INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTER SYSTEMS
E-ISSN: 2180-0650
VOLUME 11, ISSUE 2, 2025, pp. 124 – 136
DOI: https://doi.org/10.15282/ijsecs.11.2.2025.10.0142

RESEARCH ARTICLE

Comparative Analysis of Back-Translation Models for Normalization Mobile App User Reviews

Amran Salleh[1], Mohd Hafeez Osman[1*], Sa'adah Hassan[1], Mar Yah Said[1]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Jalan Universiti 1, 43400 Serdang, Selangor, Malaysia

ABSTRACT - The increase of mobile apps has led to an exponential growth of user-generated reviews, which are often noisy, informal, and linguistically diverse, thereby posing significant challenges for automated analysis in requirements engineering. This study evaluates whether back-translation (BT) can normalize informal reviews while preserving meaning, and which model (Google Translate vs Facebook M2M100_418M) offers better semantic preservation, grammatical quality, and lexical alignment. We collected 323 Google Play reviews (667 sentences) from three Malaysian government apps. Texts were cleaned, expanded for colloquial forms, and then BT was applied using Malay as an intermediate language. Evaluation used four metrics which are semantic similarity (Sentence-BERT), grammar error count (LanguageTool), BLEU (NLTK), and perplexity (GPT-2). Models differences were tested with paired t-tests and Wilcoxon signed- rank tests, while paired scatterplots showed distributional patterns. Google was significantly better on semantic similarity ($t(322)=5.38$, $p<.001$), grammar errors ($t(322)=3.66$, $p<.001$), and BLEU ($t(322)=2.99$, $p=.003$); effect sizes were small to moderate. Perplexity differences were not significant, indicating comparable sentence-level fluency. Visualizations confirmed Google's steadier performance with fewer extreme outliers. BT is a practical normalization step for noisy reviews. For the English–Malay pipeline studied here, Google provides more reliable semantic preservation and grammatical quality, while both systems are similar in fluency. However, the generalizability of these results are constrained by the relatively modest sample size (323 reviews, 667 sentences), and future work should validate results on large datasets and explore hybrid strategies combining strengths of both models.

ARTICLE HISTORY
Received : 14 July 2025
Revised : 24 October 2025
Accepted : 26 November 2025
Published : 18 December 2025

KEYWORDS
Back-translation
Mobile app reviews
Machine learning
Natural language processing
Evaluation metrics

1.0 INTRODUCTION

The widespread adoption of mobile applications has generated a vast repository of user-generated reviews that are inherently noisy, informal [1], and linguistically diverse [2]. These challenges hinder the performance of task-specific natural language processing (NLP) pipelines, particularly those involved in requirements engineering such as review normalization, sentiment extraction, and ambiguity resolution. Users commonly employ colloquial language [3]-[5], abbreviations, typographical errors [6], emoticons [7], and code-switching, which deviate from formal writing norms and complicate the identification of actionable requirements. Prior research has extensively investigated various approaches to review normalization, aiming to transform unstructured and noisy user-generated content into more structured and semantically meaningful representations. Existing review normalization approaches, including rule-based methods [8], machine learning algorithms [9], and hybrid techniques [10], [11]. Rule-based methods rely on handcrafted linguistic rules to clean and standardize textual input, but they often struggle with generalizability and language-specific variations. Meanwhile, machine learning algorithms such as Support Vector Machines (SVMs) and Decision Trees have been applied to detect noise, filter out irrelevant content, or normalize grammatical inconsistencies. Hybrid techniques have also emerged, combining statistical heuristics with shallow learning models or incorporating preprocessing pipelines that embed domain-specific dictionaries or patterns. While these approaches have shown some success in structured environments, they often fail to fully capture the semantic subtleties, contextual dependencies, and domain-specific terminologies inherent in app reviews, which are typically informal, multilingual, and highly variable.

In this context, recent advances in neural machine translation (NMT), particularly through the use of back-translation, offer promising avenues for overcoming these limitations. Back-translation (BT) involves converting a sentence from a source language into a target language and then translating it back into the original language, thereby producing synthetic data that helps improve model robustness and semantic clarity. This technique not only enhances training diversity in low-resource settings but also supports review normalization by simplifying, disambiguating, and aligning noisy expressions into more canonical forms. As noted by [12], back-translation can serve as a powerful transfer learning tool for identifying ambiguous phrases in software requirements and normalizing informal content into structured outputs. Their study reported measurable improvements in classification accuracy when back-translated texts were used to train

*CORRESPONDING AUTHOR | Mohd Hafeez Osman | ✉ hafeez@upm.edu.my

ambiguity detection models, especially using SVMs and logistic regression. However, the traditional back-translation process remains computationally intensive, particularly when applied at scale to millions of user reviews in multiple languages. To address this, [13] introduced m2m-100, a many-to-many multilingual NMT model that eliminates the English-centric bottleneck of prior systems. Unlike conventional pipelines that pivot through English, m2m-100 enables direct translation between 100 languages using shared encoder-decoder architectures augmented with sparse, language-specific parameters. This innovation significantly improves semantic preservation across translation directions, reducing topic drift and maintaining sentiment consistency. Their BLEU score comparisons show that m2m-100 consistently outperforms traditional English-centric models across non-English translation pairs, indicating its potential for high-fidelity text normalization and contextual preservation, even in linguistically diverse settings. Moreover, recent empirical studies [14] validate the utility of back-translation as a semantic alignment tool, showing that topic structures and sentiment scores remain largely consistent before and after iterative back-translation. These findings reinforce the claim that back-translation, particularly when enhanced by modern multilingual NMT models like m2m-100, can be applied not only for data augmentation but also as a core technique for improving review normalization pipelines.

