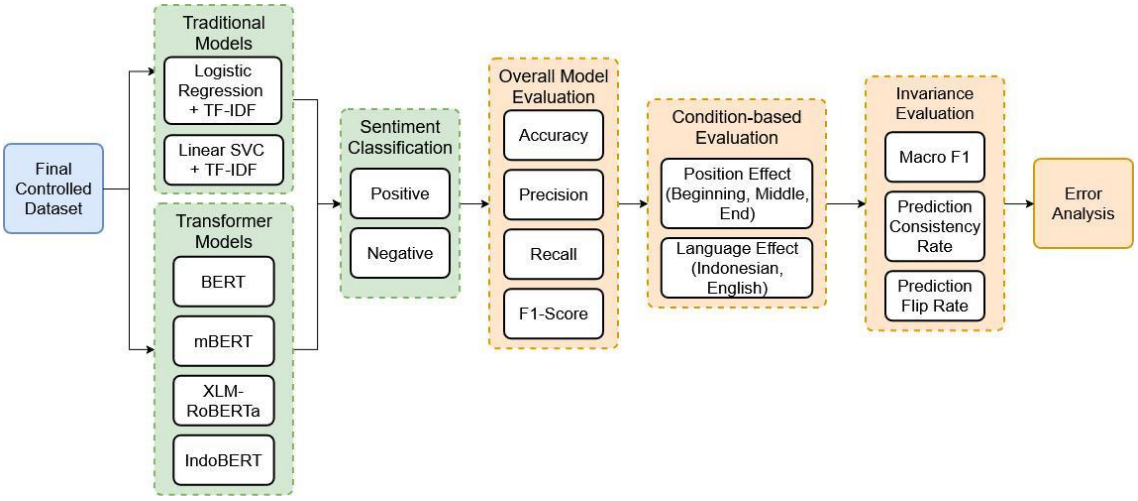



Figure 3. Experiment and analysis framework

Features of traditional machine learning models were extracted using TF-IDF (Term Frequency-Inverse Document Frequency) which captures term importance while reducing the influence of frequently occurring words [37]. For transformer-based models, tokenization was done using each transformer's tokenizer.

Given the importance of contextual and positional information in this research, stopword removal is not applied in the preprocessing pipeline. This decision was made to preserve natural linguistic structure and avoid unintended distortion of sentiment. Specifically in Indonesian-English code-switched text, function words may carry grammatical information that matters in code-switched text [13]. Punctuation was kept to preserve naturalness and syntactic structure. Prior research has shown that punctuation affects the readability and understanding of texts for both humans and machines [38].

### 2.6. Experimental setup

This subsection describes the model training stage, comparing traditional machine learning baselines and transformer-based models under the same controlled test variants. Figure 3 illustrates the flow of the experiment until evaluation and analysis. A detailed description of the experiment and analysis phase is given in the following sections.

The traditional machine learning approaches used in this study are Logistic Regression and Linear Support Vector Classification (Linear SVC). Logistic regression is a suitable baseline for text categorization because it is simple to train and can perform competitively with more complex classifiers such as SVM [39]. Linear SVC is included because support vector machines were shown to outperform other classical classifiers on TF-IDF-weighted text features in text categorization [40]. These two baselines allow the study to determine whether structural sensitivity in a code-switched sentiment classification can already be observed using non-contextual lexical features before applying transformer-based representations.

Logistic Regression was trained with the hyperparameters set to max_iter=1000, C=1.0, and solver='lbfgs' while Linear SVC was set with max_iter=1000, C=1.0, and class_weight: 'balanced' Both models uses the TF-IDF with hyperparameters set to ngram_range=(1,2), min_df=2, max_df=0.95, and stop_words=none.

For transformer-based models, classifiers including BERT, mBERT, XLM-RoBERTa, and IndoBERT are fine-tuned for binary sentiment classification. BERT serves as the general baseline for transformers because it uses bidirectional contextual representations and can be fine-tuned for downstream classification tasks by adding a task-specific output layer [41]. mBERT and XLM-RoBERTa both serve as multilingual baselines. mBERT learns representations across many languages and has been shown to support cross-lingual transfer and is suitable for testing whether a general multilingual model can handle Indonesian-English code-switched sentiment cues consistently [42]. XLM-RoBERTa is the stronger multilingual transformer because it was pretrained at a larger multilingual scale and has been shown to improve cross-lingual transfer performance compared to earlier multilingual models [43]. IndoBERT acts as the Indonesian-specific transformer because it was

developed for Indonesian NLP and evaluated on Indonesian-language tasks [44]. Since the base reviews are Indonesian, IndoBERT provides an important comparison against multilingual models. Recent work on Indonesian-English code-switched sentiment analysis using transformer models also supports the relevance of transformer-based approaches for this language-pair [11].

All four transformer models (BERT: bert-base-uncased, mBERT: bert-base-multilingual-uncased, XLM-RoBERTa: xlm-roberta-base, IndoBERT: indolem/indobert-base-uncased) were fine-tuned with identical hyperparameters: max_length=60, learning_rate=1e-5, batch_size=16, epochs=3, and the AdamW optimizer with weight decay 0.01, gradient clipping at a norm of 1.0 and no learning-rate scheduler. The random seed was set to 42 for Python, NumPy and PyTorch. The models are otherwise used in their default settings. Input text was left with its original casing intact for all models and each model's own tokenizer handled casing according to its default behavior, lowercasing the text for the uncased BERT variants while leaving it unchanged for XLM-RoBERTa. A global seed of 42 was set once at the start of the session, and all models were trained sequentially in a single run; exact replication therefore requires executing the notebook from top to bottom in the released order. The train-test split is fixed independently by the split seed of 42.

No validation set was held out and no hyperparameters were tuned. To check whether this setting left the models underfitted, we recorded the average training loss per epoch. The loss fell from 0.546 to 0.033 for BERT, from 0.339 to 0.010 for mBERT, from 0.508 to 0.035 for XLM-RoBERTa and from 0.600 to 0.108 for IndoBERT between the first and the third epoch. The models therefore fitted the training data, and IndoBERT, whose final loss was the highest, converged more slowly than the other three. A low training loss does not show that these hyperparameters are optimal. As a sensitivity check on IndoBERT, whose final loss was the highest, we fine-tuned it once more with a learning rate of 3e-5 and five epochs (seed reset to 42, all other settings unchanged) and evaluated it on the same test split. Accuracy was 0.96, PCR rose from 0.8727 to 0.9091 and PFR fell from 0.0473 to 0.0436. This was one additional configuration, not a tuning procedure, and the results in Tables 5–14 keep the pre-specified setting.

The 1,632 constructed variants, the notebook and the per-variant predictions of all six models are available from github (Github Link). Experiments were run with Python 3.13.16, scikit-learn 1.6.1, PyTorch 2.11.0, Transformers 5.20.2 and Captum 0.9.0.

### 2.7. Evaluation and analysis

Standard classification metrics of accuracy, precision, recall, and F1-score were captured for all models. F1-score is used as the primary measure since it balances precision and recall. Because these metrics cannot tell whether a model actually processes the sentiment cue or is focusing on the surrounding context, we pair them with invariance-based behavioral metrics to track prediction consistency through variations of the same review.

