INFORMATION TECHNOLOGY

Batak Toba and Batak Angkola texts written in scriptio continua form without spaces are the object of this study. The work solves a low-resource variety classification problem where the two varieties are similar and missing word boundaries introduce segmentation noise. A hybrid TRIE-BERT pipeline was developed in which trie automation performs deterministic spacing, the restored spacing is fixed, and the spaced text becomes a stable input interface for a Bidirectional Encoder Representations from Transformers (BERT) classifier. Experiments used a Batak lexicon of 19,070 word entries and 8,000 sentences, 4,000 per variety, evaluated under four data schemes from 1,000 to 8,000 sentences and five epoch settings from 5 to 50 with an 80:20 split. After lexicon recalibration of about 70 sentences, spacing reached 98 percent accuracy. The best setting at 8,000 sentences and 50 epochs achieved 0.85 test accuracy with 0.343 training loss, 0.85 ROC AUC, and 0.85 F1-score, exceeding a long short-term memory recurrent neural network baseline (LSTM-RNN) at 0.80 accuracy, 0.397 loss, 0.803 ROC AUC, and 0.80 F1-score. Class-wise evaluation yielded precision 0.81 and recall 0.92 for Toba and precision 0.90 and recall 0.79 for Angkola, explaining averaged precision 0.86 and recall 0.85. The improvement is associated with the combined use of deterministic trie-based boundary recovery and contextual BERT classification, where spacing is fixed before classification to reduce token ambiguity and stabilize the input structure. The results support Batak text processing pipelines that require automatic spacing and variety detection under limited labels, provided lexicon coverage is maintained and spelling variation is controlled

Keywords: trie, BERT, scriptio continua, low resource, Batak language

UDC 004.912
DOI: 10.15587/1729-4061.2026.352682

DEVELOPMENT OF A TRIE-BERT PIPELINE FOR AUTOMATIC SPACING AND LOW RESOURCE LANGUAGE CLASSIFICATION IN BATAK TOBA AND ANGKOLA SCRIPTIO CONTINUA TEXTS

Muhammad Anggia Muchtar
Corresponding Author
PhD*
E-mail: anggi.muchtar@usu.ac.id
ORCID: https://orcid.org/0000-0002-9020-890X
Opim Salim Sitompul
PhD*
ORCID: https://orcid.org/0000-0001-6069-1841
Maya Silvi Lydia
PhD**
ORCID: https://orcid.org/0009-0006-5779-5678
Syahril Efendi
PhD**
ORCID: https://orcid.org/0000-0002-3944-5459
*Department of Information Technology***
Department of Computer Science*
***Universitas Sumatera Utara
Dr. T. Mansur str., 9, Padang Bulan, North Sumatera, Indonesia, 20155

Received 21.01.2026
Received in revised form 10.02.2026
Accepted date 26.03.2026
Published date 30.04.2026

How to Cite: *Muchtar, M. A., Sitompul, O. S., Lydia, M. S., Efendi, S. (2026). Development of a trie-bert pipeline for automatic spacing and low resource language classification in Batak Toba and Angkola scriptio continua texts. Eastern-European Journal of Enterprise Technologies, 2 (2 (140)), 6–16. https://doi.org/10.15587/1729-4061.2026.352682*

1. Introduction

Deep learning has significantly advanced natural language processing (NLP) by enabling stronger models for both foundational language processing and widely used applications such as text classification and information extraction [1]. In today's digital environments, massive streams of textual content are produced through documents, web pages, and institutional information systems, so automated text processing is increasingly required for indexing, retrieval, and reliable categorization. For real-world deployment, these pipelines must remain robust across heterogeneous writing conventions and resource constraints, because downstream analytics typically presuppose consistent and machine-usable textual structure.

Large pre-trained transformer models have reshaped modern NLP by improving contextual representation learning and pushing performance across a wide range of tasks [2]. A complementary synthesis in the pre-trained-model literature explains that the emergence of pre-trained models has moved NLP into a new era and emphasizes how pre-training on large corpora supports adaptation to downstream tasks [3]. In parallel, trie-based string dictionary structures remain practically valuable when the system must enforce fast, consistent boundary decisions through lexicon-guided matching. This combination motivates a hybrid design in which trie-driven automatic spacing supplies stable word boundaries before a contextual encoder performs higher-level discrimination.

Low-resource language settings continue to face persistent performance limitations because high-quality large-scale data resources are often unavailable, a constraint that has been documented in survey work on low-resource neural approaches [4]. This condition closely matches the Batak Toba and Batak Angkola context described in the dissertation, where the absence of an established corpus and

limited digital resources complicate model development and evaluation. At the same time, research on similar languages, varieties, and dialects highlights that closely related varieties create recurring challenges for computational processing and that language or dialect identification becomes non-trivial under high similarity [5]. In Batak Toba and Batak Angkola, the closeness between varieties amplifies ambiguity in classification, especially when training data are limited and orthographic conventions are inconsistent.

Whitespace correction and word segmentation studies indicate that missing or misplaced spaces can substantially degrade downstream processing, which motivates methods that recover boundaries in noisy or non-standard text [6]. In the Batak Toba and Batak Angkola setting, the scriptio continua condition creates a structural bottleneck because token boundaries are not explicit, and boundary uncertainty can propagate into classification when the target varieties are highly similar. This combined constraint motivates a hybrid solution in which lexicon-guided automatic spacing establishes stable word boundaries before contextual modeling is applied for finer-grained discrimination under limited data conditions.

Batak Toba and Batak Angkola scriptio continua texts represent a relevant scientific problem because modern text-processing systems require analyzable word boundaries, while closely related varieties remain difficult to distinguish under resource-constrained conditions. In this setting, missing spaces are not a minor orthographic issue, because boundary uncertainty can propagate into later classification and reduce the reliability of downstream language processing. The topic is also important in practice because digital Batak materials drawn from dictionaries, online texts, and cultural resources require preprocessing that can support more consistent indexing, retrieval, and language-sensitive analysis. Therefore, studies devoted to automatic spacing and low-resource variety classification for Batak Toba and Angkola scriptio continua texts are of scientific and practical relevance.

2. Literature review and problem statement

Paper [7] presents a backtracking-based solution to scriptio continua in Javanese manuscript transliteration and shows that greedy and brute-force variants achieve the same reported accuracy, while the greedy variant is more efficient. Paper [8] does not address spacing as a standalone text-processing task, but it shows that prefix-trie constraints can guide decoding effectively without retraining the underlying model. These studies indicate that lexicon-guided or trie-guided mechanisms are useful for constraining candidate formation when explicit boundaries are absent. Yet boundary recovery remains sensitive to lexical coverage, spelling variation, and out-of-vocabulary forms, so dictionary control alone is not sufficient for stable segmentation in all cases.

Paper [9] shows that hybrid contextual models can improve Arabic dialect identification by combining transformer representations with BiLSTM on social media data. Paper [10] extends this direction to a low-resource setting through cross-lingual transfer from Modern Standard Arabic, while Paper [11] shows that zero-shot transfer from MSA to dialectal Arabic is weak unless adaptation mechanisms are introduced. Paper [12] further confirms that stronger contextual architectures can improve dialect classification by combining transformer models in a stacking framework. However, these studies assume that the input text is already available in analyzable form, so they strengthen variety discrimination but do not resolve the prior word-boundary problem that arises in scriptio continua Batak texts.