Despite these advances, a gap remains in systematically evaluating how different back-translation models perform in the context of noisy mobile app reviews, particularly for low-resource languages. While previous studies work on automatic evaluation metrics, these are often optimized for high-resource languages, leaving an evaluating other low-resource language [15]. Metrics such as BLEU, METEOR, and TER tend to emphasize surface-level overlap and may not sufficiently capture deeper semantic or pragmatic equivalence in user-generated content. According to Mathur et al., [16] BLEU and TER have an 80% overlap in errors. In many cases, BT often does not maintain a sufficient level of equivalence between original and translated research materials [17]. Moreover, these metrics often overlook stylistic nuances and normalization quality that are critical for downstream applications in requirements engineering. Thus, existing metrics are insufficient to guide the selection of appropriate translation models or normalization strategies, particularly in multilingual or low-resource contexts. This study addresses this gap by adopting a multi-dimensional evaluation (semantic similarity, grammar errors, BLEU, and perplexity) to provide a more holistic assessment of translation quality in the normalization context. To sharpen the study's focus, the following research question (RQ) is explicitly formulated at the outset:

RQ_1 Which translation model, between Google Translate and the Facebook M2M100 model, demonstrates superior performance in preserving semantic meaning, correcting grammatical errors, and enhancing textual fluency in the context of back-translation for review normalization? Building on this RQ, we evaluate whether Google Trans- late systematically differs from Facebook's M2M100 model in back-translation quality. To achieve this, paired-sample hypotheses were formulated for each evaluation metric. For consistency, all paired differences were defined such that positive values indicate a Google advantage. Specifically, for higher-is-better metrics (semantic similarity, BLEU), differences were computed as $d_i = G_i - F_i$, whereas for lower-is- better metrics (grammar errors, perplexity), differences were computed as $d_i = F_i - G_i$. The null hypothesis (H_0) states that there is no difference between the two systems ($\mu_d = 0$), while the alternative hypothesis (H_1) states that a difference exists ($\mu_d \neq 0$). Formally:

1. Semantic Similarity (higher is better) $H_0: \mu_d = 0$ vs. $H_1: \mu_d \neq 0$
2. Grammar Errors (lower is better) $H_0: \mu_d = 0$ vs. $H_1: \mu_d \neq 0$
3. BLEU Score (higher is better) $H_0: \mu_d = 0$ vs. $H_1: \mu_d \neq 0$
4. Perplexity (lower is better) $H_0: \mu_d = 0$ vs. $H_1: \mu_d \neq 0$

This study aims to investigate the efficacy of back-translation in cleaning and normalizing informal, noisy mobile app user reviews, with a focus on evaluating the performance of two prominent translation models (Google Translate and m2m) across multiple quantitative metrics. This research contributes several novel aspects to the intersection of machine translation and requirements engineering:

i. This study presents an empirical investigation of back-translation used as a normalization technique for user-generated mobile app reviews, rather than as an augmentation method.

ii. Low-Resource Language Pair (English and Malay): In our comparative setup, we use Malay as an intermediate pivot language. Malay is still underrepresented in large-scale translation benchmarks, which makes it useful for examining multilingual natural language processing tasks. Prior research of multilingual focus on English-Centric, where training only on data which was translated from or to English [13]. As a result, Malay remains less explored due to a scarcity of dedicated studies and benchmarks [18]. Therefore, this study provides new empirical observations, especially in the context of languages with limited resources.

iii. This study applies a four-dimensional evaluation framework consisting of semantic similarity, grammar error detection, BLEU score, and perplexity.

In the rest of this paper, the structure is organized as follows. Section 2 reviews the related work that supports this study. Section 3 describes the materials used and explains the research method. Section 4 presents the results of the experiment together with discussion and interpretation. Finally, Section 5 concludes the paper and also highlights possible directions for future work.

2.0 RELATED WORKS

Back-translation has traditionally served as a data augmentation method in NLP, with studies such as [12] and [19] demonstrating improvements in classification performance through syntactic and semantic diversification. More recent research, however, reflects a trend toward domain-specific normalization, particularly in noisy user-generated content. Despite these advances, few studies investigate BT's role in requirements engineering or evaluate its performance within low-resource language pairs. Most rely on singular metrics like BLEU [20], overlooking nuanced trade-offs in fluency and semantic integrity. Conflicting findings make the evaluation of back-translation more difficult. For example, while [21] links BLEU with task accuracy, their results reveal a paradox where higher BLEU coincides with degraded fluency, as indicated by perplexity. These tensions reinforce the need for comprehensive, task-informed evaluation frameworks, which this study addresses through its multi-metric analysis of BT efficacy in review normalization.

2.1 Back-Translation in NLP

Several studies have demonstrated the effectiveness of back-translation in improving the performance of NLP models. [22] showed that back-translation could significantly enhance the performance of neural machine translation models. This technique helps to improve the performance of NMT, especially in situations where parallel data is limited or in low-resource language settings [23]. Similarly, [24] used data augmentation for supervised code translation learning to obtain new training data. Furthermore, [12, 25-27] used transfer learning and text augmentation can be applied to small data sets in requirements engineering to resolve ambiguous requirements. Originally popularized for machine translation and data augmentation, back translation involves translating a text from its source language into a target language and then translating it back to the original language [28]. This process effectively increasing the dataset size and diversity, which is particularly beneficial for low-resource languages [29]. The efficacy of back-translation has been demonstrated across multiple NLP tasks. For example, [30] developed the 'Cue Lexicon+' and integrated it into NMT systems to further enhance translation quality.

Recent studies have explored the application of BT as a means to reduce noise and ambiguities in user-generated content. For example, [28] showed that back translation could improve translation quality and model generalization by exposing models to diverse linguistic variations. Similarly, [19] shows that using back-translation to expand the data is especially helpful when working with a small dataset. This method can also reduce the imbalance in sample distribution and improve classification performance. In addition, it helps enhance the quality of translation. Beyond requirement engineering, BT has demonstrated success in text classification [31], especially when manually annotated data is limited. For instance, a study achieved a classification accuracy of 90.7% by using back-translation combined with model ensemble techniques, demonstrating its effectiveness in improving classification tasks. However, the quality of translation engines, language pair selection, and semantic drift during translation can impact the fidelity and utility of augmented data. Hence, BT should be carefully calibrated to the task and domain.

2.2 Back-Translation in Requirement Engineering

In requirements engineering, user requirements often contain ambiguities, lexical inconsistencies, and contextual noise, which hinder automated analysis and classification [32]. Several recent efforts have leveraged NLP techniques, including BT, to address these issues. [12] pioneered the use of BT for requirements classification, hypothesizing that the paraphrasing effect of BT could help in identifying ambiguous requirements more effectively. In their experimental study, [12] translated requirements from English into Spanish and back, generating paraphrased artifacts used as augmented data. Their findings indicated that models trained on BT-augmented data achieved significant improvements in classification accuracy, notably with SVM and logistic regression, with increases of up to 9.5% and 7.4%, respectively. This suggests that BT not only increases data diversity but also helps models generalize better to linguistic variations and reduce noise-induced errors. In a similar way, the study by [33] applied BT together with several machine learning (ML) and deep learning (DL) models. Their experiments showed that the models achieved an F1-score of 98.91% when using back-translated data, compared to 98.14% when using the original data.