Condition-based analysis are reported separately by code-switching position and sentiment-bearing language. This evaluation is necessary because a model may achieve high overall performance while still showing weakness in specific code-switched condition. For example, a model may correctly classify English sentiment cues for the End variant but misclassify the Middle variant.

To check whether the model consistently predicts the label of each variant, this study adapts the Invariance Testing (INV) protocol from Ribeiro et al. [14]. This framework is extended into two separate scales. At the variant level, Prediction Flip Rate counts how many predictions flip from the original statement. At the group level, Prediction Consistency Rate measures whether predictions stay uniform across all six variants in the same group and is defined as:

$$PCR = \frac{Number\ of\ sentence\ groups\ with\ identical\ predictions\ across\ variants}{Total\ sentence\ groups} \quad (1)$$

A sentence group counts as consistent only if all six variants share the same prediction. The variant-level Prediction Flip Rate follows the formula used by Nguyen et al. and adapted as [45]:

$$PFR = \frac{Number\ of\ prediction\ changes}{Total\ number\ of\ matched\ pairs} \quad (2)$$

A high PCR together with a low PFR means that the prediction was not affected by the manipulated variable, which is not evidence of correct predictions because a model that is wrong in all six variants is still consistent. A low PCR together with a high PFR means that the prediction changed with the manipulation.

Error analysis on representative errors is conducted to identify possible causes. Weak recognition of Indonesian sentiment expressions, overreliance on English sentiment words, difficulty with specific positions, tokenization issues, or confusion from informal code-switched order are some example causes.

McNemar's test was applied for each pairwise comparison to check whether the condition-based error differences across position and language are statistically meaningful or are simply sampling noise [46]. Pairs are matched based on reviewId. Position comparisons are combined with language while language comparisons are combined with position. Testing several comparisons at once raises the risk of false positives. For each model, Holm's sequential correction is applied across its own family of four comparisons (three position pairs and one language pair) to control the familywise error rate [47]. The correction is therefore applied per model and not across the six models together.

### 2.8. Evidence tests for background-token reliance

The invariance metrics show whether predictions change with the manipulated variables, but not why a model errs. To test whether a recurring background word acts as a shortcut, four analyses were run on the same trained models and the same test split. The candidate words are words that recur in the misclassified examples of the error analysis. Because they were chosen after inspecting the errors, the analysis is post hoc.

First, token statistics. For each candidate word we count the training reviews that contain it and the share of them that are positive, and test this share against the base rate of positive reviews in the training set with Fisher's exact test on review-level counts, with Holm correction over the candidate words. We also report the Logistic Regression and Linear SVC coefficients of the word and its rank among all features.

Second, intervention tests. For every test sentence that contains the word, we delete it and record whether the prediction flips. Two controls are used: deleting one randomly chosen other word (five draws per sentence) and deleting the sentiment cue itself. We also record the change in the positive-class margin (the logit difference for the transformers and the decision function for the linear models) and report its mean absolute change under each edit. As a check, the word was also replaced with a neutral filler word ("*hal*", "*proses*", "*layanan*"). The location of the cue is exact for the Indonesian variants and is estimated by word-level comparison with the Indonesian variants for the English variants.

Third, attention controls. The attention ratio is the attention received by the word, averaged over layers, heads and query positions, divided by the mean attention received by the other non-special tokens. We compute it for all test sentences that contain the word, separately for correctly and incorrectly classified sentences. Because a word may be split into several word pieces, we also report the ratio per word piece and the ratio relative to the attention received by the sentiment cue in the same sentence.

Fourth, Integrated Gradients, with the positive-class margin as the target, an all-padding baseline and the special tokens kept in place. For each set of tokens we report its share of the total absolute attribution divided by the share expected from an even spread, and its signed contribution toward the true label. Attributions should sum to the difference between the margin of the input and the margin of the baseline, and we report the relative completeness error (the absolute deviation divided by the absolute margin difference). BERT and IndoBERT used 32 interpolation steps. For mBERT and XLM-RoBERTa the number of steps was increased per sentence (64, 128, 256) until the deviation was below 10% of the total absolute attribution, because the relative error is unstable when the margin difference is small, and rows that stayed above 20% were excluded.

All intervals are 95% cluster-bootstrap intervals (2,000 resamples) over reviewId, because the six variants of a review are not independent observations.

## 3. RESULTS AND DISCUSSION

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

### 3.1. Experiment results

Overall model metrics in Table 5 showed XLM-RoBERTa achieved the best performance out of the six models, with a perfect accuracy and macro F1-score of 1.00. Table 6 and Table 7 show this score does not change even in variants, maintaining the same performance across all position and language conditions. These condition-based numbers are not an independent finding, they follow directly from the zero-error result in Table 5. A perfect score on this dataset also meant perfect scores for condition-based groups since the test set already contains all six variants of a sentence group. It is reported for completeness and to allow a direct comparison across all six models.

XLM-RoBERTa produced no misclassifications so its results cannot be analyzed like the other evaluated models. This opens a question whether its invariance is from the model correctly processing the sentiment cue of the variations or the limited template diversity of the 330 samples test set. To answer that required evaluating XLM-RoBERTa on a larger or more naturally varied code-switched test set.

Table 5. Overall model score

| Model | | Precision | Recall | F1-score | Support |
|---|---|---|---|---|---|
| Logistic Regression | Negative | 0.96 | 0.87 | 0.91 | 150 |
| | Positive | 0.90 | 0.97 | 0.93 | 180 |
| | Accuracy | | | 0.92 | |
| Linear SVC | Negative | 0.96 | 0.89 | 0.92 | 150 |
| | Positive | 0.92 | 0.97 | 0.94 | 180 |
| | Accuracy | | | 0.93 | |
| BERT | Negative | 0.94 | 0.92 | 0.93 | 150 |
| | Positive | 0.93 | 0.95 | 0.94 | 180 |
| | Accuracy | | | 0.94 | |
| mBERT | Negative | 0.97 | 0.97 | 0.97 | 150 |
| | Positive | 0.98 | 0.98 | 0.98 | 180 |
| | Accuracy | | | 0.98 | |
| XLM-RoBERTa | Negative | 1.00 | 1.00 | 1.00 | 150 |
| | Positive | 1.00 | 1.00 | 1.00 | 180 |
| | Accuracy | | | 1.00 | |
| IndoBERT | Negative | 0.97 | 0.93 | 0.95 | 150 |
| | Positive | 0.94 | 0.98 | 0.96 | 180 |
| | Accuracy | | | 0.95 | |