Paper [13] shows that pretrained transformer encoding is effective for Korean automatic spacing and reports the highest character accuracy among the evaluated models, with pretraining contributing more than additional fine-tuning data alone. Paper [14] shows that dictionary-based analysis can be improved by transformer-based reranking when several candidate analyses are available, although the authors also note that the full system is not yet optimized for real-time processing. Paper [15] adds an important caution for low-resource segmentation by showing that model ranking can become unstable when conclusions are drawn from a single dataset. A related boundary-recovery challenge appears in Paper [16], which addresses word segmentation from raw speech using an instance lexicon and shows that segmentation remains difficult when explicit delimiters are absent. Taken together, these studies suggest that reliable boundary recovery depends not only on the segmentation model itself, but also on preprocessing stability, data conditions, and evaluation design.

Paper [17] studies scriptio continua word segmentation for ancient Tamil inscriptions using an N-gram Naive Bayes model and reports strong segmentation performance, while also noting that broader robustness requires larger and more varied historical data. In the Batak context, paper [18] shows that Trie Automation can restore spaces with high accuracy on 4,000 scriptio continua sentences, but the same study states that performance remains dependent on corpus completeness. These results confirm that deterministic boundary recovery is feasible for scriptio continua texts when lexical support is adequate. At the same time, they also indicate that spacing alone does not fully solve the downstream problem when the final task is to distinguish closely related varieties.

The reviewed studies show that dictionary-driven and trie-guided methods can recover boundaries efficiently, while contextual models can improve classification among similar varieties. The unresolved problem is that existing studies do not clearly integrate these two strengths into a single pipeline for Batak Toba and Batak Angkola scriptio continua texts, where missing spaces and high lexical similarity interact directly. All this allows to conclude that it is advisable to conduct research devoted to the development of a TRIE-BERT pipeline for automatic spacing and low-resource language classification in Batak Toba and Angkola scriptio continua texts [19–21].

3. The aim and objectives of the study

The aim of the study is to develop a Trie-BERT pipeline for automatic spacing and low-resource variety classification in Batak scriptio continua data, with Batak Toba and Angkola as the target varieties. The expected outcome is a reproducible two-stage workflow in which trie-based spacing restores word boundaries first and contextual classification is then performed on the resulting text.

To achieve the aim, the following objectives were set:

– to develop a trie-based automatic spacing module and specify a lexicon management procedure for scriptio continua Batak texts;

– to implement a TRIE-BERT based classification component and a stable interface between the spacing output and the classifier input;

– to experimentally compare TRIE-BERT and LSTM-RNN for Batak Toba and Angkola data, including spacing quality and classification performance under a consistent protocol.

4. Materials and methods

4. 1. The object and hypothesis of the study

The object of the study is Batak Toba and Batak Angkola scriptio continua text treated as low-resource input without explicit word boundaries. The main hypothesis is that reliable boundary recovery is required before downstream variety classification, and that preserving the restored word structure can improve the consistency of subsequent classification under a two-stage experimental design. The method is simplified to a fixed pipeline in which spacing is performed first and classification is applied to the resulting text.

4. 2. Batak language corpus construction

Dictionary-based solutions for scriptio continua show that boundary recovery is strongly influenced by lexicon coverage, so corpus construction is treated here as a primary methodological stage. The Batak corpus in this study is a curated collection of sentences and word forms used as the source for spacing and classification. A target of more than 5,000 entries for each of Batak Angkola and Batak Toba was defined so that trie matching would rely on a lexicon of at least 10,000 entries, and the final corpus contains 19,070 word entries together with 8,000 sentences, evenly divided across the two varieties.

Online sentence materials were collected from validated Batak-language websites, including proverb and expression compilations, a bilingual Toba–Indonesian Bible resource, and a Batak daily-conversation source. Printed lexicon entries and example sentences were taken from a Batak Toba-Indonesian dictionary and an Angkola-Indonesian dictionary. Web-derived materials were normalized and cleaned to remove collection artifacts, while dictionary-derived entries were normalized to preserve lexical form before integration into the corpus. In this way, all sources were brought into a single sentence-level inventory for later processing.

Printed materials were digitized with Google Lens to accelerate sentence-by-sentence transcription from photographed pages. This step improves collection efficiency, but it can also introduce recognition noise when page quality, character clarity, or layout conditions are imperfect. To control this risk, extracted text was consolidated into editable documents, checked during normalization, and then segmented into individual sentence records before use. For online sources, text was collected directly from editable digital materials and separated into sentences without image-based transcription.

Corpus validation in the present study is operational rather than annotator-based. The collected materials were normalized, consolidated, and then assessed through the spacing stage on 4,000 processed scriptio continua sentences, where approximately 70 mismatched outputs were identified and corrected before the spaced corpus was used in classification. This procedure does not eliminate all possible source errors, but it controls obvious transcription and preprocessing problems within the current experiment. At the same time, the resulting corpus should be understood as a controlled experimental resource rather than a complete representation of all Batak text conditions, genres, or orthographic variants. Trie-based segmentation therefore remains dependent on lexicon completeness, and unseen forms or spelling variation can still reduce spacing quality.

4. 3. Construction of training and testing data using trie automation to address scriptio continua text

A trie can be viewed as an n-ary tree in which each node represents an N-place vector, which components correspond to symbols (characters) or numbers. A node at depth h represents the set of all keys that share the same prefix of length h, and the outgoing branches are determined by the next symbol (the $(h + 1)$-th character). To avoid ambiguity between keys such as “the” and “then”, a special end marker (e.g., “#”) can be appended to every key so that no key becomes a prefix of another key. In a trie, each path from the root to a leaf corresponds to exactly one key in the set K, and each internal node corresponds to a prefix of some key in K. In this sense, a trie functions as a pattern-matching structure over a finite keyword set and can be formalized using a finite-state view.

The trie-based matcher can be described as follows:

1. S is a finite set of states (nodes).
2. I is a finite set of input symbols (characters).
3. g: $S \times I \rightarrow S\ U$ {fail} is the goto function.
4. The initial state in S is denoted by state 1.
5. F subset S is a finite set of accepting states.

A state Y belongs to F if and only if the path from the initial state to Y spells a string X in K. A transition labeled a in I from state Y to state t indicates that $g(Y, a) = t$, while the absence of a transition indicates failure. The number of outgoing links from a states is denoted by OUTDEGREE(s); a state with OUTDEGREE(s) = 0 is a terminal state. Let I^* denote the set of all strings over I, including the empty string e. The goto function can be extended to strings by defining $g(r, e) = r$ and $g(r, ax) = g(g(r, a), x)$, where a is in $I\ U$ {“#”} and x is in $(I\ U$ {“#”})*. Under this definition, trie traversal proceeds character by character, and segmentation decisions are produced through deterministic matching along trie paths as shown in Fig. 1.

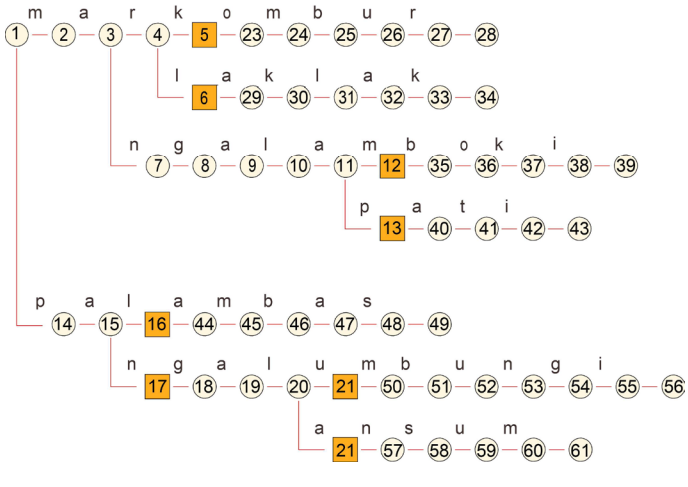

Fig. 1. Illustrates the trie structure as applied to Batak language data

Within the present study, trie traversal is used to segment continuous Batak strings through deterministic lexicon matching along character paths. The resulting spaced sentences are then stored as preprocessing output and used as input for the subsequent classification stage. The next subsection describes the BERT component that operates on this spaced input.

