IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5854∼5866
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5854-5866

Gramatika: a synthetic Indonesian grammatical error correction dataset generated using a rule-based approach

Michael Felix Haryono, Rico Tadjudin, Jeremy Victor Andre Napitupulu, Mirsa Salsabila, Ika Alfina, Arlisa Yuliawati
Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia

Article Info

Article history:

Received May 4, 2025
Revised Aug 13, 2026
Accepted Sep 19, 2026

Keywords:

Grammatical error correction
Indonesian
Natural language processing
Rule-based
Synthetic dataset

ABSTRACT

Grammatical error correction (GEC) is a natural language processing (NLP) task that detects and corrects errors in a text. This task includes correcting grammatical, semantic, and orthographic errors. GEC research for Indonesian, a low-resource language in NLP, has been hindered by the lack of high-quality datasets for training GEC models. This study aims to develop a new GEC dataset for Indonesian using a rule-based approach to automatically generate annotated sentences, as manual annotation is costly. The rule-based approach utilizes a thesaurus, a dictionary, a part-of-speech (POS) tagger, a morphological analyzer, and a dependency parser to generate errors that more closely resemble naturally occurring errors. Grammatically correct sentences from Indonesian Wikipedia are used as source sentences. The resulting dataset, named Gramatika, consists of 1.5 million sentences covering 17 error types and is available in both M.2 and parallel formats. To evaluate the quality of Gramatika, two established models for the GEC task, text-to-text transfer transformer (T5) and grammatical error correction: tag, not rewrite (GECToR), were fine-tuned on the dataset. The results show that multilingual text-to-text transfer transformer (mT5) achieved the highest $F_{0.5}$ score of 80.30%, outperforming the best-performing GECToR system by 3.84 percentage points, which achieved an $F_{0.5}$ score of 76.46%.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Ika Alfina
Faculty of Computer Science, Universitas Indonesia
Kampus UI Depok, Depok, West Java, Indonesia
Email: ika.alfina@cs.ui.ac.id

1. INTRODUCTION

Grammatical error correction (GEC) is the task of automatically detecting and correcting errors in text, including grammatical, orthographic, and semantic errors [1]. GEC systems can assist users in producing grammatically correct text for formal writing and can also support language learning by helping users identify and correct errors. These applications have motivated extensive research on GEC in natural language processing (NLP). Research on GEC has progressed rapidly for English compared to other languages. Among the state-of-the-art methods are text-to-text transfer transformer (T5) [2], grammatical error correction: tag, not rewrite (GECToR) [3], edit-based system combination (ESC) [4], ungrammatically syntax-based [5] and detection-correction structure [6]. T5 formulates NLP tasks as text-to-text generation problems, while GECToR treats GEC as a sequence-to-label task. Both approaches have achieved strong results on established English GEC datasets. The availability of high-quality datasets, such as Conference on Natural Language Learning 2014 (CoNLL-2014) [7], Building Educational Applications 2019 (BEA-2019) [8], NUS corpus of learner English (NUCLE) [9], and Lang-8 corpus of learner English (Lang-8) [10], has played an important

role in this progress. These datasets provide annotated corrections that can be represented in formats, such as M.2, which records the locations and types of edits and is commonly used for GEC evaluation [11].

In contrast, Indonesian GEC remains relatively under-resourced. Although several studies have addressed Indonesian spelling correction, fewer have focused on GEC. Some initial research about GEC for Indonesian used rule-based and statistical methods [12], [13], followed by others who used deep learning methods [14]–[16]. Lin *et al.* [14] developed a manually annotated dataset containing 13,709 sentences and reported an $F_0.5$ score of 55.1%. More recently, Adnan *et al.* [16] developed a dataset of 10,075 sentences from Indonesian sports news and evaluated T5 and bidirectional and auto-regressive transformers (BART) [17]. However, their dataset is not publicly available, limiting the reproducibility of their work.

Musyafa *et al.* [15] addressed the lack of training data by developing a large synthetic dataset containing 6,848,485 sentences with spelling, punctuation, and word-order errors. The source sentences were obtained from the common crawl 100 (CC100)-Indonesian dataset [18], which contains formal, informal, and slang language. However, their error-generation process relies primarily on random operations, such as adding or deleting characters and punctuation and randomly changing word order. Consequently, the generated errors do not explicitly take into account the linguistic context of the sentence.

These limitations motivate the development of a synthetic Indonesian GEC dataset that covers a broader range of error types and generates errors using linguistic context. A rule-based approach is therefore proposed that applies specialized error-generation rules for different error types and utilizes Indonesian NLP tools, including a POS tagger, morphological analyzer, and dependency parser. This approach is designed to generate more plausible errors than purely random corruption methods.

The resulting dataset is named Gramatika. It contains 1.5 million sentences covering 17 error types and is provided in both M.2 and parallel formats. Although it is smaller than the dataset of Musyafa *et al.* [15], Gramatika covers substantially more error types and generates errors while considering sentence context. To evaluate the dataset, two established GEC approaches, T5 [2] and GECToR [3], are fine-tuned on Gramatika. The proposed best model, multilingual text-to-text transfer transformer (mT5), achieves an $F_0.5$ score of 80.30% on the Gramatika test set.

The main contributions of this work are as follows: i) develop Gramatika, a synthetic Indonesian GEC dataset containing 1.5 million sentences and 17 error types; ii) propose a rule-based error-generation approach that uses Indonesian linguistic information to generate context-aware errors; iii) provide the dataset in M.2 and parallel formats for use in different GEC settings; and iv) evaluate Gramatika using established GEC approaches, namely T5 and GECToR. The remainder of the article is organized as follows. Section 2 describes the proposed method for developing Gramatika using a rule-based approach and an experimental setting. Section 3 discusses the statistics of the resulting dataset, experiment results, and discussion. Finally, section 4 presents the conclusion and future work.

2. METHOD

This section describes the methodology used to develop the Gramatika dataset. First, the 17 implemented error types are described, followed by the rule-based procedures for generating those errors. Finally, the experimental settings for evaluating the dataset are presented.