Furthermore, analyses of BLEU scores [20] in these studies confirmed that BT introduces meaningful paraphrasing without significantly deviating from the original semantics, thus maintaining the integrity of requirements while reducing lexical ambiguity [12, 34]. Such techniques have been shown to be particularly effective in handling lexical ambiguities and coordination ambiguities, which are common in user requirements [32]. The application of BT in requirement engineering is still emergent but growing. A seminal study by [12] demonstrated the efficacy of BT in enhancing the classification accuracy of ambiguous versus unambiguous software requirements. Their research employed BT within a transfer learning framework to augment the original dataset with synthetic, noise-reduced variants of requirement statements. Meanwhile, [35] demonstrated in their research that using BT text made it possible to achieve an accuracy of 83.4% in distinguishing between machine-generated and human-written text. This method was also effective for detecting texts used for malicious purposes, such as plagiarism and fake reviews.

2.3 Application for User Reviews

In the context of user reviews, back-translation has been used to address issues of noise and ambiguity. Previous studies, such as [12] investigated the use of BT as a text augmentation technique to classify ambiguous vs. unambiguous requirements in software artifacts. BT led to significant improvements in classification accuracy [12], [21] for models such as SVM, Logistic Regression, and Multinomial Naive Bayes.

2.4 Neural Machine Translation Systems

The evolution of Neural Machine Translation has brought about significant advancements in handling multilingual content. Two prominent models (Google Translate and Facebook m2m-100) exemplify differing paradigms in multilingual NMT design. These differences are particularly consequential in back-translation tasks involving mobile app user reviews, where translation quality directly affects sentiment extraction, intent recognition, and overall user feedback analysis. Google Translate and NMT have undergone a notable evolution in recent years, leading to significant changes in translation industry practices [36]. The system's performance on various language pairs has been extensively documented, demonstrating state-of-the-art results on standard benchmarks. NMT models are indeed very promising, especially considering that the state-of-the art [37].

Facebook's m2m-100 model represents a significant advancement in multilingual translation [38], supporting direct translation between 100 languages without English pivoting. The model's architecture and training methodology differ substantially from Google's approach, potentially leading to different performance characteristics in back-translation tasks. Empirical evaluation from [13] shows that m2m-100 outperforms English-centric models in direct translation quality across non-English language pairs. For example, BLEU score improvements average over +7 points in many cases where English pivoting is removed. Table 9 of their work illustrates consistent performance gains for regional languages such as Hindi–Marathi (+6.4 BLEU) and Xhosa–Zulu (+3.5 BLEU). These improvements are critical when user feedback involves regional dialects and low-resource languages, which are common in diverse mobile markets.

Moreover, human evaluations in the same study indicate higher semantic accuracy for m2m-100 compared to English-centric systems, especially in unrelated language pairs. This finding suggests enhanced robustness of m2m-100 in preserving meaning across linguistically distant languages. Such ability is important in maintaining the authenticity of user sentiment in app reviews. However, m2m-100 is not without limitations. The model's performance gains vary significantly depending on the availability of training data. For extremely low-resource languages or domains with limited parallel corpora, the model may still produce sub-optimal translations. Additionally, the computational cost associated with training and deploying such large-scale multilingual models is non-trivial, potentially limiting real-time integration in mobile app platforms.

3.0 METHODS AND MATERIAL

This section discusses the materials and methods we used to compare the back translation model.

3.1 Materials

In this study, several instruments were used as follows.

A. Computational Resources

The experiments were conducted on a local laptop running Windows 11. The machine was equipped with an Intel Core i5 processor and 24 GB of RAM, which provided sufficient computing resources for the tasks performed. However, because of the limited hardware resources and the absence of a GPU, the experiments were processed in relatively small batches. This setup helped reduce memory consumption and avoid performance issues during the evaluation.

B. Programming languages

In this study, scripts were developed using the Python (version 3.11.0) programming language due to its simplicity and wide applicability in scientific computing. To facilitate interactive development and result visualization, Jupyter Notebook 6.5.3 was employed as the primary environment. The combination of Python and Jupyter Notebook provided an efficient and reproducible workflow for data analysis and computational experimentation.

C. Translation Systems

Google Translation is a commercial neural machine translation service leveraging Google's proprietary transformer-based architecture, trained on massive multilingual corpora and continuously updated. In our implementation, we use the following import statement: *from deep_translator import GoogleTranslator* into our python code. Meanwhile, Facebook M2M100_418M is an open-source multilingual model from the m2m-100 family, featuring 418 million parameters and trained to translate directly between 100 languages without English pivoting.

In this context, while a multitude of neural machine translation systems exist, this study deliberately focuses on a comparative analysis of Google Translate and Facebook's M2M100_418M. These models were selected based on their distinct architectural frameworks and relevance to low-resource language processing. Google Translate, a commercial benchmark with robust fluency performance, contrasts with M2M100_418M, an open-source model supporting direct

multilingual translation across 100 languages. This contrast offers a representative baseline for assessing BT efficacy in the normalization of multilingual mobile app reviews.

3.2 Methods

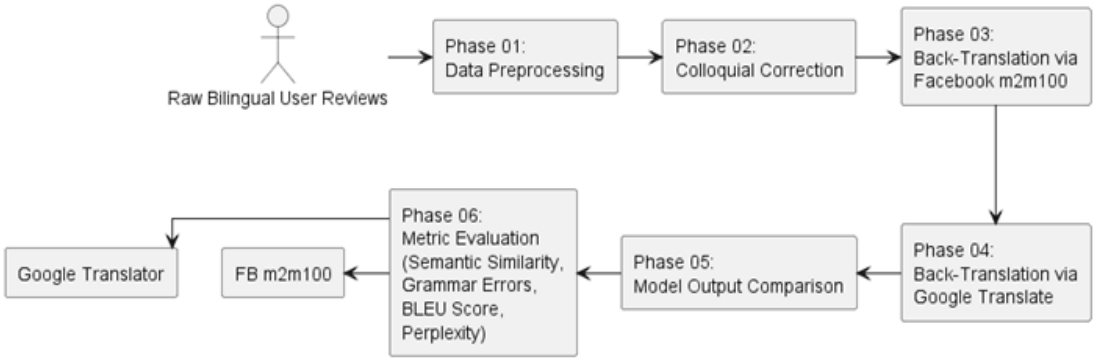


Figure 1. BT comparison methodology

In this subsection, we used user reviews collected from three government mobile applications: MyBayar PDRM, MyJPJ, and MySejahtera. A total of 323 reviews were obtained in July 2024 through automated Google Play scraping, comprising a mixture of English, Malay, and mix content. Specifically, the dataset included 88 reviews from MyBayar PDRM, 110 reviews from MyJPJ, and 125 reviews from MySejahtera, yielding 667 sentences after preprocessing. These figures are further detailed in Table 1, which summarizes the descriptive statistics of review length and sentence distribution, and the overall methodology pipeline is illustrated in Figure 1. While the dataset size is modest, we acknowledge limited power as a study limitation and recommend larger-scale validation in future work.