4. 4. Batak language in bi-directional encoder representation from transformers pre-training deep modeling

Transformer encoders are widely used to construct contextual token representations that model bidirectional dependencies within a sequence, making them suitable as a representation engine for downstream NLP tasks [22]. For scriptio continua settings, where explicit word boundaries are absent, robust representation learning is typically coupled with segmentation or spacing to prevent error propagation into later stages [17]. In low-resource language identification, pretrained transformer models have also demonstrated practical benefits by transferring knowledge from larger corpora to scarce-labeled settings [10]. Accordingly, after the Trie-based automation stage (automatic spacing), this study uses a BERT-style encoder as the second stage of the Trie-BERT pipeline to capture contextual cues for Batak Toba and Batak Angkola. The encoder is treated as a reusable backbone that can be connected to different task-specific heads without changing the core architecture. This subsection focuses on the representation design and inference procedure adopted to obtain model outputs in a controlled and reproducible manner.

Input sequences are constructed using subword tokenization (WordPiece), where a special classification token [CLS] is placed at the beginning and sentence boundaries are marked with [SEP], consistent with standard BERT-style pipelines [23]. To support sentence-level tasks, the contextual vector associated with [CLS] is used as a compact sequence representation, while token-level vectors can be routed to sequence-labeling heads when needed. As illustrated in Fig. 3, the final input embedding for each token is formed by combining token embeddings with segment indicators (sentence A vs. sentence B) and positional information that preserves order within the sequence. Positional embeddings are particularly important because self-attention alone is permutation-invariant and requires explicit order signals to distinguish sequences with the same tokens but different arrangements [24]. In this work, sentence pairs are packed into a single sequence to enable both single-sentence and paired-sentence processing under a unified representation scheme. The same representation layout is retained across training and inference so that subsequent evaluation reflects the same input encoding assumptions.

BERT-style models are commonly trained through a two-stage procedure that separates self-supervised pre-training from task-specific fine-tuning [2, 3]. During pre-training, masked language modeling (MLM) learns to recover masked tokens from bidirectional context, while next sentence prediction (NSP) models the coherence relationship between two segments. For downstream tasks, fine-tuning updates the pretrained parameters end-to-end using labeled data and a lightweight output head, which enables efficient adaptation to classification problems under limited supervision. When sentence-pair inputs are used, a standard packing format such as [CLS] A [SEP] B [SEP] can be applied, with segment indicators to help the model distinguish the two sentences (Fig. 2). The contextual vectors produced by the encoder can then be mapped to task outputs through simple projection layers, including token-level prediction via softmax for MLM-style decoding and sequence-level classification driven by the [CLS] representation. Fig. 3 summarizes the overall BERT scheme used in this study, covering the masked-token recovery mechanism and the sentence-pair coherence signal used during training. The joint training of these objectives supports stable contextual representations that are subsequently reused during fine-tuning and masked inference.

In practical transformer workflows, pretrained BERT-based models are frequently operationalized through standard libraries (e.g., Hugging Face) for controlled fine-tuning and inference in applied settings [25]. To implement next-word prediction in this study, masked-token inference is employed by appending a mask token at the end of the input string and requesting the model to predict the most likely token for the masked position. The first step loads a pretrained masked-language model and its tokenizer in evaluation mode; the second step injects a mask token at the end of the input sequence. The third step encodes the input into input_ids, locates the mask index, and performs a forward pass without gradient updates; the fourth step decodes the top-k candidates by filtering non-lexical symbols and merging subword fragments. Algorithm in BERT model provides the encoder-decoder wrapper used to execute these steps. This procedure is kept minimal to ensure that the reported results reflect model behavior rather than extensive engineering heuristics.

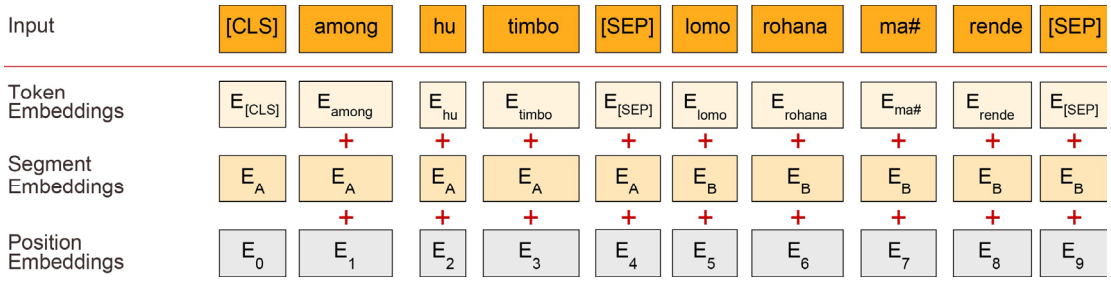

Fig. 2. Illustration of bi-directional encoder representation from transformers embeddings applied to Batak text (token, segment, and positional embeddings)

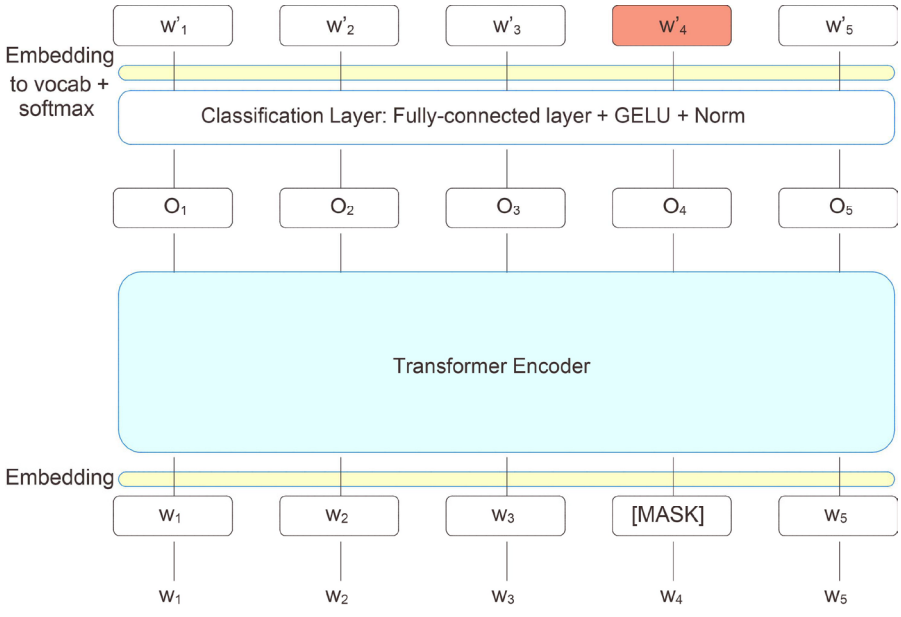

Fig. 3. Architecture of the bi-directional encoder representation from transformers scheme (sentence-level prediction)

4. 5. Long short-term memory recurrent neural nete work

To produce a Batak language-type detection scheme based on a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN), this study employs an LSTM architecture in which each node in the hidden layer is replaced by an LSTM memory unit [26]. An LSTM unit consists of an input gate i(t), an output gate o(t), a forget gate f(t), and a memory cell state c(t). These gates control which information is retained, updated, or discarded as it flows through the network via the cell state as shown in Fig. 4.