2.1. Selecting relevant error types for Indonesian

This work adopts the error types defined by Bryant *et al.* [19] for English GEC. They defined 25 types of error, as presented in Table 1. An analysis of the 25 error types was conducted by comparing them with Indonesian reference grammar [20]. The analysis revealed that 19 error types are applicable to Indonesian, while six are not: adjective form, contraction, noun number, noun possessive, verb form, and subject-verb agreement. These error types are specific to grammatical features that do not have direct equivalents in Indonesian. Among the 19 applicable types, only 17 are generated because the other and unknown categories are difficult to produce automatically.

2.2. Proposed rules for error generation

This section describes the rule-based method for generating the 17 error types in the Indonesian GEC dataset. Figure 1 shows the architecture of the proposed method. First, a valid sentence is tokenized using an Indonesian tokenizer. The errors are then generated in the error generation component using a thesaurus [21], dictionary, POS tagger, morphological analyzer, and dependency parser. Finally, in the

post-processing component, possible errors are selected, and the resulting dataset is formatted into two versions: M.2 and parallel formats. Four strategies are used to generate errors: i) substituting words with synonyms using a thesaurus for three error types; ii) changing affixes for three error types; iii) substituting words using a curated list for four error types; and iv) using specific approaches for seven error types.

Table 1. The 25 grammatical error types by Bryant *et al.* [19]

Adjective	Determiner	Noun possessive	Punctuation	Verb inflection
Adjective form	Morphology	Orthography	Spelling	Verb tense
Adverb	Noun	Particle	Subject verb agreement	Word order
Conjunction	Noun inflection	Preposition	Verb	Other
Contraction	Noun number	Pronoun	Verb form	Unknown

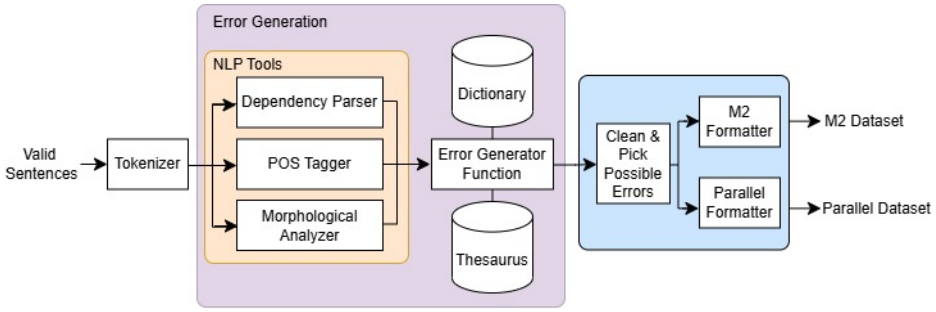

Figure 1. The architecture of the rule-based method to generate the synthetic GEC dataset

2.2.1. Substitute word with synonym using a thesaurus

This approach uses a thesaurus to replace a word with one of its synonyms, as seen in Figure 2. This approach is used to generate three error types: adjective, noun, and verb errors. A POS tagger is used to identify whether a word is a noun, verb, or adjective. If the word belongs to one of these three parts of speech, it is replaced with a synonym.

2.2.2. Change affix

This method generates errors by changing the affix of a word. This approach is used to generate three error types: morphology, noun inflection, and verb inflection errors. Figure 3 shows the proposed method of this approach. First, a POS tagger is used to identify whether a word is a noun or a verb. Its affix is then extracted using a morphological analyzer. If the required conditions are met, the affix is modified according to specific rules. Morphological errors are introduced using rules, such as changing the prefix *ber-* followed by an intransitive verb to the prefix *per-an*; and changing the prefix *me-* followed by transitive verb to the prefix *peng-an*. Prefix *me-* and *ber-* are prefixes for Indonesian verbs. The resulting word becomes a noun by changing the affix to either *per-an* or *ber-an*. Noun inflection errors are generated by changing the affixes of nouns to their alternative forms. Rules to change the affix are *per-* to *pen-* or *pe-*; *pen-* to *per-* or *pe-*; *peng-* to *per-*, or *pe-*; *pem-* to *pe-*; *peny-* to *pen-*; and *-an* to *-n*. For verb inflection, prefix *men-* followed by a lemma with the initial letter k, t, s, or p or followed by a lemma with one vowel, concatenate *men-* with a lemma. The last rule for verb inflection is to randomly change the prefix *meng-* and its derivative to one another.

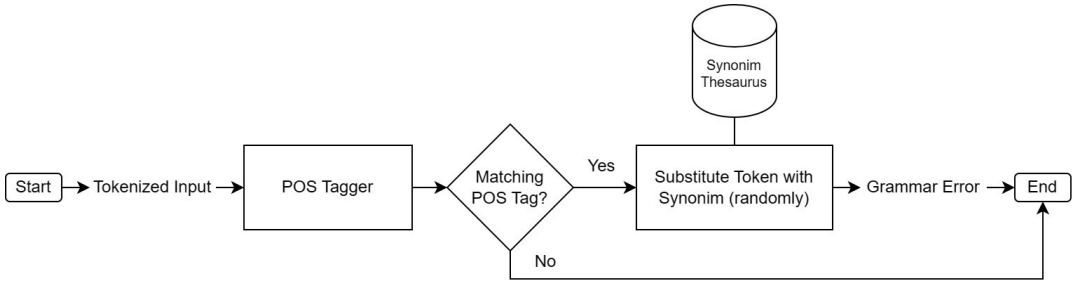

Figure 2. Rule-based procedure for generating adjective error, noun error, and verb error