Table 6. Positional result of each model

| Model | Position | Total Samples | Total Errors | Error Rate | Macro F1 |
|---|---|---|---|---|---|
| Logistic Regression | Beginning | 110 | 8 | 0.073 | 0.93 |
| | Middle | 110 | 9 | 0.082 | 0.92 |
| | End | 110 | 9 | 0.082 | 0.92 |
| Linear SVC | Beginning | 110 | 7 | 0.064 | 0.94 |
| | Middle | 110 | 8 | 0.073 | 0.93 |
| | End | 110 | 7 | 0.064 | 0.94 |
| BERT | Beginning | 110 | 6 | 0.055 | 0.94 |
| | Middle | 110 | 8 | 0.073 | 0.93 |
| | End | 110 | 7 | 0.064 | 0.94 |
| mBERT | Beginning | 110 | 2 | 0.018 | 0.98 |
| | Middle | 110 | 2 | 0.018 | 0.98 |
| | End | 110 | 4 | 0.036 | 0.96 |
| XLM-RoBERTa | Beginning | 110 | 0 | 0.000 | 1.00 |
| | Middle | 110 | 0 | 0.000 | 1.00 |
| | End | 110 | 0 | 0.000 | 1.00 |
| IndoBERT | Beginning | 110 | 5 | 0.045 | 0.95 |
| | Middle | 110 | 4 | 0.036 | 0.96 |
| | End | 110 | 6 | 0.055 | 0.94 |

Table 7. Language Result of each model

| Model | Language | Total Samples | Total Errors | Error Rate | Macro F1 |
|---|---|---|---|---|---|
| Logistic Regression | English | 165 | 14 | 0.085 | 0.91 |
| | Indonesian | 165 | 12 | 0.073 | 0.93 |
| Linear SVC | English | 165 | 13 | 0.079 | 0.92 |
| | Indonesian | 165 | 9 | 0.055 | 0.94 |
| BERT | English | 165 | 5 | 0.030 | 0.97 |
| | Indonesian | 165 | 16 | 0.097 | 0.90 |
| mBERT | English | 165 | 0 | 0.000 | 1.00 |
| | Indonesian | 165 | 8 | 0.048 | 0.95 |
| XLM-RoBERTa | English | 165 | 0 | 0.000 | 1.00 |
| | Indonesian | 165 | 0 | 0.000 | 1.00 |
| IndoBERT | English | 165 | 8 | 0.048 | 0.95 |
| | Indonesian | 165 | 7 | 0.042 | 0.96 |

Table 8. Prediction consistency rate and prediction flip rate of models

| Model | Prediction Consistency Rate | Prediction Flip Rate |
|---|---|---|
| Logistic Regression | 0.9091 | 0.0509 |
| Linear SVC | 0.9636 | 0.0145 |
| BERT | 0.9091 | 0.0473 |
| mBERT | 0.9273 | 0.0436 |
| XLM-RoBERTa | 1.0000 | 0.0000 |
| IndoBERT | 0.8727 | 0.0473 |

Table 9. Shared misclassification of one review (Each model predicted wrong at least one variant)

| Review | Position | Language | True Label | Predicted |
|---|---|---|---|---|
| Cukup membingungkan aplikasi ini awal awal y. | Beginning | Indonesian | Negative | Positive |
| aplikasi ini Cukup membingungkan awal awal y. | Middle | Indonesian | Negative | Positive |
| aplikasi ini awal awal Cukup membingungkan. | End | Indonesian | Negative | Positive |
| Quite confusing aplikasi ini awal awal y. | Beginning | English | Negative | Positive |
| aplikasi ini Quite confusing awal awal y. | Middle | English | Negative | Positive |
| aplikasi ini awal awal Quite confusing. | End | English | Negative | Positive |

The errors are concentrated in few reviews. Thirteen distinct reviews account for all 92 misclassified variants across the five models that made errors, and two reviews account for 41 of them (45%). The number of distinct reviews behind each model's errors is 7 for Logistic Regression (26 variants), 5 for Linear SVC (22), 6 for BERT (21), 4 for mBERT (8) and 7 for IndoBERT (15). Because one review contributes up to six variants, the examples in Tables 9 to 13 are descriptive and cannot be treated as independent observations.

McNemar's test was applied to each pairwise position and language comparison per model, with Holm correction applied within each model's family of four comparisons. None of the position-based comparisons is statistically significant. For language, the difference between Indonesian and English cues is significant for BERT (adjusted $p = 0.0137$) and mBERT (adjusted $p = 0.0313$), and not for the other models. In Table 7 both BERT (16 errors on Indonesian cues against 5 on English) and mBERT (8 against 0) show a pronounced drop on Indonesian-language sentiment cues.

Table 8 reports the Prediction Consistency Rate and Prediction Flip Rate of each model. XLM-RoBERTa achieved a perfect Prediction Consistency Rate of 1.0000 and a Prediction Flip Rate of 0.0000. This confirmed its zero-error result in Table 5 extends to complete structural invariance. Out of the five models, Linear SVC has the highest consistency rate of 0.9636 and the lowest flip rate of 0.0145. IndoBERT has the lowest consistency rate of 0.8727 and Logistic Regression has the highest flip rate

of 0.0509. A high consistency rate is not a sign of better performance. It only measures how invariant the model is under mutation. If the model predicted an incorrect label for all variants, it is still considered consistent.

The Prediction Flip Rate checks whether the model is actually classifying the variant differently from the original statement. The higher the flip rate, the more variants differ in prediction from the original statement, which does not by itself show whether those predictions are correct. The review in Table 9 is a direct example. Logistic Regression and Linear SVC predicted positive for all six variants of this review, including the original. This group therefore contributed full consistency and zero flips for these two models, even though the predictions are wrong. mBERT's predictions for the same review, discussed later in the section, were not uniform across its variants.

By design, the established guidelines created a controlled dataset with little template variance which helped isolate the variables to modify. Some examples being "*aplikasi yang sangat membantu dalam pemesanan tiket pesawat dan hotel*" and "*Aplikasinya sangat membantu saya mencari hotel terbaik*". Because of this specific design, a small set of recurring domain tokens like "*aplikasi*" is present in many reviews. One specific sample serves as a useful case to compare how different each model handled the same gap in training coverage. The review shown in Table 9 is misclassified at least for one variant by all the models except XLM-RoBERTa.

The review structure was not included in the training as a result of the train-test split. Logistic Regression and Linear SVC completely misclassified the six variants of the statement "*Cukup membingungkan aplikasi ini awal awal y.*". BERT and mBERT misclassified the three Indonesian variants only, and IndoBERT misclassified the three Indonesian variants and the English End variant. XLM-RoBERTa classified every variant correctly. TF-IDF is based on Bag-of-Words, which discards information about word order and semantics and maps strictly on word presence [48]. For the two linear models, the six variants of this review share every word except the sentiment cue, so identical predictions across the six variants are what a model that gives the cue little weight would produce. Table 10 shows further examples in which both linear models misclassified variants of reviews that are absent from the training set.

The three reviews in Table 10 are also absent from the training set. In the first two, an English variant of a Negative review was predicted Positive by both models. In the third, an Indonesian variant of a Positive review was predicted Negative. The error counts of the two linear models do not differ significantly between positions or languages based on McNemar's test, yet Table 8 shows that they flipped 5.1% and 1.5% of the predictions relative to the original variant, so they were not insensitive to the manipulation.

In Table 9, BERT classified the three English variants correctly and the three Indonesian variants incorrectly, in line with the language difference reported above (16 errors on Indonesian cues against 5 on English, adjusted $p = 0.0137$). Table 11 lists four BERT errors from four different review groups, two on Indonesian variants and two on English variants, so the language difference does not account for every BERT error.