In general, the LSTM must decide which information should be removed from the previous cell state $c(t-1)$. This decision is governed by the forget gate $f(t)$, implemented using a sigmoid activation function that takes the previous hidden output $h(t-1)$ and the current input $x(t)$, producing values in the range [0, 1], where a value close to 1 indicates retention of information and a value close to 0 indicates deletion

$$f(t) = \delta(Wf\,[x(t), h(t-1)] + bf), \quad (1)$$

where $x(t)$ denotes the input vector at time step t, $h(t-1)$ is the hidden output from the previous time step, and $f(t)$ is the forget-gate activation. The symbol $\delta(\cdot)$ denotes the sigmoid activation function, Wf is the forget-gate weight matrix, and bf is the corresponding bias term. The previous memory content is denoted by $c(t-1)$.

Next, the input gate determines which parts of the cell state will be updated, while a tanh layer generates candidate values to be added to the state

$$i(t) = \delta(Wi\,[x(t), h(t-1)] + bi), \quad (2)$$

where $i(t)$ denotes the input-gate activation and $\tilde{c}(t)$ denotes the candidate cell content generated for possible insertion into the memory state. Wi and Wc are the weight matrices associated with the input gate and the candidate-state transformation, while bi and bc are the corresponding bias terms. The function $\tanh(\cdot)$ denotes the hyperbolic tangent activation

$$\tilde{c}(t) = \tanh(Wc[x(t) + h(t-1)] + bc). \quad (3)$$

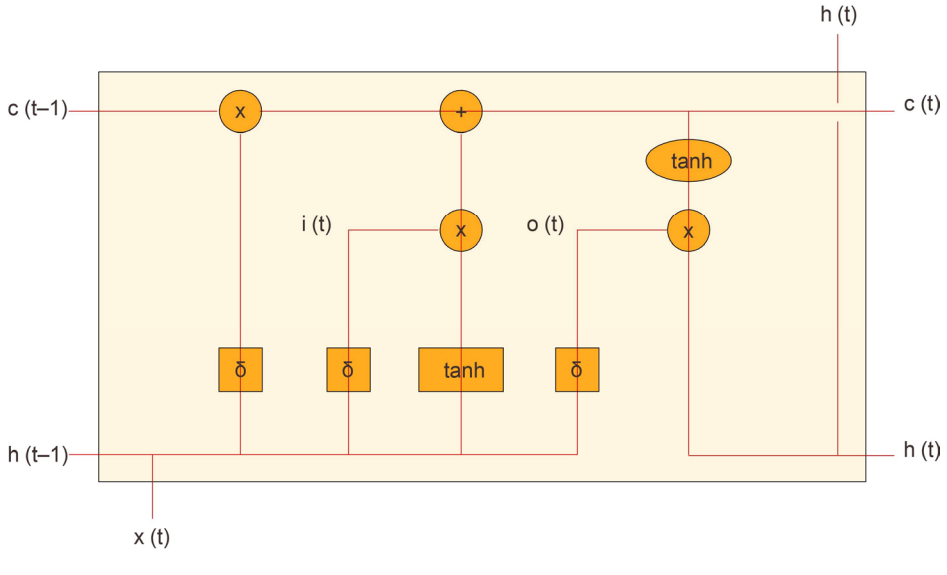

Fig. 4. Structure of the LSTM-RNN memory unit

In the updated memory expression, $c(t)$ denotes the new cell state after the previous state and the candidate content have been combined through the forget and input gates. For the output stage, $o(t)$ denotes the output-gate activation, Wo is the output-gate weight matrix, bo is the corresponding bias term, and $h(t)$ is the hidden output produced at time step t

$$c(t) = f(t) \times c(t-1) + i(t) \times \tilde{c}(t). \quad (4)$$

For the output gate $o(t)$, a sigmoid activation function is used to determine which parts of the current cell state contribute to the output. The filtered cell state is then passed through tanh to produce the hidden output:

$$o(t) = \delta(Wo[x(t), h(t-1)] + bo), \quad (5)$$

$$ht = ot \tanh c(t). \quad (6)$$

Wf, Wi, Wc, and Wo denote the weight matrices associated with the respective gates and the candidate state transformation, while bf, bi, bc, and bo represent the corresponding bias terms.

The overall structure of the LSTM-RNN model used in this study is presented in Fig. 5. The above computations are repeated for each time step in the input sequence. During training, the model parameters (weights and biases) are updated by minimizing the discrepancy between the LSTM outputs and the training samples.

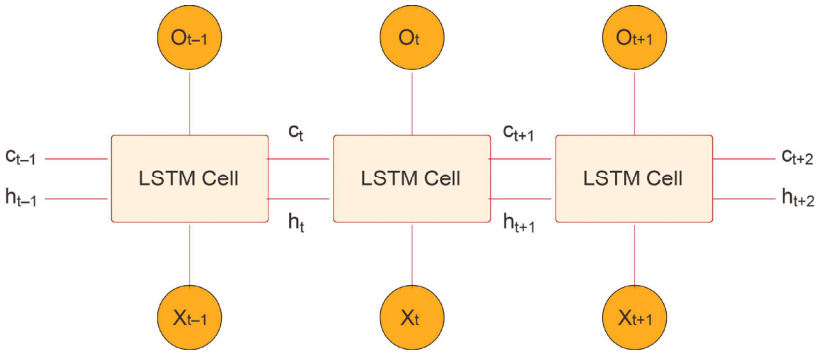

Fig. 5. Structure of the long short-term memory recurrent neural network model

5. Results of the TRIE-BERT pipeline validation

5. 1. Results of trie-based automatic spacing

The first objective was addressed by developing a trie-based automatic spacing module for Batak scriptio continua texts. As shown in Fig. 6, raw data from Batak dictionaries and online sentence sources are consolidated into a Batak corpus and lexicon before deterministic spacing. This design is consistent with scriptio continua studies showing that dictionary-based restoration depends on lexical support [7]. The spaced output is then forwarded to the classifier, and the downstream evaluation is carried out in a low-resource variety-identification setting [10]. The same spaced input is also retained for the comparative LSTM-RNN baseline.