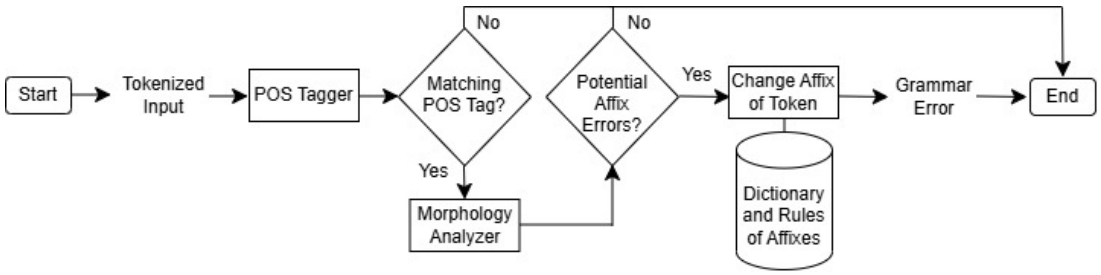

Figure 3. Rule-based procedure for generating morphology error, noun inflection error, and verb inflection error

2.2.3. Substitute word using a curated list

This method substitutes a word with selected candidates from a curated list instead of a synonym. This method is used to generate four error types: determiner, preposition, conjunction, and particle errors. Figure 4 shows the idea of this approach. To produce determiner errors, a list of classifier words in Indonesian is available, such *orang, ekor*, and *buah*. Certain classifiers can only be followed by certain nouns. For example, classifier *orang* can only be followed by a noun representing a human, and classifier *ekor* can only be followed by a noun representing an animal. Thus, changing the classifier from *orang* to *ekor* will cause errors. A curated list of prepositions and particles was created to generate preposition and particle errors. Errors were generated by replacing a preposition or particle with a similar word. For example, the preposition *di* ('at') is replaced with the preposition *pada* ('on'). For particle errors, the particle *kah*, which is usually used in interrogative sentences, is replaced with the particle *lah*, which is usually used in imperative sentences. To generate conjunction errors, two curated lists are used because there are two types of conjunctions: subordinating and coordinating conjunctions. Errors are also introduced by replacing a conjunction with a similar conjunction. For example, *dan* ('and') is replaced with *atau* ('or').

2.2.4. Generate adverb error

Figure 5 shows the proposed rules to generate adverb errors. After a word is identified as an adverb using the POS tagger, errors are generated using two rules: i) replacing the adverb with another similar adverb using a thesaurus. For example, the following change is made: *sangat* ('very') to *terlalu* ('too'); and ii) for phrasal adverbs in the form of a preposition + adjective, the phrase is corrupted by deleting the preposition. For example, *dengan baik* ('well') is replaced by *baik* ('good').

2.2.5. Generate pronoun error

This work handles two types of pronouns: personal pronouns, such as I, and interrogative pronouns, such as what and who. This approach is similar to the third approach, which uses a curated list of personal and interrogative pronouns. For example, the following changes are made: *apa* ('what') to *siapa* ('who') to create an interrogative pronoun error. Figure 6 shows this proposed rule.

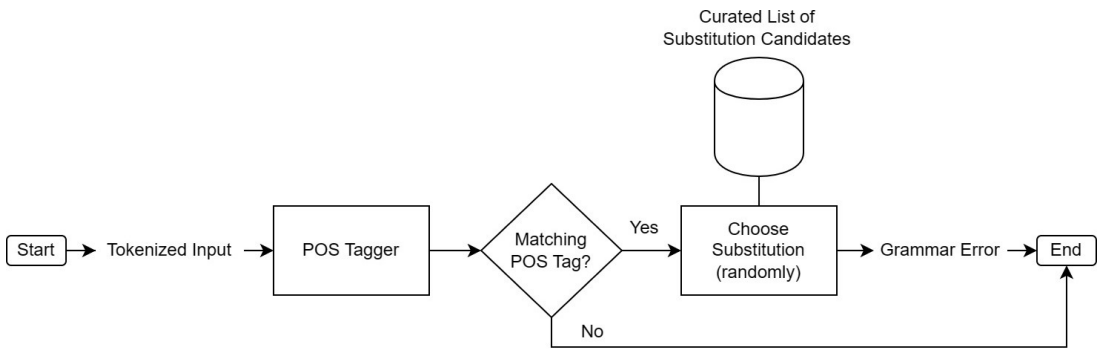

Figure 4. Rule-based procedure for generating determiner error, preposition error, conjunction error and particle error