Table 10. Examples of shared errors from Logistic Regression and Linear SVC

| Review | Position | Language | True Label | Predicted |
|---|---|---|---|---|
| cari multipayment utk transferpay Quite complicated. | End | English | Negative | Positive |
| unclear Ticketing untuk semua konser K-Pop. | Beginning | English | Negative | Positive |
| Terasa keren seperti punya travel agent sendiri. | Middle | Indonesian | Positive | Negative |

Table 11. BERT error samples

| Review | Position | Language | True Label | Predicted |
|---|---|---|---|---|
| sejak diperbarui Aplikasi tidak dapat digunakan. | End | Indonesian | Negative | Positive |
| Terasa seperti punya travel agent sendiri.cool | End | English | Positive | Negative |
| stasiun jurang mangu, dissapointing, kaga ada. | Middle | English | Negative | Positive |

| | | | | |
|---|---|---|---|---|
| untuk aplikasinya masih sering galat, untuk lihat my order nya | Middle | Indonesian | Negative | Positive |

Table 12. mBERT error samples

| Review | Position | Language | True Label | Predicted |
|---|---|---|---|---|
| pesan hotel lebih hemat dari tiket. com | Middle | Indonesian | Positive | Negative |
| Terasa seperti punya travel agent sendiri.keren | End | Indonesian | Positive | Negative |
| untuk lihat my order nya, untuk aplikasinya masih sering galat | End | Indonesian | Negative | Positive |

Table 13. IndoBERT error samples

| Review | Position | Language | True Label | Predicted |
|---|---|---|---|---|
| cari multipayment Agak ribet utk transferpay. | Middle | Indonesian | Negative | Positive |
| untuk lihat my order nya, untuk aplikasinya still prone to error | End | English | Negative | Positive |
| most recommended Aplikasi kalau mau dipakai untuk inspirasi bepergian | Beginning | English | Positive | Negative |

mBERT correctly classified half of the variants in Table 9, while IndoBERT misclassified four of the six variants (three Indonesian and the English End variant). The group is inconsistent so it lowered PCR and several of its predictions diverge from the original and raised PFR. This is different from the high PCR but low PFR shown before. This reflected partial and unstable sensitivity to the sentiment cue rather than complete indifference. Table 7 showed mBERT had 0 errors on English cues and 8 on Indonesian cues, indicating a higher error rate on Indonesian cues (adjusted $p = 0.0313$). The eight errors come from four review groups: the three in the review of Table 9, the three in "*pesan hotel lebih hemat dari tiket. com*", and one each in the other two reviews of Table 12. For "*pesan hotel lebih hemat dari tiket. com*", all three English variants were classified correctly while all three Indonesian variants were predicted Negative. The second review in Table 12 is one whose variants were also misclassified by Logistic Regression, Linear SVC and BERT shown in Tables 10 and 11. Every mBERT error is on an Indonesian-language cue.

IndoBERT's accuracy was 0.95, yet it had the lowest Prediction Consistency Rate of the six models (0.8727). In Table 9 it classified only two of the six variants correctly: it misclassified the three Indonesian variants and the English End variant. Unlike BERT and mBERT, IndoBERT's difference between Indonesian and English cues was not statistically significant, so we cannot attribute its errors to the language of the cue. Its 15 misclassified variants come from seven reviews, and the review of Table 9 alone contributes four of them.

Table 13 shows three examples. In the first, IndoBERT predicted Positive for the three Indonesian variants of a Negative review ("*cari multipayment Agak ribet utk transferpay*"), while the English variants of the same review were classified correctly. In the second and third examples the misclassified variants are English variants: a Negative review predicted Positive and a Positive review predicted Negative. All Indonesian variants of the third review were classified correctly. IndoBERT's 15 errors are split almost evenly between Indonesian and English variants in Table 7.

To address the question whether XLM-RoBERTa's invariance is from the model correctly processing the sentiment cue of the variations or the limited template diversity of the 330 samples test set, we also ran random stratified splitting across five different random seeds grouped by reviewId [49]. Table 14 showed accuracy, macro F1, PCR and PFR, averaged over five repeated splits instead of a single split that was reported previously. Every model's mean accuracy and macro F1 are lower across the five splits, especially for IndoBERT which fell from 0.95 to a mean of 0.907 and BERT which fell from 0.94 to a mean of 0.871. XLM-RoBERTa is not an exception as it no longer has zero errors after the five splits, dropping to a mean accuracy and macro F1 of 0.977. Nonetheless, it remained the best performing model with the smallest variance of all six models on PCR and PFR. The split also showed mBERT scored higher than IndoBERT on the single split (0.98 against 0.95) and on the five-split mean

p-ISSN: 2541-1179
e-ISSN: 2581-1711
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

(0.942 against 0.907). mBERT also showed the highest stability of any model with 0.012-0.013 standard deviation, even though its PCR is comparatively volatile as BERT.

### 3.2. Evidence tests for background-token reliance

Across Tables 9 to 13, a few words recur in the misclassified review groups: "*aplikasi*" (Tables 9, 11, 12 and 13), "*travel*" (Tables 10, 11 and 12), "*cari*" (Tables 10 and 13), "*dari*" (Table 12), "*kalau*" (Table 13) and "*semua*" (Table 10). These six words were selected after inspecting the errors, so the tests below are post hoc.

Table 15 shows that "*aplikasi*" appears in 61 of 217 training reviews (28.1%) and that 85.2% of them are positive, against a base rate of 53.5% (odds ratio 8.31, Fisher p = 1.6e-9, Holm-adjusted p < 0.001 over six tokens). Its coefficient is the largest of 3,077 features in both Logistic Regression (2.79) and Linear SVC (1.37). None of the other five tokens remained significant after correction: "*cari*" (82.4% positive, adjusted p = 0.10) and "*travel*" (87.5% positive, adjusted p = 0.28) lean positive, "*dari*" occurs in 2 training reviews, "*kalau*" in 5 (40% positive) and "*semua*" in 3. For the third review of Table 10, which contains "*travel*", the Logistic Regression and Linear SVC coefficients of the word are negative (-0.24 and -0.13) although 7 of its 8 training reviews are positive.