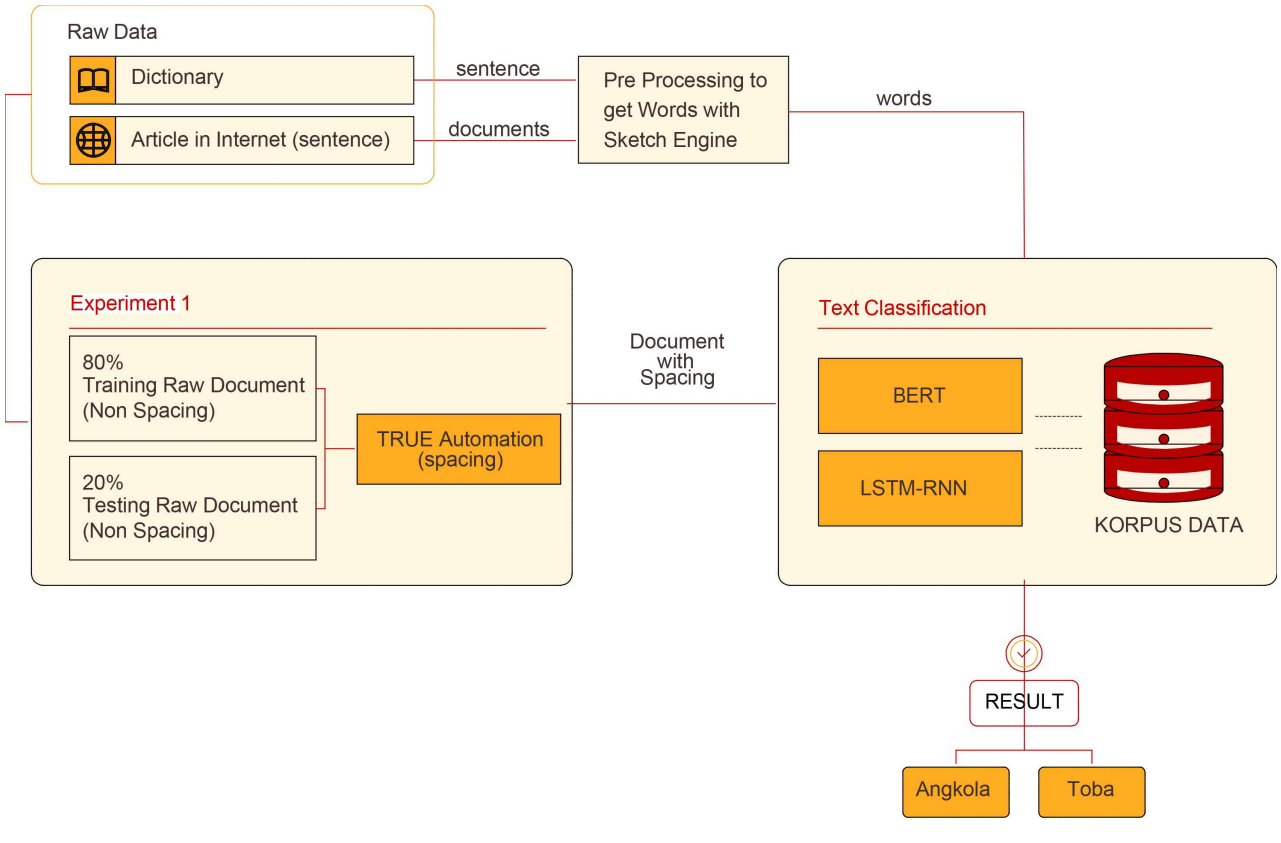

Fig. 6. Architecture of the TRIE-BERT pipeline for automatic spacing

To support the spacing module, a Batak corpus was constructed from internet articles and printed dictionaries, where web sources were collected via crawling and dictionary content was processed through text parsing and text cleaning. The resulting corpus contains 19,070 words entries, comprising 8,372 Batak Angkola words and 10,698 Batak Toba words, and it also includes 4,000 sentences for Batak Angkola and 4,000 sentences for Batak Toba. The sentences are initially in *scriptio continua* form (without spaces), and representative input-output examples produced by trie automation are provided in Table 1. These examples illustrate how the spacing module restores word boundaries by transforming continuous character strings into spaced sequences that can be used in subsequent modeling stages. To ensure a controlled experimental pipeline, all sentences are first processed by trie automation and only then passed to the language classifier. This spacing stage therefore functions as the mandatory preprocessing component that enables downstream training and testing.

Table 1

Input and output sentences to be processed in Batak and Angkola language

Input	Output
taon1986maahusorangtuportibion	taon 1986 ma ahu sorang tu portibi on
disadajabunametmetdibagasankeluarganatungmansaimarhahurangandopeditingkii	di sada jabu na metmet di bagasan keluarga na tung mansai marhahurangan dope di tingki i
natorashupartanihuhutmartigatigadengke	natorashu partani huhut martigatiga dengke
bernitjaladangoldotaheparngoluonnami	bernit jala dangol do tahe parngoluon nami
ahuma borusiampudansianonommarhamaranggi/marpinaribot	ahu ma boru siampudan sian onom marhamaranggi/marpinaribot
tadingmahamidisadahutanamargoarnarumontak	tading ma hami di sada huta na margoar narumontak
sadahutasianbagiannikecamatanporseakabupatentobasadisumaterautara	sada huta sian bagian ni kecamatan porsea kabupaten tobasa di sumatera utara
moloangkanatoropdihutaonmartanidoulaonna	molo angka natorop di huta on martani do ulaonna
adongdonadebahuhutmarporlakdiangkaluatmananghutanaasing	adong do nadeba huhut marporlak di angka luat manang huta na asing
songontuhutanalelasihorbomanangtusibisa	songon tu huta nalela sihorbo manang tu sibisa

To examine how dataset size and optimization length influence model behavior, four experimental schemes are defined and applied consistently after spacing. Scheme 1 uses 1,000 sentences with 800 for training and 200 for testing, while Scheme 2, Scheme 3, and Scheme 4 use 3,000 (2,400/600), 5,000 (4,000/1,000), and 8,000 (6,400/1,600) sentences, respectively, all with a 50:50 balance between Batak Angkola and Batak Toba. Each scheme is trained under five epoch settings (5, 10, 15, 30, and 50) so that performance trends can be observed under both short and long optimization schedules. For trie automation itself, spacing accuracy is measured on 4,000 processed sentences, and the procedure reports 98% accuracy with approximately 70 sentences requiring correction. A recalibration step is then performed by correcting the identified cases before the spaced sentences are used as inputs to the subsequent BERT stage. This workflow keeps the spacing stage fixed, so the subsequent classification results reflect model behavior rather than variation in boundary reconstruction.

5. 2. Results of TRIE-BERT classification across experimental schemes

The classification behavior of TRIE-BERT is summarized through confusion matrices produced across all experimental schemes, as shown in Fig. 7. The matrix structure allows the results to be interpreted in terms of true positives, true negatives, false positives, and false negatives by comparing predicted labels against the ground truth labels. To illustrate the interpretation procedure, the text provides an example for testing data 1,600 and epoch 50, where correct and incorrect predictions can be read directly from the corresponding cells in the matrix. Under this example, the number of misclassified samples is obtained by summing the off-diagonal counts (e.g., 167 and 68), yielding the total number of incorrect predictions for that configuration. Across schemes, the matrices indicate that some classification errors persist, which is consistent with the limited data scale and the high similarity between the two varieties in the evaluated setting. The confusion-matrix view is therefore used as the primary diagnostic artifact to connect per-configuration outcomes to error patterns.

Quantitative evaluation is conducted using a set of criteria defined in the text and reported consistently across schemes. The first metric is test accuracy, which captures the fraction of correctly classified samples in the testing set. The second metric is training loss, computed via cross-entropy where the model prediction is compared against the probabilistic label representation. The third metric is ROC AUC, which is reported on a 0–1 scale to reflect the model's separability behavior across thresholds. The final metric is the F1-score using precision and recall, so that performance is summarized in a way that is sensitive to both false positives and false negatives. Together, these metrics support a more complete view than accuracy alone, while keeping reporting consistent across all configurations. The aggregated outcomes of the experimental runs are reported in Table 2, which presents test accuracy, train loss, ROC AUC, F1-score, precision, and recall for all combinations of dataset size and epoch setting.

Table 2

Evaluation criteria results for TRIE-BERT

Data size	Epoch	Test accuracy	Train loss	ROC AUC	F1-score	Precision	Recall
1000	5	0.66	0.622	0.655	0.65	0.67	0.66
3000		0.78	0.485	0.781	0.78	0.81	0.78
5000		0.77	0.496	0.771	0.77	0.78	0.77
8000		0.82	0.41	0.82	0.82	0.84	0.82
1000	10	0.72	0.555	0.72	0.72	0.72	0.72
3000		0.8	0.448	0.798	0.79	0.83	0.8
5000		0.82	0.398	0.823	0.82	0.84	0.82
8000		0.83	0.385	0.826	0.82	0.85	0.83
1000	15	0.76	0.5	0.75	0.75	0.8	0.75
3000		0.8	0.43	0.798	0.79	0.82	0.8
5000		0.81	0.39	0.812	0.81	0.82	0.81
8000		0.8	0.395	0.801	0.8	0.8	0.8
1000	30	0.77	0.455	0.76	0.76	0.77	0.77
3000		0.81	0.4	0.81	0.81	0.83	0.81
5000		0.82	0.38	0.824	0.82	0.83	0.81
8000		0.81	0.39	0.808	0.81	0.81	0.81
1000	50	0.76	0.45	0.755	0.75	0.79	0.76
3000		0.81	0.388	0.8	0.8	0.82	0.81
5000		0.83	0.354	0.827	0.83	0.83	0.83
8000		0.85	0.343	0.85	0.85	0.86	0.85