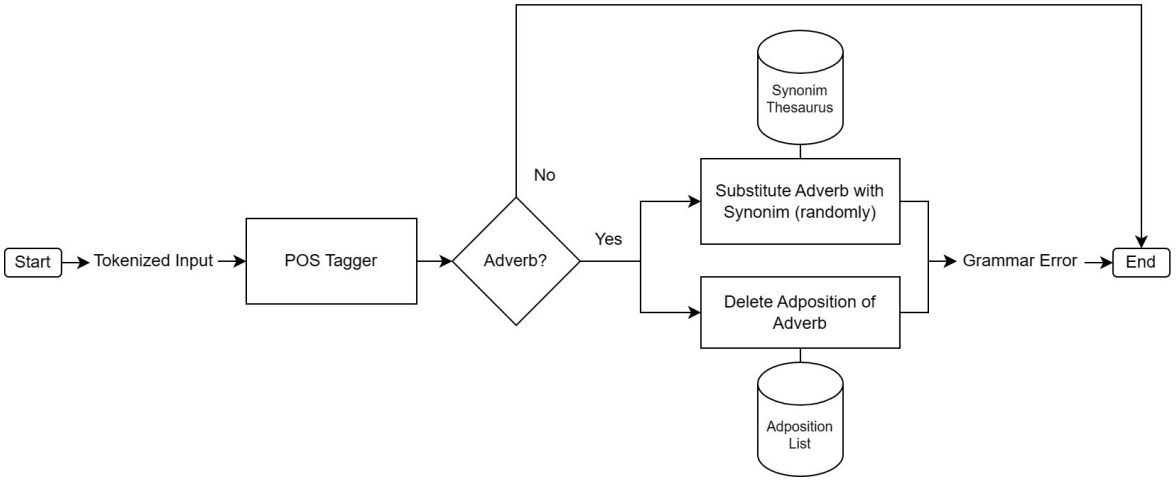

Figure 5. Rule-based procedure for generating adverb error

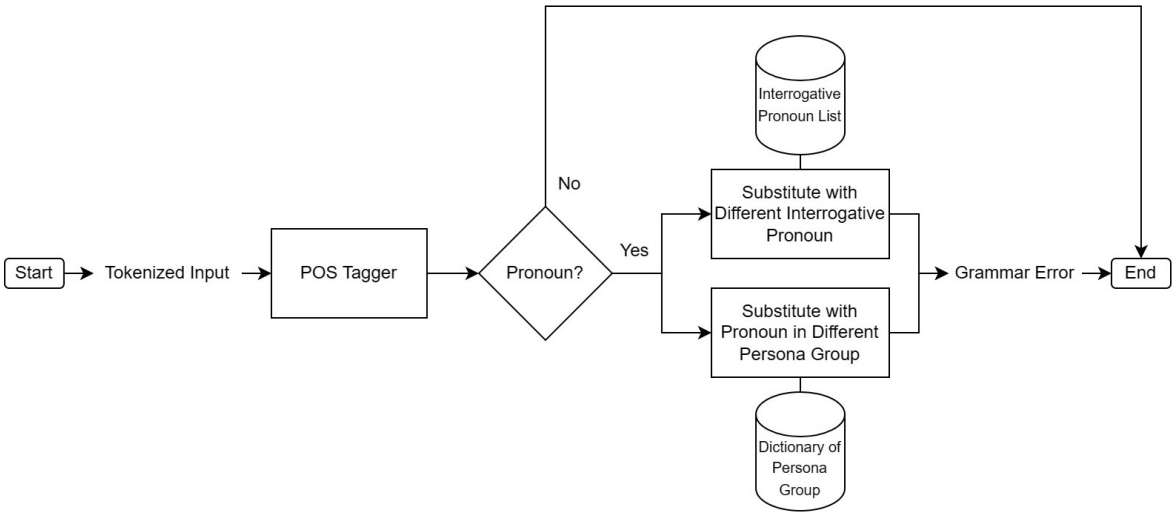

Figure 6. Rule-based procedure for generating pronoun substitution errors

2.2.6. Generate orthography error

For orthography error, three rules are introduced, as shown in Figure 7: i) generate unnecessary whitespace by adding a whitespace after prefix *di, ke, ter, ber, per* , or before the pronoun *-nya* and particle *lah, kah*. For example, the verb *dikontrakkan* ('be rented') is corrupted into *di kontrakkan*; ii) generate missing whitespace by deleting whitespace after the preposition *di, ke*. For example, the valid phrase *di rumah* ('at home') is replaced with *dirumah*; and iii) introduce incorrect capitalization. For example, the valid sentence *Dia pergi ke pasar* ('She went to the market') is replaced with *Dia pergi ke Pasar*.

2.2.7. Generate punctuation error

To generate punctuation errors, apart from using a POS tagger, a dependency parser is also required, as shown in Figure 8. A dependency parser is needed to identify complex sentences containing main and subordinate clauses. Three rules are proposed for this purpose: i) deleting or changing any punctuation mark. For example, an exclamation mark or question mark can be replaced with a period; ii) deleting commas used before conjunctions. For example, *A, B,* and *C* is corrupted into *A, B*, and *C*; and iii) delete commas used to separate subordinate clauses from their main clauses. For example, "If you are ready, we can go" is corrupted into "If you are ready we can go" by removing the comma.