Table 14. Repeated stratified split stability results

| Model | Accuracy | Macro F1 | PCR | PFR |
|---|---|---|---|---|
| Logistic Regression | 0.901 ± 0.031 | 0.899 ± 0.031 | 0.916 ± 0.038 | 0.042 ± 0.021 |
| Linear SVC | 0.908 ± 0.026 | 0.906 ± 0.026 | 0.924 ± 0.030 | 0.033 ± 0.018 |
| BERT | 0.871 ± 0.034 | 0.867 ± 0.037 | 0.829 ± 0.064 | 0.085 ± 0.025 |
| mBERT | 0.942 ± 0.012 | 0.941 ± 0.013 | 0.909 ± 0.062 | 0.039 ± 0.022 |
| XLM-RoBERTa | 0.977 ± 0.017 | 0.977 ± 0.017 | 0.978 ± 0.008 | 0.013 ± 0.008 |
| IndoBERT | 0.907 ± 0.039 | 0.904 ± 0.042 | 0.869 ± 0.041 | 0.061 ± 0.016 |

Table 15. Training-set statistics of the six background tokens

| Token | Training reviews (%) | Positive | Odds ratio | Fisher p | Holm p | LR coef. (rank) | SVC coef. (rank) | Test reviews |
|---|---|---|---|---|---|---|---|---|
| aplikasi | 61 (28.1%) | 85.2% | 8.31 | 1.6e-09 | 9.8e-09 | 2.79 (1) | 1.37 (1) | 18 |
| dari | 2 (0.9%) | 100.0% | inf | 0.50 | 1.00 | 0.54 (106) | 0.38 (161) | 5 |
| kalau | 5 (2.3%) | 40.0% | 0.57 | 0.67 | 1.00 | -0.04 (1388) | 0.03 (1268) | 1 |
| cari | 17 (7.8%) | 82.4% | 4.48 | 0.020 | 0.101 | 0.32 (374) | 0.28 (267) | 4 |
| semua | 3 (1.4%) | 66.7% | 1.75 | 1.00 | 1.00 | 0.14 (886) | 0.05 (1129) | 1 |
| travel | 8 (3.7%) | 87.5% | 6.42 | 0.070 | 0.282 | -0.24 (2491) | -0.13 (2279) | 4 |

Table 16. Effect of deleting "*aplikasi*" on the 108 test variants (18 reviews) that contain it

| Model | Flip after deleting "aplikasi" | Flip after deleting a random word | Repaired errors | Margin change: token minus random (mean abs.) | Token / cue margin change |
|---|---|---|---|---|---|
| Logistic Regression | 7.4% [0.0, 17.6] | 1.3% | 4/10 | 0.35 [0.27, 0.46] | 0.90 |
| Linear SVC | 6.5% [0.0, 18.5] | 1.7% | 0/6 | 0.16 [0.13, 0.21] | 0.87 |
| BERT | 8.3% [0.9, 16.7] | 5.4% | 6/11 | 0.67 [-0.02, 1.56] | 1.03 |
| mBERT | 13.0% [1.9, 26.9] | 0.9% | 4/4 | 1.38 [0.19, 2.90] | 0.73 |
| XLM-RoBERTa | 0.0% [0.0, 0.0] | 0.0% | n/a (No Errors) | -0.01 [-0.04, 0.03] | 0.03 |
| IndoBERT | 10.2% [2.8, 19.5] | 4.8% | 5/8 | 0.91 [0.18, 1.88] | 0.79 |

Table 16 reports the effect of deleting "*aplikasi*" in the 108 test variants (18 reviews) that contain it. Deleting the token flipped 6.5% to 13.0% of the predictions of Logistic Regression, Linear SVC, BERT, mBERT and IndoBERT and none of XLM-RoBERTa's. Deleting a random other word flipped 0.9% to 5.4% for the same five models and 0.0% for XLM-RoBERTa. The 95% interval of the paired difference in flip rate included zero or had zero as its lower bound for every model. The sentiment cue itself was also not decisive by this measure: deleting it flipped only 10% to 15% of predictions in all six models, so the flip rate is a coarse measure in this dataset. For the other five tokens, the paired difference in flip rate between deleting the token and deleting a random word was never reliably above zero for any model (every interval included zero or was negative; "*kalau*" and "*semua*" each occur in one test review only). Deleting "*dari*" did not correct any of the three misclassified mBERT variants that contain it, and deleting "*kalau*" did not correct either of the two misclassified IndoBERT variants that contain it. Replacing "*aplikasi*" with a neutral filler word gave the same pattern as deleting it and is not reported separately.

The mean absolute margin change is more informative. Deleting "*aplikasi*" changed the margin more than deleting a random word for Logistic Regression (difference 0.35, interval 0.27 to 0.46), Linear SVC (0.16, 0.13 to 0.21), mBERT (1.38, 0.19 to 2.90) and IndoBERT (0.91, 0.18 to 1.88), and with an interval that just includes zero for BERT (0.67, -0.02 to 1.56). For XLM-RoBERTa the difference was -0.01 (-0.04 to 0.03). Relative to deleting the sentiment cue, deleting "*aplikasi*" produced between 0.73 and 1.03 times the margin change for the five models other than XLM-RoBERTa and 0.03 times for XLM-RoBERTa, whose margin changed by a mean of 2.83 when the cue was deleted. Among the variants that were originally misclassified and contain "*aplikasi*", deleting the token corrected 6 of 11 for BERT, 5 of 8 for IndoBERT, 4 of 4 for mBERT, 4 of 10 for Logistic Regression and 0 of 6 for Linear SVC, against 9%, 0%, 15%, 8% and 0% for random deletion. These counts come from 4, 4, 2, 2 and 1 reviews. On the review of Table 9, deleting "*aplikasi*" did not change the Logistic Regression or Linear SVC prediction in any of the six variants. Margin changes are in each model's own units and are compared only within a model.

Table 17 reports the attention ratio defined previously for the test sentences that contain "*aplikasi*". For the 11 misclassified BERT sentences (4 reviews) the ratio ranged from 2.62 to 4.03, but the mean ratio is the same for the 97 correctly classified sentences containing the token (3.14, interval 2.91 to 3.36) as for the misclassified ones (3.08, 2.68 to 3.56). The BERT tokenizer splits "*aplikasi*" into 3 to 4 word pieces, and the ratio per word piece is 0.95 for correct and 0.94 for misclassified sentences. Relative to the sentiment cue in the same sentence, the token received 1.14 times the cue's attention in correct sentences and 0.47 times (interval 0.31 to 0.65; 9 sentences from 3 reviews) in misclassified

sentences. mBERT, XLM-RoBERTa and IndoBERT showed ratios of 1.31 to 1.64 on correct sentences, with per-piece ratios of 1.01 to 1.30.

The median relative completeness errors were 0.034 for BERT, 0.002 for IndoBERT, 0.013 for mBERT and 0.054 for XLM-RoBERTa; 21 of 410 XLM-RoBERTa rows exceeded the exclusion threshold and were left out. In correctly classified sentences, "*aplikasi*" received 0.89 to 1.82 times the attribution expected from an even spread, and the sentiment cue 1.19 to 1.64 times (Table 18). In misclassified sentences the attribution to "*aplikasi*" had wide intervals (BERT 1.17, IndoBERT 2.02, mBERT 2.29; 4, 4 and 2 reviews) and its signed contribution (the signed attribution divided by the total absolute attribution, so positive values push towards the correct label) towards the true label was negative (-0.14, -0.22 and -0.24), with intervals reaching zero for BERT and IndoBERT. In the same misclassified sentences, the sentiment cue also contributed negatively towards the true label (BERT -0.15, interval -0.27 to -0.07; mBERT -0.17, interval -0.39 to -0.06). For XLM-RoBERTa, "*aplikasi*" received 0.89 times the uniform share with a signed contribution of 0.03 (interval -0.02 to 0.07), against 1.19 and 0.16 for the cue.