Table 1. Summary statistics of reviews and sentences

Metric	Value
Number of reviews	323
Average review length (in words)	16.12
Total number of sentences	667
Average sentence length (in words)	7.81
Number of sentences per review	2.07

A. Phase 01

Data preprocessing: Text preprocessing techniques include transformation, such as removing emojis and icons, and converting text to lowercase. In addition, spelling correction is applied, including handling of contractions.

B. Phase 02

Colloquial Correction Process: All user feedback will be cleaned using a three-stage approach. First, in the colloquial expansion stage, informal or abbreviated expressions will be translated into their standard forms. For example, *"Nk bukak memanjang no connection..bukak app lain okay je..banyak kali buat..sama je"*→ *"hendak buka selalu no connection. buka mobile apps lain okay banyak kali buat. Sama",* where the word *memanjang* changed to *selalu*.

Second, the domain-specific lexicon will be expanded to improve clarity. For instance, "*Maybe a MIUI bug when a non launcher registers itself as one"* → *"maybe a mobile user interface,bug when a non launcher registers itself as one"*, where *miui* will be replaced with *mobile user interface*, which is more descriptive. Finally, in the alphanumeric normalization stage, words that contain a mixture of letters and numbers will be converted into their full textual forms. An example *"ingtkn on9 semua xperlu ke kaunter"* → *"ingatkan online semua tidak perlu ke kaunter"*, where this is changing *on9* to *online*. This process helps to improve the quality of the data by making it more standardized and suitable for further analysis.

C. Phase 03 & 04

1. Forward translation: English → Intermediate Language (Malay selected for linguistic distance)
2. Backward translation: Intermediate Language → English

This translation process involves two main steps. First, the original English text is translated into an intermediate language. In this case, Malay was chosen because it has a significant linguistic distance from English, which can help reveal structural or semantic changes during translation. Then, in the backward translation step, the Malay version is

translated back into English. This allows researchers to identify any meaning shifts, ambiguities, or translation issues that may arise between the original and the translated texts.

D. Phase 05

In this phase, the semantic equivalence between the original and back-translated sentences was quantitatively evaluated using a sentence embedding model. The purpose of this step was to determine which translation model - Google Translate or Facebook's m2m100 - preserved the original meaning more effectively after the back-translation process. To achieve this, the Sentence-BERT (`SBERT`) model, specifically the `all-MiniLM-L6-v2` variant, was employed due to its capability to capture contextualized semantic representations of sentences.

Each review was encoded into a fixed-dimensional vector space using the SBERT model. Similarly, the corresponding back-translated sentences generated by Google Translate and Facebook m2m100 were independently encoded into their respective embeddings. The semantic similarity between the original sentence and each of its back-translated counterparts was then computed using the cosine similarity metric, defined as:

$$\text{Cosine Similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|\,|\mathbf{b}|} \qquad (1)$$

where **a** and **b** denote the embedding vectors of the original and back-translated sentences, respectively. The resulting similarity scores range from 0 to 1, where higher values indicate stronger semantic preservation.

D. Phase 06

Evaluation Metric: Automated evaluation (as shown in Figure 2) using the four metrics across all user review.

1. Semantic Similarity: Using Sentence-BERT (all-MiniLM-L6-v2), we computed cosine similarity between embeddings of original and back-translated sentences (`score = util.pytorch_cos_sim(emb1, emb2).item()`).
2. Grammar Errors: Grammar errors were identified using the `language_tool_python` library.
3. BLEU Score: Lexical overlap was computed using NLTK's `sentence_bleu`, capturing 1–4 gram matches between original and back-translated sentences:Computed using a pre-trained NLP model to assess the quality of the generated text (`bleu = sentence_bleu(ref_tokens, cand_tokens)`).
4. Perplexity: Evaluated based on the language model's ability to predict the next word in the translated text. The fluency was estimated via GPT-2 (gpt2 model), calculating the exponential of the negative log-likelihood (`perplexity = np.exp(loss.item())`).

```python
# Define metric functions
def compute_perplexity(text):
    inputs = gpt2_tokenizer(text, return_tensors="pt")
    with torch.no_grad():
        outputs = gpt2_model(**inputs, labels=inputs["input_ids"])
    loss = outputs.loss
    return np.exp(loss.item())

def compute_semantic_similarity(sent1, sent2):
    emb1 = semantic_model.encode(sent1, convert_to_tensor=True)
    emb2 = semantic_model.encode(sent2, convert_to_tensor=True)
    score = util.pytorch_cos_sim(emb1, emb2).item()
    return score

def compute_grammar_errors(text):
    matches = grammar_tool.check(text)
    return len(matches)

def compute_bleu(reference, candidate):
    ref_tokens = [word_tokenize(reference.lower())]
    cand_tokens = word_tokenize(candidate.lower())
    return sentence_bleu(ref_tokens, cand_tokens)
```

Figure 2. Evaluation metrics with GPT-2 and NLP library

In this study, the evaluation and statistical tests were conducted on the same dataset of 667 sentences, derived from 323 user reviews (**Table 1**). Each original sentence was paired with its back-translated outputs (Google and Facebook models), producing a one-to-one alignment for semantic similarity, grammar error count, BLEU score, and perplexity.

4.0 RESULTS AND DISCUSSION

This study investigates the effectiveness of back-translation in cleaning and normalizing informal, noisy mobile app user reviews. By using BT, these unstructured texts can be converted into semantically meaningful and syntactically well-

formed inputs. Two prominent translation systems, the Google Translation and the Facebook m2m100_418M model, were compared across four quantitative metrics: *semantic similarity*, *grammar errors*, *BLEU score*, and *perplexity*. Each metric captures a specific dimension of text quality and fidelity. The aggregated results are summarized in Table 2 and were visualized using paired scatterplot (Figure 4). At the same time, the examples of successful cases and unsuccessful cases are reported in Table 3 and Table 4, respectively. These tables are included to give a clearer view of how the approach performed in different situations.

Table 2. Multi-dimensional Performance Evaluation of Back-Translation Models

Model	Mean Score Semantic Similarity ↑	Mean Score Grammar Errors ↓	Mean Score BLEU Score ↑	Mean Score Perplexity ↓
Google Translate	**0.7258**	**5.8421**	**0.2419**	**732.24**
Facebook	0.6731	6.7895	0.2112	1582.80

Note: Arrows indicate the desired direction for optimal performance for each metric. Bold values indicate the superior model for a given metric, with Perplexity requiring nuanced interpretation as discussed in the text.