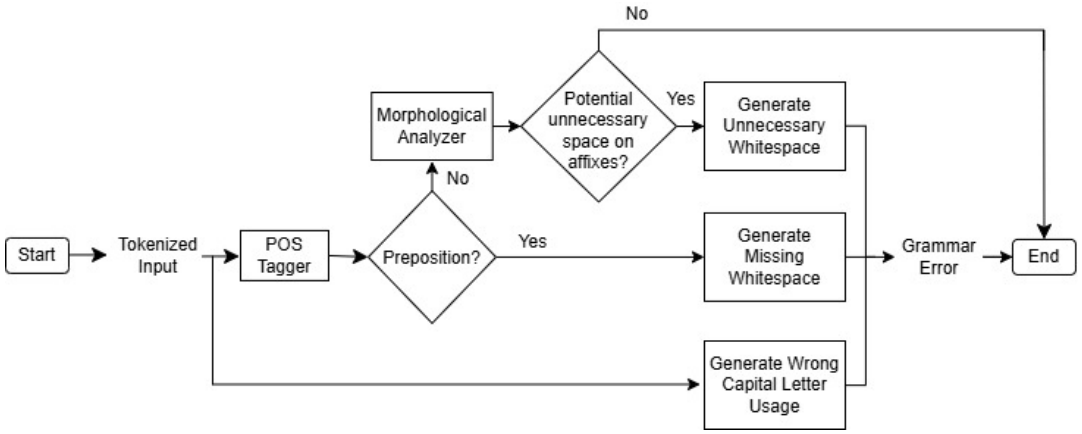

Figure 7. Rule-based procedure for generating orthography errors

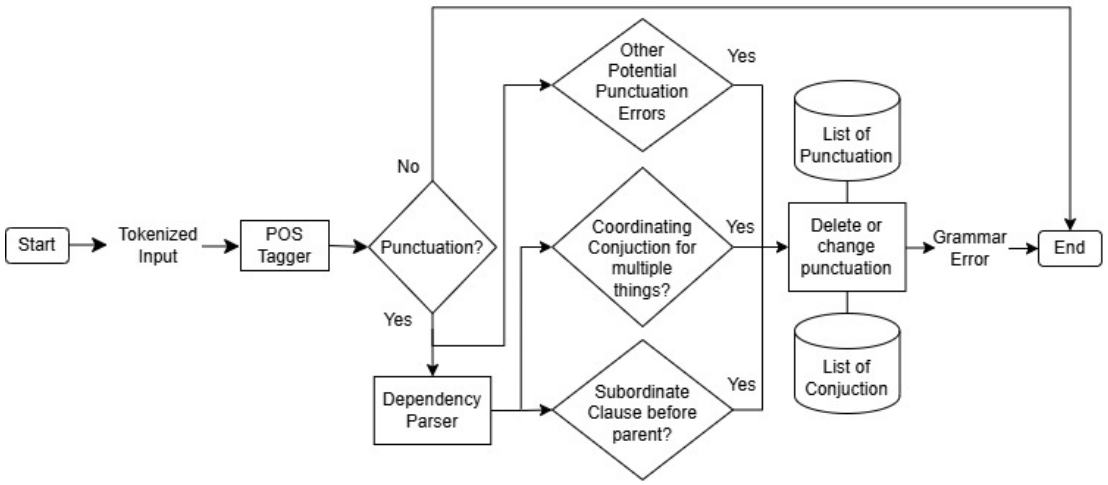

Figure 8. Rule-based procedure for generating punctuation errors

2.2.8. Generate spelling error

Four rules are proposed to generate spelling errors, as shown in Figure 9: i) adding a character, ii) deleting a character, iii) swapping two neighboring characters, and iv) deleting vowels. For adding, deleting, and swapping, the affected characters or words are picked randomly. For the last rule, all vowels will be deleted from a word with less than seven characters. This rule is inspired by the common practice of shortening words in Indonesian text messages by omitting vowels. For example, the word *jangan* ('do not') is often typed as *jgn*.

2.2.9. Generate verb tense error

Verb tense error is generated by changing the verb from passive to active form and vice versa. For this purpose, a POS tagger, a morphological analyzer, and a dependency parser were utilized. The proposed rule can be seen in Figure 10. A POS tagger is used to identify whether a word is a verb, while a morphological analyzer is needed to determine whether a verb is an active or passive form. The problem is that not all active verbs have corresponding passive verbs, but fortunately, all passive verbs have corresponding active verbs. To determine whether an active verb has the corresponding passive verbs, it is necessary to check whether the verb has a noun as its object. A dependency parser can provide this information.

2.2.10. Generate word order error

Indonesian noun phrases are generally head-initial, unlike English noun phrases, which are generally head-final. Word-order errors are therefore generated by switching the positions of a noun and a following adjective or possessive pronoun. A POS tagger identifies the relevant words, while a dependency parser locates

the corresponding noun phrases. The resulting phrases are then switched, as illustrated in Figure 11. The rules were implemented using Python, and an Indonesian NLP tool named Aksara [22] was utilized. Aksara provides tools for tokenization, POS tagging, morphological analysis, and dependency parsing.