Table 17. Attention received by "*aplikasi*" in test sentences that contain it

| Model | Outcome | Sentences / reviews | Ratio (token vs. other tokens) | Ratio per word piece | Token / cue ratio (sentences/reviews) |
|---|---|---|---|---|---|
| BERT | Correct | 97 / 18 | 3.14 [2.91, 3.36] | 0.95 [0.91, 0.98] | 1.14 [0.90, 1.43] (96/18) |
| BERT | Error | 11 / 4 | 3.08 [2.68, 3.56] | 0.94 [0.89, 0.99] | 0.47 [0.31, 0.65] (9/3) |
| mBERT | Correct | 104 / 18 | 1.64 [1.47, 1.83] | 1.30 [1.25, 1.35] | 0.63 [0.50, 0.79] (102/18) |
| mBERT | Error | 4 / 2 | 1.45 [1.27, 1.98] | 1.20 [0.99, 1.27] | 0.36 (3/1) |
| XLM-RoBERTa | Correct | 108 / 18 | 1.50 [1.32, 1.70] | 1.13 [1.08, 1.17] | 0.73 [0.57, 0.92] (105/18) |
| IndoBERT | Correct | 100 / 18 | 1.31 [1.15, 1.48] | 1.01 [0.98, 1.06] | 0.64 [0.49, 0.84] (97/18) |
| IndoBERT | Error | 8 / 4 | 1.37 [1.08, 2.04] | 1.12 [0.98, 1.34] | 0.43 [0.35, 0.49] (8/4) |

Table 18. Integrated Gradients attribution to the token "*aplikasi*" and to the sentiment cue

| Model | Outcome | Rows/reviews (token; cue) | Token share / uniform | Token toward true label | Cue share / uniform | Cue toward true label |
|---|---|---|---|---|---|---|
| BERT | Correct | 97/18; 283/54 | 1.15 [0.98, 1.30] | 0.18 [0.12, 0.24] | 1.64 [1.46, 1.83] | 0.22 [0.19, 0.25] |
| BERT | Error | 11/4; 19/5 | 1.17 [0.71, 1.59] | -0.14 [-0.28, 0.01] | 1.29 [0.94, 1.54] | -0.15 [-0.27, -0.07] |
| mBERT | Correct | 104/18; 295/55 | 1.82 [1.03, 2.92] | 0.16 [0.09, 0.24] | 1.43 [1.25, 1.61] | 0.28 [0.23, 0.32] |
| mBERT | Error | 4/2; 7/3 | 2.29 [1.02, 2.71] | -0.24 [-0.28, -0.11] | 1.37 [0.95, 3.88] | -0.17 [-0.39, -0.06] |
| XLM-RoBERTa | Correct | 104/18; 285/55 | 0.89 [0.76, 1.03] | 0.03 [-0.02, 0.07] | 1.19 [1.08, 1.31] | 0.16 [0.12, 0.20] |

| Model | Outcome | Rows/reviews (token; cue) | Token share / uniform | Token toward true label | Cue share / uniform | Cue toward true label |
|---|---|---|---|---|---|---|
| IndoBERT | Correct | 100/18; 287/55 | 1.11 [0.86, 1.40] | 0.07 [0.00, 0.12] | 1.23 [1.13, 1.33] | 0.23 [0.20, 0.26] |
| IndoBERT | Error | 8/4; 15/7 | 2.02 [0.96, 2.73] | -0.22 [-0.34, 0.01] | 0.73 [0.57, 0.99] | -0.06 [-0.16, 0.04] |

### 3.3. Discussion

The results support the central argument of this study that aggregate scores say little about how a model treats the sentiment cue, but they support the shortcut explanation only for some of the models. The clearest case is the token "*aplikasi*" in the two TF-IDF models. It is the most positive feature of both models and appears in more than a quarter of the training reviews, and deleting it changes their output margins by about as much as deleting the sentiment cue (Table 16). The token does not explain every error, however. On the review of Table 9, Logistic Regression and Linear SVC kept their wrong prediction in all six variants after "*aplikasi*" was deleted, so other words of the same review also contribute. Because TF-IDF is a bag-of-words representation, these two models are also insensitive to the position of the cue, and their error differences between conditions were not significant (Table 6 and Table 7). They are not blind to the manipulation either, since they flipped 5.1% and 1.5% of their predictions relative to the original variant (Table 8).

Among the transformers the picture is less uniform. Deleting "*aplikasi*" changed the margins of mBERT and IndoBERT clearly more than deleting a random word, and that of BERT by a comparable amount (0.67), although its interval just includes zero. In each of these three models the token produced between 0.73 and 1.03 times the margin change caused by deleting the sentiment cue (Table 16). This is consistent with some reliance on the token, but it rests on 18 reviews and one training run per model, and the counts of corrected errors come from only two to four reviews. XLM-RoBERTa behaved differently on every test: its margin did not change when "*aplikasi*" was deleted, it changed strongly when the cue was deleted, and the integrated-gradients attribution to the token was below the uniform share with a signed contribution close to zero (Table 18). This agrees with its perfect scores on the reported split. Its larger multilingual pretraining corpus and its SentencePiece tokenizer [43] are possible reasons but we did not test them. It does not make the model immune, because it made errors on other splits (mean accuracy 0.977, Table 14) and the test set has limited template diversity.

The tests do not support reading BERT's attention as evidence that the model concentrates on a background token. The ratio of 2.62 to 4.03 between the attention received by "*aplikasi*" and by the average other token is the same in correctly and incorrectly classified sentences (3.14 and 3.08). It is largely a consequence of the tokenizer splitting "*aplikasi*" into 3 to 4 word pieces, which gives about 0.95 per piece, and relative to the sentiment cue the token received less attention in misclassified sentences (0.47) than in correct ones (1.14) (Table 17). Attention weights are in any case not a faithful explanation of a prediction [33], [34] so we treat them as a control and let the interventions of Table 16 carry the argument.

The same caution applies to the other words that appear in the error samples. "*Dari*" occurs in only 2 training reviews and "*kalau*" in 5 (40% positive, so not negatively associated), and deleting either word changed predictions no more often than deleting a random word. The data therefore do not show that these words drive the errors of mBERT and IndoBERT, and we keep these cases as observations: mBERT misclassified the Indonesian variants of "*pesan hotel lebih hemat dari tiket. com*", and IndoBERT misclassified English variants of a review that contains "*kalau*". Many errors need not come from background words at all. In misclassified sentences the sentiment cue itself pushed the prediction away from the true label for BERT and mBERT (Table 18), which could reflect an ambiguous cue or label noise as much as a shortcut, and two of the six reviews behind BERT's errors do not contain "*aplikasi*".

The language of the cue matters for two of the models. BERT and mBERT made significantly more errors on Indonesian than on English cues (Table 7; adjusted $p = 0.0137$ and 0.0313). The BERT checkpoint used here is English-only and mBERT's Indonesian pretraining data are limited compared

with English, which are plausible reasons [18], but our tests do not separate them from other causes. IndoBERT was pretrained on Indonesian text [44], so the language of the cue alone does not explain why its predictions are less consistent than those of the multilingual models. IndoBERT and XLM-RoBERTa showed no significant language difference, although with 55 test reviews and few discordant pairs the test has limited power.