Google Translate performed better than Facebook's model across all four-evaluation metrics (see Table 2). In addition, a Wilcoxon signed-rank test was conducted to compare the overall performance of Google and Facebook m2m-100 models across four evaluation metrics (semantic similarity, grammar error, BLEU score, and perplexity) relevant to back-translation quality.

Table 3. Multi-dimensional Performance Evaluation of Back-Translation Models (Success Cases)

Phase / Model	Input / Output	Semantic Similarity↑	Grammar Error↓	BLEU Score↑	Perplexity↓
Raw user reviews	cannot even register, useless app released by useless gov.	-	-	-	-
Phase 1-2 (Processed)	cannot even register, useless mobile apps released by useless government.	-	-	-	-
Google Translate	Cannot register, useless mobile applications issued by useless governments.	0.9290	0	0.2090	1376.66
Facebook	not to register, useless mobile applications issued by useless government.	0.8726	1	0.2627	784.42

Table 4. Multi-dimensional Performance Evaluation of Back-Translation Models (Failure Cases)

Phase / Model	Input / Output	Semantic Similarity↑	Grammar Error↓	BLEU Score↑	Perplexity↓
BT Facebook Model					
Raw user reviews	pls help my booster dose complete 2 months ago but mysejahtera profile still not show the booster dose qr code & digital certificate. the shop not let me in have write to helpdesk few time but still not help, still not update my status.				
Phase 1-2 (Processed)	please help my booster dose complete dua months ago but mysejahtera profile still not show the booster dose quick-response (qr) code digital certificate. the shop not let me in have write to helpdesk few time but still not help, still not update my status.				
Back-translate	Please Please Please Please Please Please Please Please Please Please Please Please	0.0979	1	0.0026	1.27
BT Google Translate					
Raw user review	bayaq senang.				
Phase 1-2 (Processed)	bayar senang.				
Back-translate	Lying down.	0.1253	0	0.000	286.39

These findings are in line with the conclusions reported in [12], where BT was applied as a text augmentation method and showed clear improvements in classification accuracy, especially when using SVM and logistic regression models. The use of BT not only improved the readability and consistency of the text at the surface level but also helped in reducing semantic and syntactic variations. This normalization process is important because it allows the model to focus on the main meaning of the data rather than being distracted by noise or irregular patterns in the language.

4.1 Efficacy in Semantic Preservation and Grammatical Enhancement

The primary objective of back-translation is to improve linguistic quality without sacrificing the original user's intent. Google Translate demonstrated higher semantic preservation, aligning with its broader training corpus and commercial-grade optimization. To support this claim, we present results from an empirical analysis using statistical tests. In addition, we provide a scatter plot to illustrate the relationship between the two compared measurements. This combination of statistical evidence and visual representation helps to strengthen the argument that Google Translate performs more consistently in preserving the semantic intention of the original text.

4.1.1 Empirical result

The results of the paired t-tests and Wilcoxon signed-rank tests are presented in Table 5. The analysis was performed on 323 paired sentence-level observations across four evaluation metrics which are semantic similarity, grammar errors, BLEU and perplexity.

Table 5. Paired t-tests and Wilcoxon signed-rank tests (positive difference = Google better)

Metric	*n*	Paired *t*-test (two-sided)					Wilcoxon (two-sided)	
		Mean diff	95% CI	*t*(df)	*p*	d_z	*W*	*p*
Semantic similarity	323	0.053	[0.033, 0.072]	5.383 (322)	1.42e-07	0.299	1.64e+04	4.90e-06
Grammar Error	323	0.947	[0.438, 1.457]	3.659 (322)	2.96e-04	0.204	5189.500	4.43e-07
BLEU	323	0.031	[0.010, 0.051]	2.991 (322)	0.003	0.166	1.36e+04	5.35e-04
Perplexity	323	850.561	[-882.908, 2584.031]	0.965 (322)	0.335	0.054	2.25e+04	0.267

Notes. Differences defined so that positive values favor Google: for higher-is-better metrics (Semantic similarity, BLEU), $d_i = Google(G_i) - facebook(F_i)$; for lower-is-better metrics (Grammar errors, Perplexity), $d_i = F_i - G_i$. Effect size $d_z = \bar{d}/s_d$.

Semantic Similarity: Google achieved significantly higher semantic similarity scores than Facebook ($t(322) = 5.38, p < 0.001, d_z = 0.299$; Wilcoxon $W = 1.64 \times 10^4, p < 0.001$). The mean difference of 0.053 (95% CI [0.033, 0.072]) indicates that Google more reliably preserved the meaning of original sentences during back-translation. Grammar Errors: The results also favored Google, with fewer grammar errors on average compared to Facebook ($t(322) = 3.66$, $p < 0.001$, $d_z = 0.204$; Wilcoxon $W = 5189.5$, $p < 0.001$). The mean difference of 0.95 errors (95% CI [0.44, 1.46]) demonstrates a consistent, though modest, grammatical advantage for Google.

BLEU Score: Google outperformed Facebook on BLEU, with a mean difference of 0.031 (95% CI [0.010, 0.051]). The effect was statistically significant ($t(322) = 2.99$, $p = 0.003$, $d_z = 0.166$; Wilcoxon $W = 1.36 \times 10^4$, $p < 0.001$), though the magnitude of the improvement was relatively small. Perplexity: In contrast, no significant differences were observed for perplexity ($t(322) = 0.97$, $p = 0.335$; Wilcoxon $W = 2.25 \times 10^4$, $p = 0.267$). The wide confidence interval ([-882.91, 2584.03]) suggests high variability between sentences, and thus fluency as measured by perplexity appears comparable between the two systems.

Figure 3 illustrates the statistical distribution used to evaluate whether the observed differences between the two back-translation models are statistically significant. The t-distribution curve with degrees of freedom (df = 322) shows the critical values at (±1.97), which represent the rejection region for the null hypothesis at the 0.05 significance level (two-tailed test). The observed t-values for the three significant metrics—semantic similarity (t = 5.38), grammar errors (t = 3.66), and BLEU score (t = 2.99)—all fall beyond these critical boundaries, visually confirming that Google Translate performs significantly better than Facebook's M2M100 model for these measures.

In contrast, the observed value for perplexity (t = 0.97) lies within the acceptance region, supporting the interpretation that there is no statistically significant difference in fluency between the two models. Hence, Figure 3 serves as the visual statistical counterpart to Table 5, enhancing interpretability by visually confirming the statistical outcomes reported in the table.