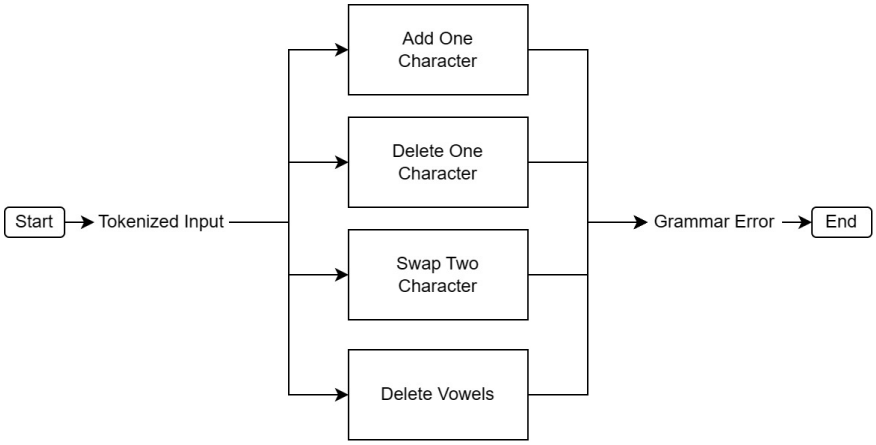

Figure 9. Rule-based procedure for generating spelling errors

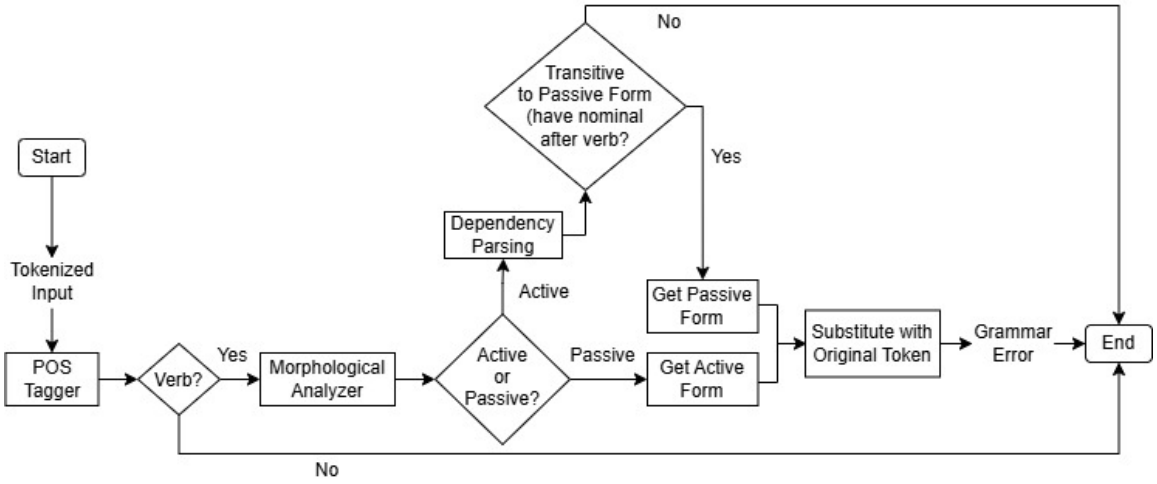

Figure 10. Rule-based procedure for generating verb tense error

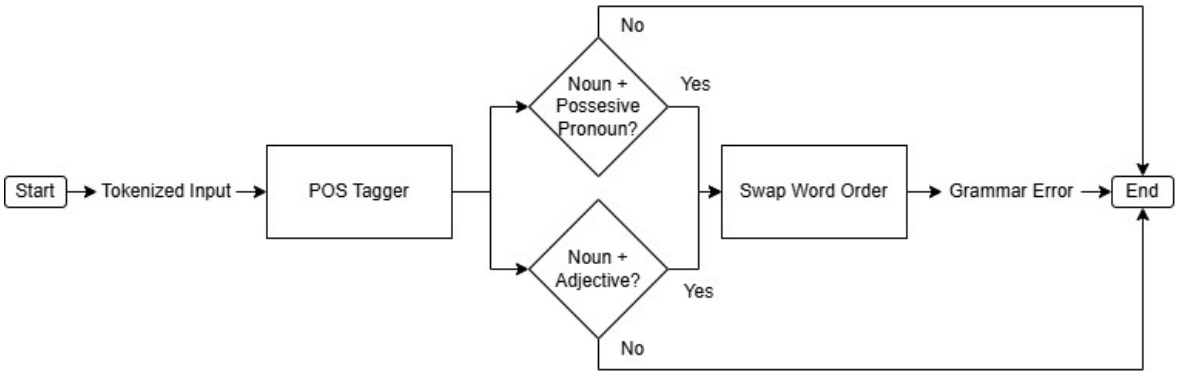

Figure 11. Rule-based procedure for generating word order error

2.3. Experiment setting

This subsection describes the experimental settings used to evaluate the quality of the Gramatika dataset. The evaluation method and datasets are presented. The evaluation metrics and tools used in the experiments are also described.

2.3.1. Method

Two established GEC approaches, T5 and GECToR, are adopted and fine-tuned on the Gramatika training and development splits. For T5, mT5 large [23] and T5 large [24] are used. mT5 is multilingual and pretrained on Indonesian, while T5 large enables a comparison between multilingual and language-specific pretraining. For GECToR, cross-lingual language model-robustly optimized bidirectional encoder representations from transformers pretraining approach (XLM-RoBERTa) large and multilingual BERT (mBERT) are used because they support multilingual input, including Indonesian.

2.3.2. Datasets

This paper evaluates the quality of the new Gramatika dataset by comparing the accuracy of GEC models trained on Gramatika. The evaluation uses two test datasets. These datasets are the Gramatika test split and the Musyafa *et al.* [15].

2.3.3. Metric and tool

To evaluate the previously mentioned dataset, MaxMatch score (M.2) [11]. The value considered from the evaluation results is the $F_{0.5}$ value obtained from the system's prediction using the gold standard (correct annotation) M.2 file as a reference. The $F_{0.5}$ score gives the precision weight two times greater than the recall weight since in the GEC task, the number of correct guesses is more critical than the ability of the GEC system to correct all errors [1]. Another tool used in this research is the bilingual evaluation understudy (BLEU) module from evaluate [25], [26] to obtain the BLEU value, which is commonly used to evaluate machine translation task, the original task of recent established GEC methods.

3. RESULTS AND DISCUSSION

This section consists of three parts. First, the statistics of Gramatika, the developed dataset, are described. Next, the experimental results for evaluating the GEC models trained using Gramatika are presented. Finally, this section concludes with a discussion.

3.1. Statistics of the resulting dataset

The Gramatika dataset has a total of 1.5 million sentences with 4,666,185 errors. Among all sentences, only 30,000 (2%) are correct sentences with no mistakes. Each sentence has a maximum of 6 errors, and there can only be 2 of the same error type in each sentence. The dataset is also divided into three splits: training, development, and test, with a proportion of 8:1:1 (comprising 1,199,705, 150,171, and 150,124 sentences, respectively). The proportion of valid sentences in each split is 2%, consisting of 24,000 sentences in the training split and 3,000 sentences in each of the development and test splits. Furthermore, the proportion of each error type is maintained consistently across all splits. For example, noun errors account for 7.5% of each split, while particle errors account for only 0.3% of each split.

3.2. Experiment results

Two variants of each method are used to compare the performance of T5 and GECToR. For T5, mT5 large and T5 large are used, while XLM-RoBERTa large and mBERT are used for GECToR. In total, four models are included in the comparison. The results are presented in Table 2. Both T5 models outperform the GECToR models, with mT5 large achieving the highest $F_0.5$ score of 80.30%. The lowest $F_0.5$ score is achieved by XLM-RoBERTa large (GECToR), with a score of 76.10

The recall values obtained by the four models for sentence correction across each error type in the Gramatika test split are further evaluated, as shown in Figure 12. The T5 models demonstrate superior performance across 13 error types, specifically morphology, verb errors, pronouns, orthography, spelling, adjectives, verb tense, word order, verb inflection, noun inflection, nouns, particles, and adverbs. In contrast, the GECToR models excel particularly in handling punctuation errors. The T5 and GECToR models perform similarly well on conjunction and determiner errors, indicating comparable effectiveness in these categories.

Furthermore, Table 3 shows the results using the Musyafa *et al.* [15] test dataset. For this purpose, BLEU is used as the evaluation metric. All our fine-tuned models show poor performance on the Musyafa *et al.* [15] test dataset, achieving only 16.89% with our best-performing model, mT5, in comparison with the results reported in Musyafa *et al.* [15] that achieved 78.13%.

Table 2. Test results on the Gramatika test dataset

Method	Model	Params	Gramatika test		
			Precision	Recall	$F_0.5$
T5	mT5 large	1.2B	78.58	88.02	80.30
	T5 large	738M	76.97	86.17	78.65
GECToR	XLM RoBERTa large	561M	78.65	67.35	76.10
	mBERT	179M	79.40	66.61	76.46