These results support and complicate previous work on code-switched sentiment classification. Astuti et al. reported that IndoBERTweet performed best among the models they tested on a naturally occurring Indonesian-English Twitter dataset, with an average accuracy of 76.56% [11]. Chanda et al. showed that adding a language identification step improves mBERT's performance on Dravidian code-switched comments [9]. These datasets differ from ours, so the scores are not directly comparable, but both studies evaluate models only on naturally occurring instances, whereas the aggregate scores in Table 5 are above 0.90 macro F1 for every transformer here. The condition-based and invariance results add a different view. IndoBERT is third in accuracy on the reported split (0.95) yet has the lowest PCR on the reported configuration (0.8727, which rose to 0.9091 in the sensitivity run), and mBERT has the second-highest accuracy (0.98) with a PCR below Linear SVC's (0.9273 against 0.9636). Over the five splits, the two baselines have higher mean PCR (0.916 and 0.924) than BERT (0.829), IndoBERT (0.869) and mBERT (0.909), although mBERT and XLM-RoBERTa have higher mean accuracy than both baselines (Table 14). A high score on naturally occurring data therefore does not guarantee the same reliability once the sentiment cue is moved or translated, and whether a transformer is more reliable than a bag-of-words baseline depends on the model and on the metric. No general statement should be drawn from these six models.

We cannot say whether these patterns are specific to Indonesian-English reviews or reflect a more general property of the architectures. Studies of English natural language inference and of BERT attention report related heuristics [15], [16], but this study did not test other language pairs, domains or tasks, and it does not show that reliance on frequent surface tokens is a general training artifact. Replicating the protocol on other pairs, in particular pairs with a different script or a resource asymmetry such as Hindi-English or Javanese-Indonesian, is future work.

Invariance checks on matched minimal pairs could be run alongside F1 as a diagnostic before a code-switched sentiment model is deployed, and the token-level interventions of Table 16 give a way to find which background tokens a model leans on. This study does not validate PCR or PFR as acceptance criteria, because no threshold was calibrated and we did not relate the metrics to failures in deployment. Judging a model only by its score on a fixed test set can hide whether that score holds once the sentiment cue is moved or translated.

## 4. CONCLUSION

This study reframed Indonesian-English code-switched sentiment analysis as a controlled behavioral invariance probe, testing whether sentiment classifiers keep their prediction when the sentiment cue is moved within a sentence or translated. Position and language of the cue were manipulated across constructed minimal pairs, and a group-based split kept all variants of a review together.

Standard accuracy did not predict invariance: IndoBERT and mBERT scored 0.95 and 0.98 on the reported split yet had lower prediction consistency than Linear SVC, and BERT and mBERT made significantly more errors on Indonesian than on English cues. Intervention tests showed that the frequent token "aplikasi" is the most heavily weighted feature of Logistic Regression and Linear SVC and that deleting it changes their output margins nearly as much as deleting the sentiment cue; it also changed the margins of mBERT and IndoBERT more than a random deletion, whereas XLM-RoBERTa was not affected by it on the reported split. Attention weights, examined with controls, did not provide evidence of shortcut reliance. These findings come from one split, one training run per model and 18 test reviews that contain the token, so they describe these models on this dataset and not transformers in general.

These results must be interpreted alongside the dataset's limitations. The variants were validated by a single reviewer, so naturalness and acceptability judgments rely on one person rather than inter-annotator agreement. Both the cue translations and the variations audit were produced by the first author. Moving the cue creates word orders that are less natural than the original sentences by design, so some errors may reflect that manipulation and not a shortcut. The test set has 55 review groups, errors are

concentrated in 13 of them (two reviews account for 45% of all misclassified variants), and the McNemar comparisons have limited power. Each transformer was fine-tuned once on the reported split with identical hyperparameters and no validation set. The training loss fell to between 0.010 and 0.108, so we found no sign of underfitting, but other hyperparameters could change the results. A sensitivity run of IndoBERT with a higher learning rate and more epochs raised its PCR by two review groups, a change smaller than the standard deviation of its PCR across the five splits (0.041), so the position of IndoBERT as the least consistent model on the reported split is specific to this configuration. In addition, the five repeated splits (Table 14) give mean accuracies up to 0.07 below the reported split and a different order of the models, so the ranking of models is not stable. The six background tokens were chosen after inspecting the errors. The location of the cue is exact for Indonesian variants and estimated for English variants. Part of the error may come from ambiguous cues or label noise rather than from shortcuts. Attention and Integrated Gradients are explanation methods, and only the interventions change the input and observe the prediction. Finally, all results come from one domain (Tiket.com app reviews) and one language pair (Indonesian-English).

On this dataset, high-performance scores did not guarantee consistent predictions when the sentiment cue was moved or translated. If the model is to handle real, messy and informal code-switched input, then standard evaluation metrics on a fixed test set are not enough on their own, as the predictions of some models here changed once the position or language of the cue changed. Future evaluations of code-switched sentiment models could include controlled minimal-pair tests such as these alongside standard metrics.

## REFERENCES

[1] B. Liu, *Sentiment analysis : mining opinions, sentiments, and emotions*. Cambridge University Press, 2020. doi: 10.1017/9781108639286.

[2] S. Poplack, “Sometimes I’ll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching,” *Linguistics*, vol. 18, no. 7–8, 1980, doi: 10.1515/ling.1980.18.7-8.581.

[3] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi, “Pre-trained language model for code-mixed text in Indonesian, Javanese, and English using transformer,” *Soc. Netw. Anal. Min.*, vol. 15, no. 1, Dec. 2025, doi: 10.1007/s13278-025-01444-9.

[4] G. Winata, A. F. Aji, Z. X. Yong, and T. Solorio, “The Decades Progress on Code-Switching Research in NLP: A Systematic Survey on Trends and Challenges,” in *Findings of the Association for Computational Linguistics: ACL 2023*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 2936–2978. doi: 10.18653/v1/2023.findings-acl.185.

[5] A. F. Hidayatullah, R. A. Apong, D. T. C. Lai, and A. Qazi, “Corpus creation and language identification for code-mixed Indonesian-Javanese-English Tweets,” *PeerJ Comput. Sci.*, vol. 9, p. e1312, 2023, doi: 10.7717/peerj-cs.1312.

[6] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, and P. Fung, “Are Multilingual Models Effective in Code-Switching?,” in *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2021, pp. 142–153. doi: 10.18653/v1/2021.calcs-1.20.

[7] S. Khanuja, S. Dandapat, A. Srinivasan, S. Sitaram, and M. Choudhury, “GLUECoS: An Evaluation Benchmark for Code-Switched NLP,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 3575–3585. doi: 10.18653/v1/2020.acl-main.329.

[8] P. Patwa *et al.*, “SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Stroudsburg, PA, USA: International Committee for Computational Linguistics, 2020, pp. 774–790. doi: 10.18653/v1/2020.semeval-1.100.

[9] S. Chanda, A. Mishra, and S. Pal, “Sentiment analysis of code-mixed Dravidian languages leveraging pretrained model and word-level language tag,” *Natural Language Processing*, vol. 31, no. 2, pp. 477–499, Mar. 2025, doi: 10.1017/nlp.2024.30.