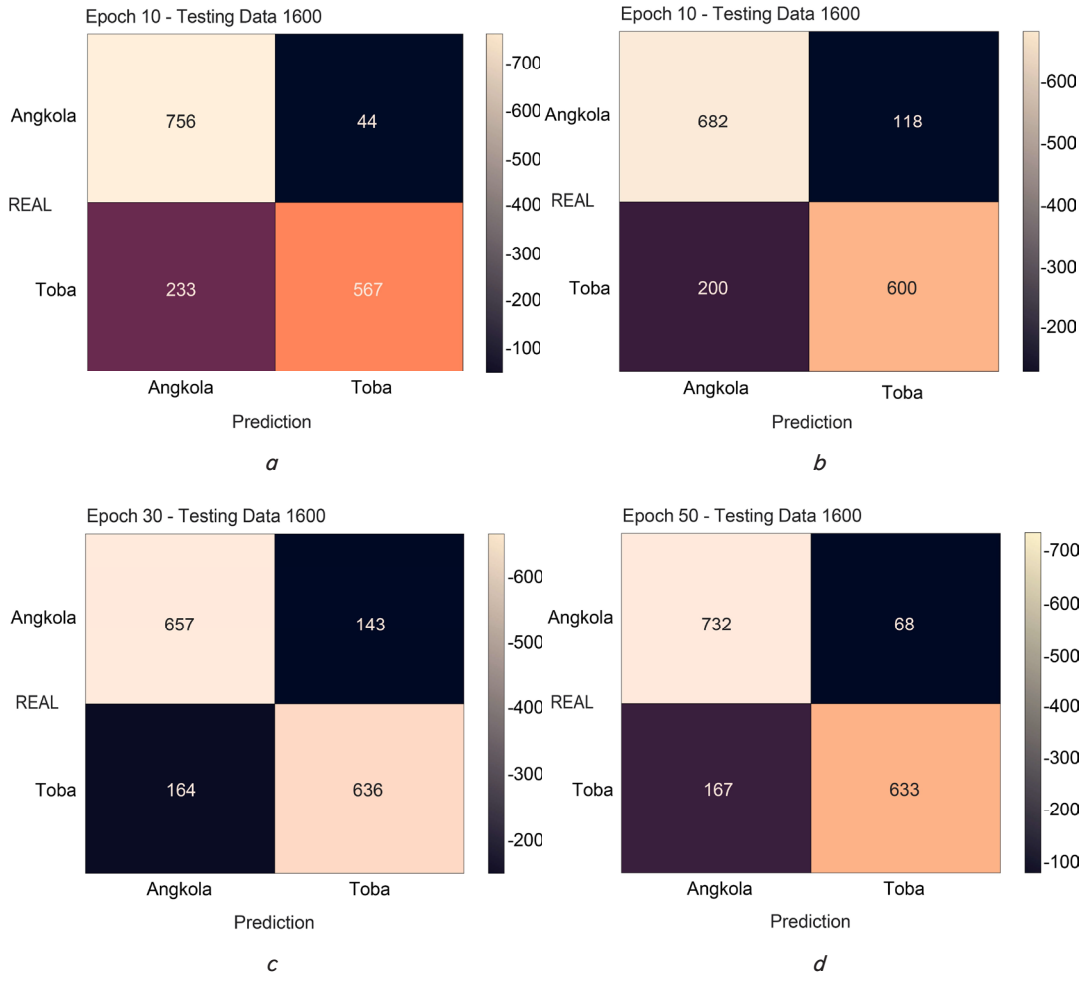

Fig. 7. Confusion matrix results with 1600 data from various experiments at different epochs: *a* — epoch 10; *b* — epoch 15; *c* — epoch 30; d — epoch 50

The table shows a clear dependence on data scale, where larger datasets generally yield higher testing accuracy even at low epoch values, such as the increase observed when moving from 1,000 to 8,000 samples under epoch 5. The best-performing configuration reported in the table is 8,000 data with epoch 50, yielding test accuracy 0.85, train loss 0.343, ROC AUC 0.85, and F1-score 0.85. A closer examination of the F1-score, precision, and recall shows that the values reported in Table 2 represent averages across the two classes, namely Batak Toba and Batak Angkola. More specific class-wise values are therefore reported in Table 3.

Table 3

Detail precision, recall, F1-score with epoch 50 Data 8000

Language	Precision	Recall	F1-score
Toba	0.81	0.92	0.86
Angkola	0.90	0.79	0.84

Table 3 shows that Batak Toba has a precision of 0.81 and Batak Angkola has a precision of 0.90, yielding an average precision of 0.86. For recall, the values are 0.92 for Batak Toba and 0.79 for Batak Angkola, resulting in an average recall of 0.85. The F1-score is likewise reported for both class perspectives: when Batak Toba is treated as the positive class (true positive), the F1-score is 0.86, whereas when Batak Angkola is treated as the positive class, the F1-score is 0.84.

The remaining errors can be interpreted in two main groups. The first group arises in spacing, where a continuous Batak string cannot be segmented cleanly because a required form is absent from the lexicon, because spelling varies across sources, or because digitized input introduces a noisy character sequence. The second group arises after spacing, where the restored sentence still contains lexical or contextual cues that overlap with the other variety and is therefore assigned to the wrong class. This behavior is consistent with the confusion matrices and with the class asymmetry observed in the best configuration, where Toba shows higher recall and Angkola shows higher precision. In the current article, the error output is available at the sentence and class level rather than as a separate audited inventory of misrecognized words, so the analysis is reported here as an interpretation of observed error behavior within the processed corpus. The progression of the training loss can be observed along with the effects of the epoch setting and the size of the training data, as shown in Fig. 8.

For the smallest dataset (800 training samples and 200 testing samples), the training loss reaches a satisfactory value only at epoch 50. In contrast, for the configuration with 2,400 training samples and 600 testing samples, the training loss converges by epoch 30, after which the decrease becomes relatively slow up to epoch 50. A similar pattern is observed for the larger datasets (5,000 and 8,000 samples), where an effective epoch range is likewise above 30.