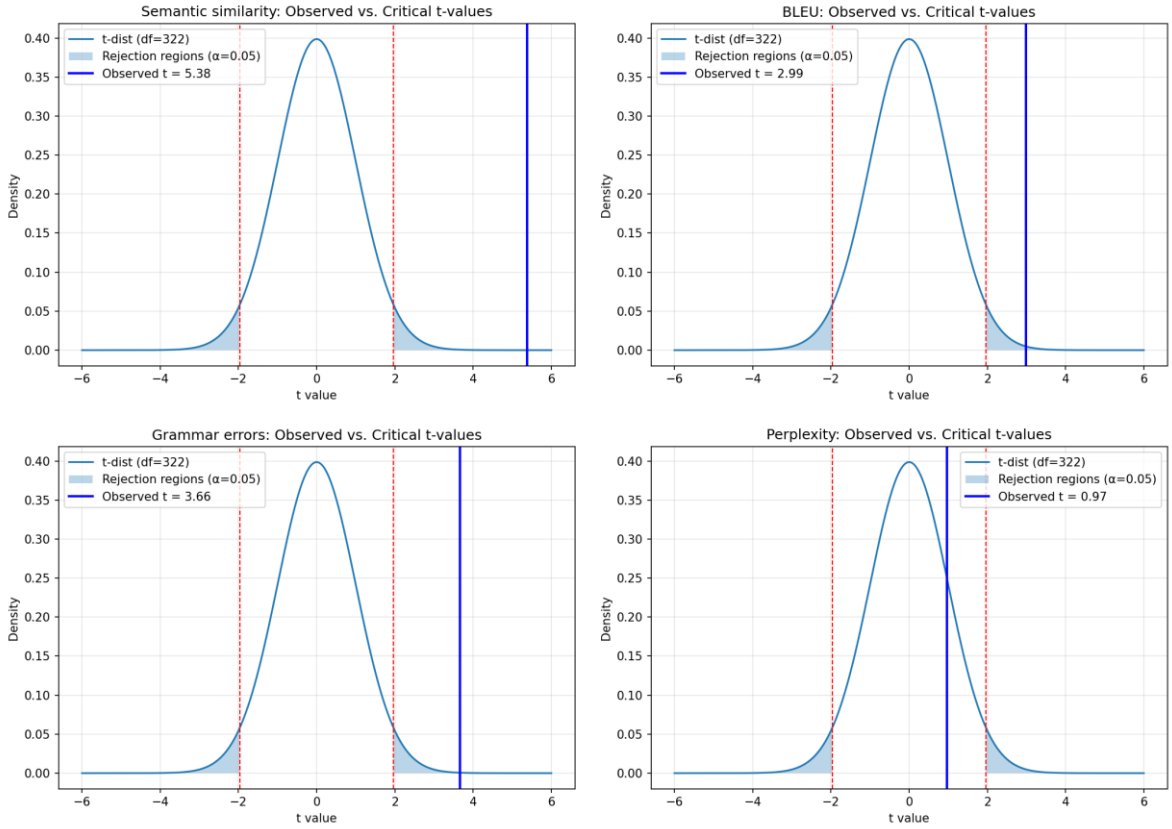


Figure 3. t-distribution with critical (± 1.97) and observed values (df=322)

4.1.2 Data visualization

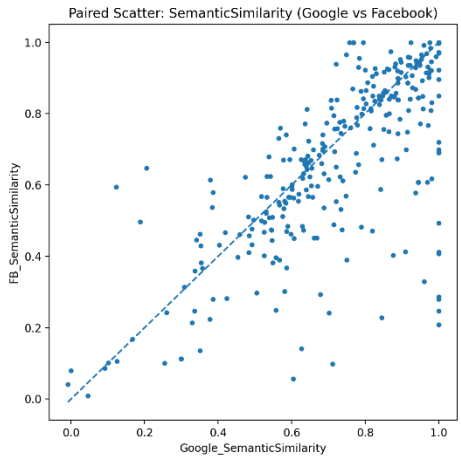
Figure 4 displays paired scatterplots comparing Google Translate and Facebook's m2m-100 back-translation outputs across four quantitative metrics which are semantic similarity, grammar errors, BLEU score, and perplexity. Each point in the scatterplot represents a single sentence-level pair (n = 667), where the x-axis denotes Google's value and the y-axis denotes Facebook's value. The mean scores reported in Table 2 (Google = 0.7258 and Facebook = 0.6731 for semantic similarity) were computed as the arithmetic mean of all sentence-level paired observations within each metric. Formally, for metric M and model j, the mean is defined as

$$\overline{M_j} = \frac{1}{n}\sum_{i=1}^{n} M_{ij}, \qquad (2)$$

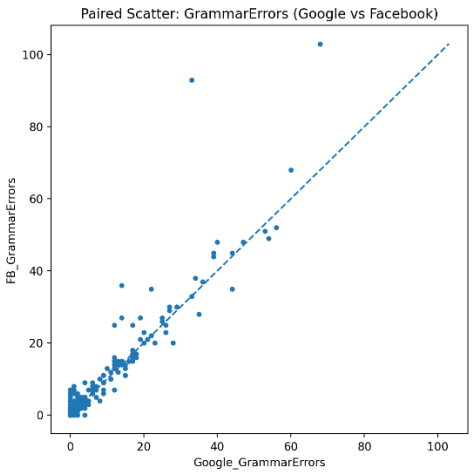
where M_{ij} is the score of sentences i produced by model j (Google or Facebook). This aggregation was performed automatically within the analysis script, which calculated both `mean` and `median` values for each metric using the `numpy.nanmean()` and `numpy.nanmedian()` functions.

The comparative analysis across the four-evaluation metrics further reinforces the superiority of Google Translate over Facebook's M2M100 model in maintaining text quality during back-translation. For semantic similarity, Google achieved a higher mean score (0.7258) compared to Facebook (0.6731), indicating stronger preservation of the original meaning. As illustrated in the scatterplot, the distribution of sentence-level scores is generally balanced around the parity line, yet Google exhibits a consistent advantage, particularly within the mid-to-high similarity range (0.4–0.7). Regarding grammar errors, where lower values denote better performance, Google again outperformed Facebook, recording a lower mean error count (5.8421 vs. 6.7895). The corresponding scatterplot demonstrates that Google consistently produced grammatically cleaner outputs.