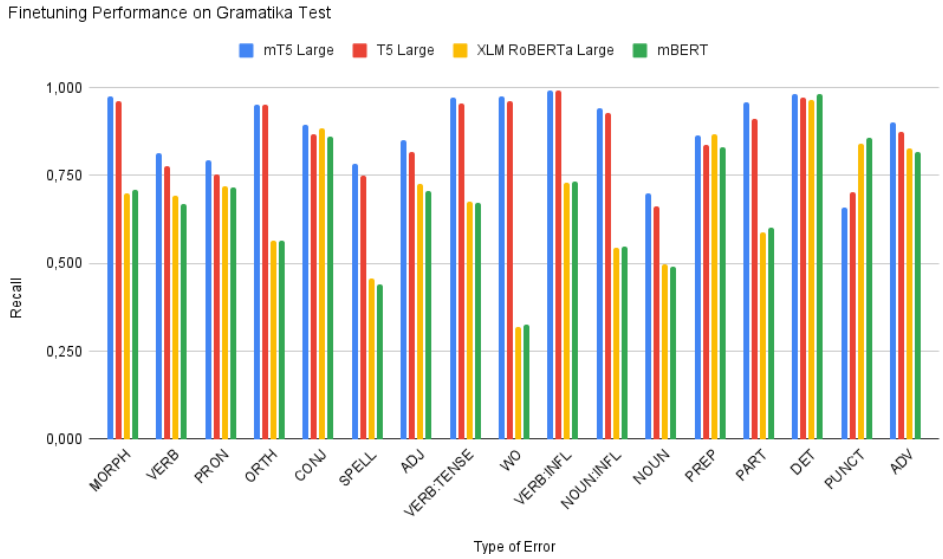

Figure 12. Recalls for each error type using the Gramatika test dataset

Table 3. Test results on the Musyafa *et al.* [15] (MUS) test dataset

Method	Model	Train and dev	MUS test BLEU
T5	mT5 large	Gramatika	16.8
	T5 Large	Gramatika	14.45
GECToR	XLM RoBERTa large	Gramatika	14.36
	mBERT	Gramatika	16.15
Musyafa *et al.* [15]	Transformer	Musyafa	78.13

3.3. Discussion

The proposed models performed well on the Gramatika test set but poorly on the Musyafa *et al.* [15] test set. Three possible factors are identified that may explain this difference.

i) Source characteristics: Gramatika uses formal Wikipedia text, whereas the Musyafa *et al.* [15] dataset is derived from CC100-Indonesian and contains formal, informal, and slang language. This difference in source characteristics may reduce the generalization of models trained on Gramatika.

ii) Error density: Gramatika limits each sentence to a maximum of six errors, whereas the Musyafa dataset may contain substantially more errors. For example, an 18-token sentence containing 12 errors was identified in their test set, as shown in Figure 13.

iii) Error plausibility: Gramatika modifies dictionary words while preserving names and abbreviations, whereas the Musyafa *et al.* [15] method can corrupt non-dictionary words and apply multiple corruptions to the same word. In the proposed inspection, some such examples were difficult for native Indonesian speakers to correct manually. This suggests that Gramatika produces more plausible errors, although multiple corruptions could be considered in future work. Examples are shown at Figure 14.

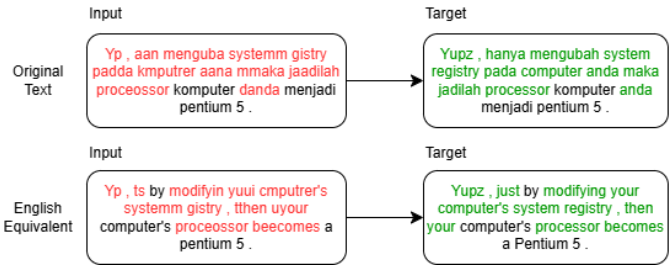

Figure 13. A sentence in the Musyafa *et al.* [15] test dataset with a high number of errors

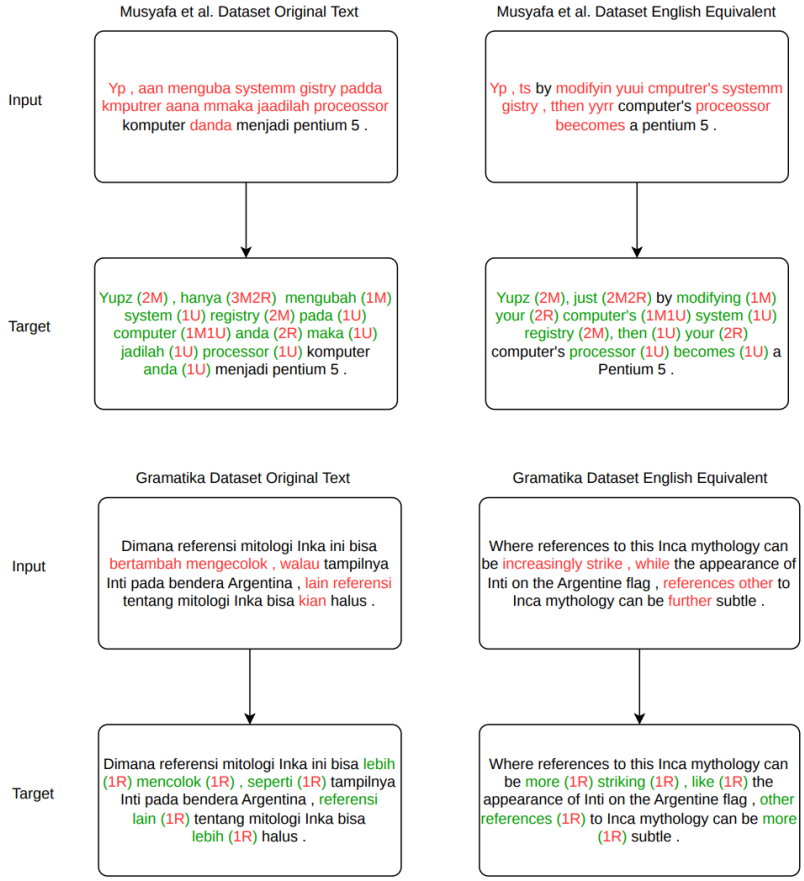

Figure 14. Example of sentence correction breakdown

4. CONCLUSIONS AND FUTURE WORK

Gramatika was developed as a synthetic Indonesian GEC dataset containing 1.5 million sentences, 4,666,185 errors, and 17 error types. A rule-based error-generation method was proposed using Indonesian NLP tools, including a tokenizer, POS tagger, morphological analyzer, and dependency parser, to generate errors according to sentence context. The dataset is provided in M.2 and parallel formats. Gramatika was evaluated by fine-tuning T5 and GECToR models. The best-performing model, mT5 large, achieved an $F_0.5$ score of 80.30% on the Gramatika test set, while the GECToR model using mBERT achieved 76.46%. The proposed analysis also identified several limitations. The current method does not generate multiple errors within the same word, limits each sentence to a maximum of six errors, and cannot generate some error types such as word repetition. The synonym-based generation method can also produce corrections that differ from the target while remaining semantically acceptable. Future work will investigate these limitations and improve the error-generation rules.