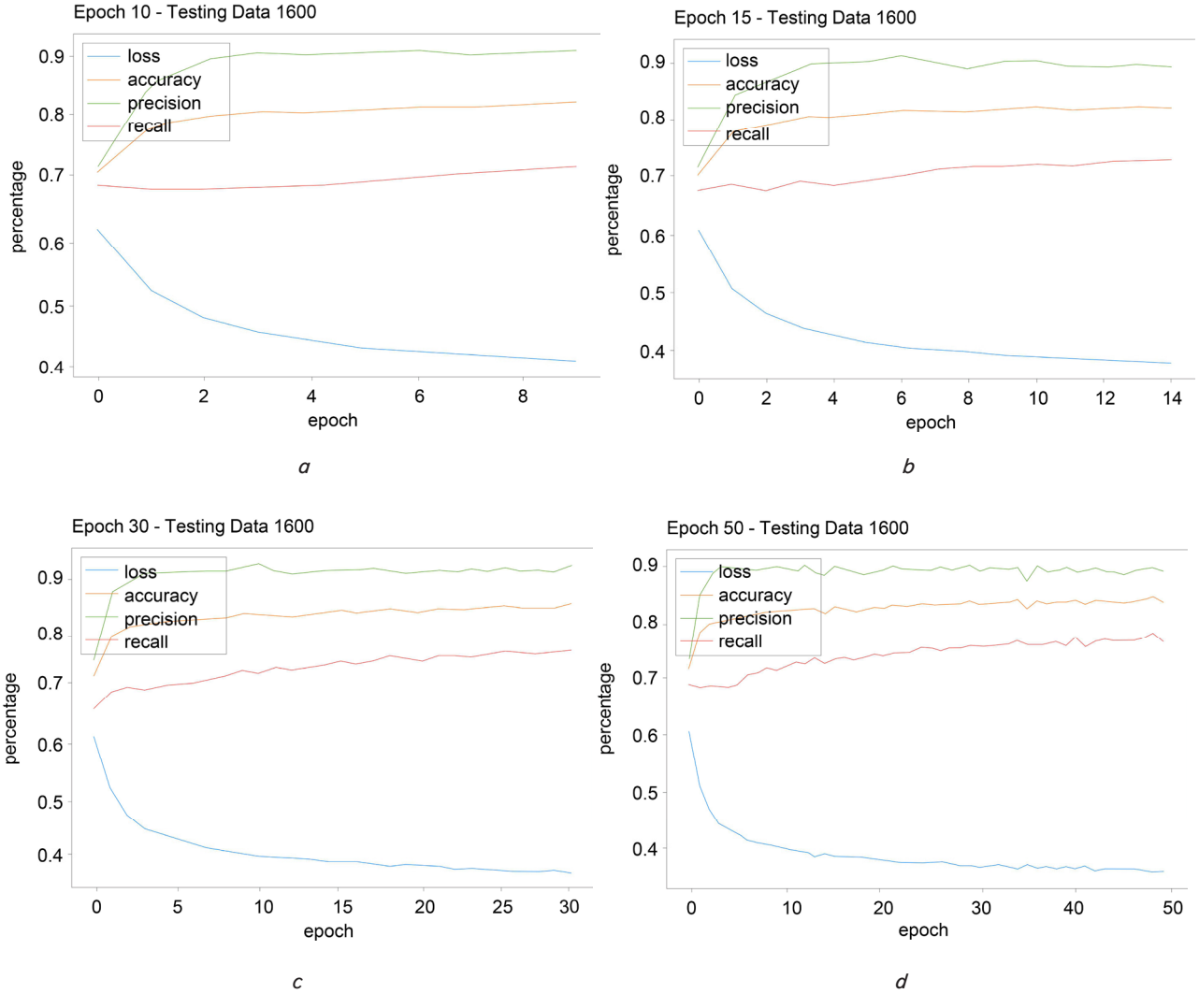

Fig. 8. Comparison results of accuracy, loss, precision, and recall with 1600 data from various experiments at different epochs: *a* — epoch 10; *b* — epoch 15; *c* — epoch 30; *d* — epoch 50

For accuracy, the trend remains upward as the dataset becomes larger and the number of epochs increases. However, the improvement remains moderate across configurations. The present study reports descriptive differences only, because no repeated-run statistical test was performed to examine significance or stability across configurations.

5. 3. Result of comparative validation against LSTM-RNN

In addition to the proposed TRIE-BERT pipeline, this study also compares the results with an LSTM-RNN baseline under the same data split and preprocessing pathway. Based on the experiments summarized in Table 4, TRIE-BERT achieves higher descriptive results than LSTM-RNN, reaching test accuracy 0.85 compared with 0.80, train loss 0.343 compared with 0.397, ROC AUC 0.85 compared with 0.803, and F1-score 0.85 compared with 0.80. These values indicate that the proposed pipeline performs better than the recurrent baseline under the reported evaluation protocol.

A likely reason for this difference is that trie-based spacing reduces token ambiguity before contextual classification begins. Once the spacing stage has restored word boundaries, the classifier receives a more regular input structure than a model that must learn sequence discrimination without an equally explicit boundary-recovery stage. The observed advantage should nevertheless be interpreted as a descriptive result, because the present study does not include repeated-run statistical testing or a formal significance analysis of the difference between the two models. In addition, the current comparison is limited to one baseline model, so broader evaluation against other modern architectures remains part of future work.

Table 4

Evaluation results for both model

Model	Test accuracy	Train loss	ROC AUC	F1-score
TRIE-BERT	0.85	0.343	0.85	0.85
LSTM-RNN	0.80	0.397	0.803	0.80

6. Discussion of TRIE-BERT spacing and classification results for Batak scriptio continua texts

Deterministic spacing explains the first stage of the reported results because the classifier operates only after continuous Batak strings have been converted into spaced text.

As shown in Table 1, trie automation restores word boundaries before classification, and the spacing audit reports 98% accuracy after correction of about 70 mismatched cases. The constructed corpus contains 19,070 lexicon entries and 8,000 sentences, which provides the lexical support required for consistent matching during spacing. This result indicates that lexicon-guided boundary recovery is workable for the evaluated Batak data, although its quality still depends directly on lexicon completeness.

The classification results remain constrained by residual overlap between Batak Toba and Batak Angkola. Fig. 7 shows persistent off-diagonal counts, which means that some restored sentences still carry cues that are not sufficiently distinctive for perfect class separation. Table 2 shows that accuracy improves as the dataset becomes larger, while Fig. 8 shows that loss reduction becomes slower after the middle training range. Table 3 further shows class asymmetry, with higher recall for Toba and higher precision for Angkola. A cautious interpretation is that some Angkola cases are recognized correctly only when their cues are more distinctive, whereas broader shared patterns allow more Toba cases to be retrieved. The present study does not include a dedicated lexical audit of each misclassified sentence, so this explanation should be read as an interpretation of observed behavior rather than as a completed linguistic account.

In comparison with existing studies, the present findings stand between pure spacing systems and contextual variety classifiers. The earlier Batak Trie Automation study also reported 98% spacing accuracy on 4,000 scriptio continua Batak sentences, which supports the usefulness of lexicon-guided segmentation in this setting [18]. The Javanese backtracking study reported 81.64% accuracy and also depended on dictionary coverage, which indicates that scriptio continua restoration remains closely tied to lexical support [7]. Korean automatic spacing research reported high character-level accuracy with pretrained transformers, while Korean reranking research showed that contextual selection can improve dictionary-based analysis [13, 14]. Arabic dialect studies further showed that hybrid contextual architectures can improve discrimination among related varieties and that stronger statistical validation can be added when repeated runs and significance tests are available [10]. An important feature of the obtained results is that deterministic spacing and contextual classification are connected in one Batak pipeline, so the restored sentence is fixed first and then reused as a stable classifier input across all experimental schemes. This feature distinguishes the present results from studies that evaluate spacing and variety classification separately.

Several limitations are inherent in the present study. The evaluated corpus reaches up to 8,000 sentences and should be regarded as a controlled experimental resource rather than a full representation of all Batak text conditions. Spacing quality remains dependent on lexicon completeness, so unseen forms, spelling variation, and digitization noise can still affect the classifier indirectly. The results are therefore adequate for the present corpus and protocol, but they should not yet be generalized to other Batak collections or to streaming input conditions.

In addition, no repeated-seed analysis or formal significance test was carried out, even though the comparison between 0.85 and 0.80 would be stronger if stability were tested explicitly. Another weakness is that residual error behavior is interpreted from confusion matrices and class metrics, but the article does not yet provide a separate audited inventory of misrecognized words or sentence-level error categories. These shortcomings can be addressed by extending the baseline set, adding repeated-run statistical checks, and collecting a dedicated error-analysis subset. Future development should also include external Batak datasets and streaming inputs, although such work will face difficulties related to corpus availability, annotation consistency, and increased experimental cost.

7. Conclusions

1. A trie-based automatic spacing module was developed for Batak scriptio continua text. The module organizes lexical entries into deterministic character-matching paths, restores spaces before classification, and fixes the restored sentence as a stable interface for subsequent modeling. Using a Batak lexical resource of 19,070 entries, the spacing stage reached 98% accuracy after correction of about 70 mismatched cases on 4,000 processed sentences. This result shows that lexicon-guided boundary recovery is feasible for the evaluated corpus, although its quality still depends on lexicon completeness and input cleanliness.