[10] E. Yulianti, A. Kurnia, M. Adriani, and Y. S. Duto, "Normalisation of Indonesian-English Code-Mixed Text and its Effect on Emotion Classification," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021, doi: 10.14569/IJACSA.2021.0121177.

[11] L. W. Astuti, Y. Sari, and Suprapto, "Code-Mixed Sentiment Analysis using Transformer for Twitter Social Media Data," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, 2023, doi: 10.14569/IJACSA.2023.0141053.

[12] D. Sankoff and S. Poplack, "A formal grammar for code-switching," *Paper in Linguistics*, vol. 14, no. 1, pp. 3–45, Jan. 1981, doi: 10.1080/08351818109370523.

[13] C. Myers-Scotton, *Duelling Languages: Grammatical Structure in Codeswitching*. Oxford University Press, 1993. doi: 10.1093/oso/9780198240594.001.0001.

[14] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond Accuracy: Behavioral Testing of NLP Models with CheckList," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 4902–4912. doi: 10.18653/v1/2020.acl-main.442.

[15] R. T. McCoy, E. Pavlick, and T. Linzen, "Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 3428–3448. doi: 10.18653/v1/P19-1334.

[16] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does BERT Look at? An Analysis of BERT's Attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 276–286. doi: 10.18653/v1/W19-4828.

[17] A. Vaswani *et al.*, "Attention Is All You Need," 2017, Accessed: May 15, 2026. [Online]. Available: http://arxiv.org/abs/1706.03762

[18] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych, "How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 3118–3135. doi: 10.18653/v1/2021.acl-long.243.

[19] Y. Aliyu, A. Sarlan, K. U. Danyaro, A. S. B. A. Rahman, and M. Abdullahi, "Sentiment Analysis in Low-Resource Settings: A Comprehensive Review of Approaches, Languages, and Data Sources," *IEEE Access*, vol. 12, pp. 66883–66909, 2024, doi: 10.1109/ACCESS.2024.3398635.

[20] M. K. Nazir, M. Bilal, and S. C. Shongwe, "Sentiment analysis for code-mixed low-resource languages: a systematic review of approaches, techniques, applications, challenges, and future directions," *Soc. Netw. Anal. Min.*, vol. 16, no. 1, p. 47, Mar. 2026, doi: 10.1007/s13278-026-01588-2.

[21] G. I. Winata *et al.*, "NusaX: Multilingual Parallel Sentiment Dataset for 10 Indonesian Local Languages," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 815–834. doi: 10.18653/v1/2023.eacl-main.57.

[22] G. Aguilar, S. Kar, and T. Solorio, "LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 1803–1813.

[23] B. Wilie *et al.*, "IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 843–857. doi: 10.18653/v1/2020.aacl-main.85.

[24] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 10660–10668. doi: 10.18653/v1/2021.emnlp-main.833.

[25] E. Hlavnova and S. Ruder, "Empowering Cross-lingual Behavioral Testing of NLP Models with Typological Features," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2023, pp. 7181–7198. doi: 10.18653/v1/2023.acl-long.396.

[26] M. Gardner *et al.*, "Evaluating Models' Local Decision Boundaries via Contrast Sets," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 1307–1323. doi: 10.18653/v1/2020.findings-emnlp.117.

[27] D. Kaushik, E. Hovy, and Z. C. Lipton, "Learning the Difference that Makes a Difference with Counterfactually-Augmented Data," Feb. 2020, [Online]. Available: http://arxiv.org/abs/1909.12434

[28] S. Tan and S. Joty, "Code-Mixing on Sesame Street: Dawn of the Adversarial Polyglots," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 3596–3616. doi: 10.18653/v1/2021.naacl-main.282.

[29] S. D. Das, A. Basak, S. Mandal, and D. Das, "AdvCodeMix: Adversarial Attack on Code-Mixed Data," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jan. 2022, pp. 125–129. doi: 10.1145/3493700.3493705.

[30] M. F. Adilazuarda, S. Cahyawijaya, G. I. Winata, P. Fung, and A. Purwarianti, "IndoRobusta: Towards Robustness Against Diverse Code-Mixed Indonesian Local Languages," in *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 25–34. doi: 10.18653/v1/2022.sumeval-1.5.

[31] S. B. T. Krishnaram, Y. Hu, and K. Saravanan, "When Surface Form Changes Moderation Decisions: A Paired Study of Code-Mixed Workflow Instability," Jun. 2026, [Online]. Available: http://arxiv.org/abs/2606.05654

[32] R. Geirhos *et al.*, "Shortcut learning in deep neural networks," *Nat. Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020, doi: 10.1038/s42256-020-00257-z.

[33] S. Jain and B. C. Wallace, "Attention is not Explanation," May 2019, [Online]. Available: http://arxiv.org/abs/1902.10186

[34] S. Wiegreffe and Y. Pinter, "Attention is not not Explanation," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1908.04626

[35] A. Valdivia, M. V. Luzón, E. Cambria, and F. Herrera, "Consensus vote models for detecting and filtering neutrality in sentiment analysis," *Information Fusion*, vol. 44, pp. 126–135, Nov. 2018, doi: 10.1016/j.inffus.2018.03.007.

[36] D. Rajput, W. J. Wang, and C. C. Chen, "Evaluation of a decided sample size in machine learning applications," *BMC Bioinformatics*, vol. 24, no. 1, Dec. 2023, doi: 10.1186/s12859-023-05156-9.

[37] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, Jan. 1988, doi: 10.1016/0306-4573(88)90021-0.

[38] A. M. Bakare, K. S. M. Anbananthen, S. Muthaiyah, J. Krishnan, and S. Kannan, "Punctuation Restoration with Transformer Model on Social Media Data," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031685.

[39] O. M. Alyasiri and Y. N. Cheah, "Multi-Class Text Classification using Machine Learning Techniques," *Engineering, Technology and Applied Science Research*, vol. 15, no. 3, pp. 22598–22604, Jun. 2025, doi: 10.48084/etasr.9994.

[40] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *Machine Learning: ECML-98*, C. Nédellec and C. Rouveirol, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[42] T. Pires, E. Schlinger, and D. Garrette, "How Multilingual is Multilingual BERT?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4996–5001. doi: 10.18653/v1/P19-1493.

[43] A. Conneau *et al.*, "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 8440–8451. doi: 10.18653/v1/2020.acl-main.747.

[44] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, Stroudsburg, PA, USA: International Committee on Computational Linguistics, 2020, pp. 757–770. doi: 10.18653/v1/2020.coling-main.66.

[45] V. B. Nguyen, C. Seifert, and J. Schlötterer, "CEval: A Benchmark for Evaluating Counterfactual Text Generation," in *Proceedings of the 17th International Natural Language Generation Conference*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 55–69. doi: 10.18653/v1/2024.inlg-main.6.

[46] Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947, doi: 10.1007/BF02295996.

[47] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[48] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.

[49] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines," Mar. 2021, [Online]. Available: http://arxiv.org/abs/2006.04884