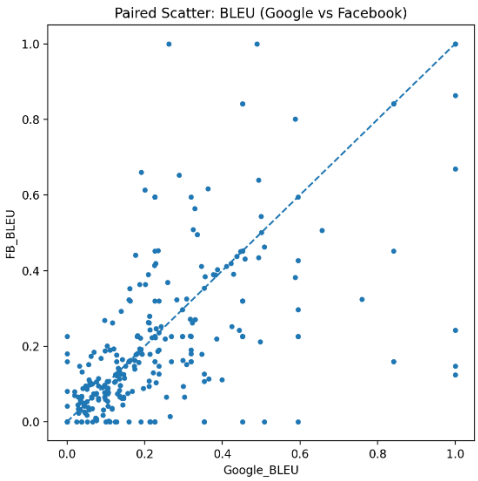
In terms of the BLEU score, Google attained a slightly higher mean (0.2419) relative to Facebook (0.2112), suggesting modest improvements in lexical alignment. Finally, the analysis of perplexity, which measures sentence-level fluency (lower values indicating greater fluency), revealed that Google's translations were substantially more fluent, with a mean perplexity of 732.24 compared to Facebook's 1582.80. Collectively, these findings demonstrate Google's more consistent performance across semantic fidelity, grammatical accuracy, lexical alignment, and textual fluency dimensions.



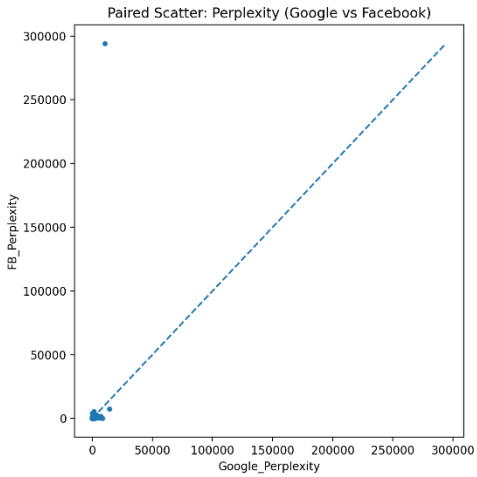
(a) Semantic Similarity (Google vs Facebook)



(b) Grammar Errors (Google vs Facebook)



(c) BLEU (Google vs Facebook)



(d) Perplexity (Google vs Facebook)

Figure 4. Paired scatterplots comparing Google and Facebook back-translation outputs across four metrics: (a) Semantic Similarity, (b) Grammar Errors, (c) BLEU, and (d) Perplexity. Points below the diagonal indicate Google superiority; points above indicate Facebook superiority

5. Result Summary

The combined evidence from statistical testing and data visualization provides a more comprehensive understanding of the comparative performance of Google Translate and Facebook's m2m-100 model. While both methods of analysis point in the same general direction, their emphasis differs and interpreting them together gives a clearer picture of model efficacy. From the empirical results, paired *t*-tests and Wilcoxon signed-rank tests showed that Google achieved significantly better outcomes in semantic similarity, grammar errors, and BLEU scores. These findings suggest that Google more consistently preserved meaning, produced fewer grammatical errors, and achieved closer lexical overlap with the source sentences. Although the effect sizes were small to moderate, the consistency across multiple metrics strengthens the conclusion that Google provides a measurable advantage. For perplexity, however, the statistical tests did not detect a significant difference, indicating that the fluency of generated outputs remains broadly comparable between the two systems.

The data visualizations complement these results by showing the distributional characteristics that underlie the statistical outcomes. Scatterplots revealed that, although Google generally performed better, Facebook remained competitive in certain instances, particularly in semantic similarity where many points clustered close to the parity line. For BLEU, the visual impression of near parity reflects the relatively small numerical advantage observed in the tests, which is expected given the short and noisy nature of user reviews. For perplexity, the mean values suggested a large advantage for Google, but the scatterplots revealed high variability and several outliers. These atypical cases inflated Facebook's average, making the overall difference less reliable as an indicator of fluency. By considering both statistical testing and visualization, we can summarize Google consistently improves semantic preservation, grammatical correctness, and lexical alignment, while both models perform similarly in terms of fluency. Importantly, the dual use of

statistical testing and visualization not only establishes quantitative significance but also highlights areas where performance overlaps or where conclusions should be drawn cautiously. This balance ensures that the findings are both statistically robust and contextually transparent, allowing for a more reliable assessment of back-translation quality.

5.0 CONCLUSIONS

Back-translation is a viable approach for normalizing noisy mobile user reviews. Google Translate consistently outperformed the Facebook m2m-100 model in maintaining semantic fidelity, reducing grammatical errors, and producing more fluent text. Beyond its conventional use in data augmentation, back-translation also acts as a normalization mechanism that enhances the textual clarity of informal user reviews. By translating content to a pivot language and back, BT introduces paraphrased constructs that suppress noise, align colloquial expressions to formal equivalents, and reduce grammatical inconsistencies.

Therefore, while our findings confirm that BT improves the textual clarity of informal user reviews, it is important to acknowledge the limitations of the present study. The experiments were conducted exclusively on back-translated user reviews and did not directly evaluate end-to-end performance within requirements engineering pipelines. As such, the benefits of BT for tasks such as requirement elicitation, traceability, or ambiguity detection remain hypothetical until empirically validated in real RE environments and should be regarded as potential rather than proven. For the futures direction, there is a need for larger datasets to strengthen generalizability, and further work may explore hybrid systems that ensemble Google and Facebook outputs. For instance, ensemble pipelines combining Google Translate's fluency with m2m-100's lexical fidelity may produce more balanced outputs.

ACKNOWLEDGEMENTS

Authors express their sincere gratitude to Universiti Putra Malaysia (UPM) for providing support and resources throughout this research.

AUTHORS CONTRIBUTION

Amran Salleh: Writing–original draft, Visualization, Validation, Methodology, Data curation, Software, Investigation. Mohd Hafeez Osman: Conceptualization, Methodology, Supervision, Project administration, Funding acquisition, Writing–review and editing. Sa'adah Hassan: Supervision, Project administration, Validation, Methodology, Writing–review and editing. Mar Yah Said: Supervision, Project administration, Validation, Methodology, Writing–review and editing.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] Naseem U, Khan SK, Razzak I, Hameed IA. Hybrid words representation for airlines sentiment analysis. In: Australasian Joint Conference on Artificial Intelligence. Cham: Springer International Publishing; 2019. p. 381-92.

[2] Jha N, Mahmoud A. Using frame semantics for classifying and summarizing application store reviews. Empirical Software Engineering. 2018;23(6):3734-67.