2. The TRIE-BERT pipeline was then used to classify Batak Toba and Batak Angkola after spacing restoration. The best experimental configuration was obtained with 8,000 sentences and 50 epochs, where test accuracy reached 0.85 with training loss 0.343. Class behavior remained asymmetric, with higher recall for Toba and higher precision for Angkola, which indicates that residual overlap between the two varieties persists even after spacing has been restored.

3. Compared with the LSTM-RNN baseline, the proposed pipeline produced better descriptive results under the same evaluation protocol. Test accuracy increased from 0.80 to 0.85, training loss decreased from 0.397 to 0.343, ROC AUC increased from 0.803 to 0.85, and F1-score increased from 0.80 to 0.85. Within the evaluated corpus, these results indicate that a sequential design based on spacing restoration followed by contextual classification is useful for Batak text segmentation, variety identification, and related downstream processing.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence

The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

Authors' contributions

Muhammad Anggia Muchtar: Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing; **Opim Salim Sitompul**: Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision; **Maya Silvi Lydia**: Methodology, Validation, Investigation, Visualization, Supervision; **Syahril Efendi**: Methodology, Validation, Writing – review & editing, Visualization, Supervision.

References

1. Otter, D. W., Medina, J. R., Kalita, J. K. (2021). A Survey of the Usages of Deep Learning for Natural Language Processing. IEEE Transactions on Neural Networks and Learning Systems, 32 (2), 604–624. https://doi.org/10.1109/tnnls.2020.2979670
2. Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E. et al. (2023). Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. ACM Computing Surveys, 56 (2), 1–40. https://doi.org/10.1145/3605943
3. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X. (2020). Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 63 (10), 1872–1897. https://doi.org/10.1007/s11431-020-1647-3
4. Ranathunga, S., Lee, E.-S. A., Prifti Skenduli, M., Shekhar, R., Alam, M., Kaur, R. (2023). Neural Machine Translation for Low-resource Languages: A Survey. ACM Computing Surveys, 55 (11), 1–37. https://doi.org/10.1145/3567592
5. Zampieri, M., Nakov, P., Scherrer, Y. (2020). Natural language processing for similar languages, varieties, and dialects: A survey. Natural Language Engineering, 26 (6), 595–612. https://doi.org/10.1017/s1351324920000492
6. Haq, I., Qiu, W., Guo, J., Tang, P. (2023). Correction of whitespace and word segmentation in noisy Pashto text using CRF. Speech Communication, 153, 102970. https://doi.org/10.1016/j.specom.2023.102970
7. Widiarti, A. R., Pulungan, R. (2020). A method for solving scriptio continua in Javanese manuscript transliteration. Heliyon, 6 (4), e03827. https://doi.org/10.1016/j.heliyon.2020.e03827
8. Liu, C., Peng, Y., Chng, E. S. (2025). Zero-shot Context Biasing with Trie-based Decoding using Synthetic Multi-Pronunciation. 2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 873–878. https://doi.org/10.1109/apsipaasc65261.2025.11249064
9. Alsuwaylimi, A. A. (2024). Arabic dialect identification in social media: A hybrid model with transformer models and BiLSTM. Heliyon, 10 (17), e36280. https://doi.org/10.1016/j.heliyon.2024.e36280
10. Chabane, M., Harrag, F., Shaalan, K. (2025). Advancing low-resource dialect identification: A hybrid cross-lingual model leveraging CAMeLBERT and FastText for Algerian Arabic. Expert Systems with Applications, 284, 127816. https://doi.org/10.1016/j.eswa.2025.127816
11. El Mekki, A., El Mahdaouy, A., Berrada, I., Khoumsi, A. (2022). AdaSL: An Unsupervised Domain Adaptation framework for Arabic multi-dialectal Sequence Labeling. Information Processing & Management, 59 (4), 102964. https://doi.org/10.1016/j.ipm.2022.102964
12. Saleh, H., AlMohimeed, A., Hassan, R., Ibrahim, M. M., Alsamhi, S. H., Hassan, M. R., Mostafa, S. (2025). Advancing arabic dialect detection with hybrid stacked transformer models. Frontiers in Human Neuroscience, 19. https://doi.org/10.3389/fnhum.2025.1498297
13. Hwang, T., Jung, S., Roh, Y. (2021). Korean automatic spacing using pretrained transformer encoder and analysis. ETRI Journal, 43 (6), 1049–1057. https://doi.org/10.4218/etrij.2020-0092
14. Ryu, J., Lim, S., Kwon, O., Na, S. (2024). Transformer-based reranking for improving Korean morphological analysis systems. ETRI Journal, 46 (1), 137–153. https://doi.org/10.4218/etrij.2023-0364
15. Liu, Z., Prud'hommeaux, E. (2022). Data-driven Model Generalizability in Crosslinguistic Low-resource Morphological Segmentation. Transactions of the Association for Computational Linguistics, 10, 393–413. https://doi.org/10.1162/tacl_a_00467
16. Algayres, R., Ricoul, T., Karadayi, J., Laurençon, H., Zaiem, S., Mohamed, A. et al. (2022). DP-Parse: Finding Word Boundaries from Raw Speech with an Instance Lexicon. Transactions of the Association for Computational Linguistics, 10, 1051–1065. https://doi.org/10.1162/tacl_a_00505
17. Sandeep, S., Sanjith, S., Sudarsan, B. (2025). Word segmentation of ancient Tamil text extracted from inscriptions. Npj Heritage Science, 13 (1). https://doi.org/10.1038/s40494-025-01612-2
18. Muchtar, M. A., Salim Sitompul, O., Lydia, M. S., Efendi, S. (2022). Implementation of Trie Automation Algorithm for Problem Solving Scriptio Continua. 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), 28–32. https://doi.org/10.1109/ismode53584.2022.9743133
19. Purba, M. A. (2019). Bibel Batak Toba-Indonesia. Available at: https://bibeltobaindonesia.wordpress.com/
20. Kamus bahasa Batak Toba-Indonesia. Available at: https://digilib.usu.ac.id/en/detail.php?ib=16214&i=
21. Lubis, S., Lubis, S., Mariahati, M., Naibaho, J. (1995) Kamus bahasa Indonesia - Angkola. Pusat Pembinaan dan Pengembangan Bahasa, Jakarta. Available at: https://repositori.kemendikdasmen.go.id/26811/
22. Ortakci, Y., Borhan, B. (2025). Optimizing SBERT for long text clustering: two novel approaches with empirical insights. The Journal of Supercomputing, 81 (8). https://doi.org/10.1007/s11227-025-07414-4
23. RKaban, R., Sihombing, P., Efendi, S., Lydia, M. S. (2025). Enhancing retrieval performance in social media with corpus-based query expansion using bidirectional encoder representations from transformers. Eastern-European Journal of Enterprise Technologies, 5 (2 (137)), 70–83. https://doi.org/10.15587/1729-4061.2025.340258
24. Mswahili, M. E., Hwang, J., Rajapakse, J. C., Jo, K., Jeong, Y.-S. (2025). Positional embeddings and zero-shot learning using BERT for molecular-property prediction. Journal of Cheminformatics, 17 (1). https://doi.org/10.1186/s13321-025-00959-9
25. Anggrainingsih, R., Hassan, G. M., Datta, A. (2025). Evaluating BERT-based language models for detecting misinformation. Neural Computing and Applications, 37(16), 9937–9968. https://doi.org/10.1007/s00521-025-11101-z
26. Shi, J., He, Q., Wang, Z. (2021). An LSTM-based severity evaluation method for intermittent open faults of an electrical connector under a shock test. Measurement, 173, 108653. https://doi.org/10.1016/j.measurement.2020.108653