[3] Shermamatova GB, Shermamatova ZA, Baxronov AB, Roziqov OV. Usage of colloquial style. Mental Enlightenment Scientific-Methodological Journal. 2024;5(05):278-86.

[4] Al-Obaidi AY, Samawi VW. Opinion mining: Analysis of comments written in Arabic colloquial. In: Proceedings of the World Congress on Engineering and Computer Science. 2016;1.

[5] Genc-Nayebi N, Abran A. A systematic literature review: Opinion mining studies from mobile app store user reviews. Journal of Systems and Software. 2017;125:207-19.

[6] Eftee SY, Khan MY, Noor R, Mahmud H, Hasan MK. Extraction of app problems and its corresponding user action from user review of apps using few-shot learning. SSRN. Available from: https://ssrn.com/abstract=5093717

[7] Szeto MD, Barber C, Ranpariya VK, Anderson J, Hatch J, Ward J, et al. Emojis and emoticons in health care and dermatology communication: Narrative review. JMIR Dermatology. 2022;5(3):e33851.

[8] Aslam A, Hussian B. Emotion recognition techniques with rule-based and machine learning approaches. arXiv preprint. 2021;arXiv:2103.00658.

[9] Noori B. Classification of customer reviews using machine learning algorithms. Applied Artificial Intelligence. 2021;35(8):567-88.

[10] Sangeetha J, Kumaran U. Sentiment analysis of Amazon user reviews using a hybrid approach. Measurement: Sensors. 2023;27:100790.

[11] Ahmad M, Aftab S, Ali I, Hameed N. Hybrid tools and techniques for sentiment analysis: A review. International Journal of Multidisciplinary Science and Engineering. 2017;8(3):29-33.

[12] Subedi IM, Singh M, Ramasamy V, Walia GS. Application of back-translation: A transfer learning approach to identify ambiguous software requirements. In: Proceedings of the 2021 ACM Southeast Conference. 2021. p. 130-7.

[13] Fan A, Bhosale S, Schwenk H, Ma Z, El-Kishky A, Goyal S, et al. Beyond English-centric multilingual machine translation. Journal of Machine Learning Research. 2021;22(107):1-48.

[14] Chew E, Chakraborti M, Weisman W, Frey S. Machine translation for accessible multi-language text analysis. Computational Communication Research. 2025;7(1):1.

[15] Mukherjee A, Shrivastava M. Lost in translation? Found in evaluation: A comprehensive survey on sentence-level translation evaluation. ACM Computing Surveys. 2025.

[16] Mathur N, Baldwin T, Cohn T. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. arXiv preprint. 2020;arXiv:2006.06264.

[17] Klotz AC, Swider BW, Kwon SH. Back-translation practices in organizational research: Avoiding loss in translation. Journal of Applied Psychology. 2023;108(5):699.

[18] Poh S, Yang SJ, Tan J, Chieng L, Tan J, Yu Z, et al. MalayMMLU: A multitask benchmark for the low-resource Malay language. In: Findings of the Association for Computational Linguistics: EMNLP 2024. 2024. p. 650-69.

[19] Ma J, Li L. Data augmentation for Chinese text classification using back-translation. In: Journal of Physics: Conference Series. 2020;1651(1):012039.

[20] Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: A method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002. p. 311-8.

[21] Lee J, Kim J, Kang P. Back-translated task adaptive pretraining: Improving accuracy and robustness on text classification. arXiv preprint. 2021;arXiv:2107.10474.

[22] Edunov S, Ott M, Auli M, Grangier D. Understanding back-translation at scale. arXiv preprint. 2018;arXiv:1808.09381.

[23] Tong Y, Chen Y, Zhang G, Zheng J, Zhu H, Shi X. Generating diverse back-translations via constraint random decoding. In: China Conference on Machine Translation. Singapore: Springer; 2021. p. 92-104.

[24] Chen B, Golebiowski J, Abedjan Z. Data augmentation for supervised code translation learning. In: Proceedings of the 21st International Conference on Mining Software Repositories. 2024. p. 444-56.

[25] Prenner JA, Robbes R. Making the most of small software engineering datasets with modern machine learning. IEEE Transactions on Software Engineering. 2021;48(12):5050-67.

[26] Brigato L, Iocchi L. A close look at deep learning with small data. In: 25th International Conference on Pattern Recognition. IEEE; 2021. p. 2490-7.

[27] Shadman R, Murshed MS, Verenich E, Velasquez A, Hussain F. The utility of feature reuse: Transfer learning in data-starved regimes. In: 2023 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE; 2023. p. 37-42.

[28] Feldman I, Coto-Solano R. Neural machine translation models with back-translation for the extremely low-resource indigenous language Bribri. In: Proceedings of the 28th International Conference on Computational Linguistics. 2020. p. 3965-76.

[29] McNamee P, Duh K. An extensive exploration of back-translation in 60 languages. In: Findings of the Association for Computational Linguistics: ACL 2023. 2023. p. 8166-83.

[30] Liu S, Zhu W. An analysis of the evaluation of the translation quality of neural machine translation application systems. Applied Artificial Intelligence. 2023;37(1):2214460.

[31] Song J, Zan H, Liu T, Zhang K, Ji X, Cui T. Text classification based on multilingual back-translation and model ensemble. In: China Health Information Processing Conference. Singapore: Springer; 2023. p. 231-41.

[32] Yadav A, Patel A, Shah M. A comprehensive review on resolving ambiguities in natural language processing. AI Open. 2021;2:85-92.

[33] Boussougou MKM, Hamandawana P, Park DJ. Enhancing voice phishing detection using multilingual back-translation and SMOTE: An empirical study. IEEE Access. 2025.

[34] Do QM, Zeng K, Paik I. Resolving lexical ambiguity in English-Japanese neural machine translation. In: Proceedings of the 3rd Artificial Intelligence and Cloud Computing Conference. 2020. p. 46-51.

[35] Kiyomoto S, Hidano S, Nguyen-Son HQ, Phuong TT. Detecting machine-translated text using back translation. In: Proceedings of the 12th International Conference on Natural Language Generation. 2019.

[36] Benmansour M, Hdouch Y. The role of the latest technologies in the translation industry. Emirati Journal of Education and Literatures. 2023;1(2):31-6.

[37] Castilho S, Moorkens J, Gaspari F, Calixto I, Tinsley J, Way A. Is neural machine translation the new state of the art? The Prague Bulletin of Mathematical Linguistics. 2017;(108).

[38] Wongso W, Joyoadikusumo A, Buana BS, Suhartono D. Many-to-many multilingual translation model for languages of Indonesia. IEEE Access. 2023;11:91385-97.