ACKNOWLEDGMENTS

The authors thank Muhammad Reza Qorib, Ph.D. for his valuable insights and advice on GEC research.

FUNDING INFORMATION

This work was supported by the Faculty of Computer Science, Universitas Indonesia, under Hibah Riset Internal 2023 (Grant No. NKB-007/ UN2.F11.D/HKP.05.00/2023).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Michael Felix Haryono	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Rico Tadjudin	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Jeremy Victor Andre Napitupulu	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Mirsa Salsabila	✓	✓	✓	✓	✓	✓		✓	✓					
Ika Alfina	✓	✓			✓	✓	✓		✓	✓		✓	✓	✓
Arlisa Yuliawati	✓	✓			✓	✓			✓	✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

The author declares that there are no known conflicts of interest associated with this publication. There are no financial or personal relationships that could inappropriately influence or bias the content of this work.

INFORMED CONSENT

Not applicable. This study did not involve human participants, human data, or any personally identifiable information. All data used were either publicly available, fully anonymized, or derived from non-human sources, and therefore no informed consent was required from individuals.

ETHICAL APPROVAL

Not applicable. This research did not involve human subjects, human biological materials, or experimental procedures on animals. The work was conducted solely on computational models, publicly available datasets, or non-sensitive data that did not require intervention with living organisms. Therefore, ethical approval from an institutional review board or animal ethics committee was not necessary for this study.

DATA AVAILABILITY

Gramatika, the dataset we created, is available at https://github.com/ir-nlp-csui/gramatika after publication.

REFERENCES

[1] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical error correction: a survey of the state of the art," *Computational Linguistics*, vol. 49, no. 3, pp. 643–701, 2023, doi: 10.1162/coli_a_00478.

[2] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, "A simple recipe for multilingual grammatical error correction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, vol. 2, pp. 702–707, doi: 10.18653/v1/2021.acl-short.89.

[3] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, "GECToR – grammatical error correction: tag, not rewrite," in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 163–170, doi: 10.18653/v1/2020.bea-1.16.

[4] M. R. Qorib, S.-H. Na, and H. T. Ng,"Frustratingly easy system combination for grammatical error correction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 1964-1974, doi: 10.18653/v1/2022.naacl-main.143.

[5] C. Tang, F. Qu, and Y. Wu,"Ungrammatical-syntax-based in-context example selection for grammatical error correction," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 1758-1770, doi: 10.18653/v1/2024.naacl-long.99.

[6] W. Li and H. Wang,"Detection-correction structure via general language model for grammatical error correction," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 1748-1763, doi: 10.18653/v1/2024.acl-long.96.

[7] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The CoNLL-2014 shared task on grammatical error correction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014, pp. 1–14, doi: 10.3115/v1/W14-1701.

[8] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 52–75, doi: 10.18653/v1/W19-4406.

[9] D. Dahlmeier, H. T. Ng, and S. M. Wu, "Building a large annotated corpus of learner English: the NUS corpus of learner English," in *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications*, BEA 2013, 2013, pp. 22–31.

[10] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning SNS for automated Japanese error correction of second language learners," in *IJCNLP 2011 - Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011, pp. 147–155.

[11] D. Dahlmeier and H. T. Ng, "Better evaluation for grammatical error correction," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012, pp. 568–572.

[12] A. Fahda and A. Purwarianti, "A statistical and rule-based spelling and grammar checker for Indonesian text," in *2017 International Conference on Data and Software Engineering (ICoDSE)*, Nov. 2017, pp. 1–6, doi: 10.1109/ICODSE.2017.8285846.

[13] F. Rahutomo, A. S. Mulyo, and P. Y. Saputra, "Automatic grammar checking system for Indonesian," in *2018 International Conference on Applied Science and Technology (iCAST)*, Oct. 2018, pp. 308–313, doi: 10.1109/iCAST1.2018.8751591.

[14] N. Lin, B. Chen, X. Lin, K. Wattanachote, and S. Jiang, "A framework for Indonesian grammar error correction," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 4, 2021, doi: 10.1145/3440993.

[15] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, "Automatic correction of Indonesian grammatical errors based on transformer," *Applied Sciences*, vol. 12, no. 20, 2022, doi: 10.3390/app122010380.

[16] M. Adnan *et al.*, "Grammar error correction in Indonesian sports news: comparing the performance of pre-trained T5 and BART models," in *2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C)*, Dec. 2024, pp. 142–147, doi: 10.1109/BTS-I2C63534.2024.10941994.

[17] M. Lewis *et al.*, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880, doi: 10.18653/v1/2020.acl-main.703.

[18] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.

[19] C. Bryant, M. Felice, and T. Briscoe, "Automatic annotation and evaluation of error types for grammatical error correction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, vol. 1, pp. 793–805, doi: 10.18653/v1/P17-1074.

[20] H. Alwi, S. Dardjowidjojo, H. Lapoliwa, and A. M. Moeliono, *Tata bahasa baku bahasa Indonesia*. Jakarta, Indonesia: Perum Balai Pustaka, 1998.

[21] Agency for Language Development and Cultivation, Ministry of Primary and Secondary Education, Republic of Indonesia, *Tesaurus bahasa Indonesia pusat bahasa*. Jakarta, Indonesia: Ministry of Primary and Secondary Education, Republic of Indonesia, 2008.

[22] M. Y. Hanifmuti and I. Alfina, "Aksara: an Indonesian morphological analyzer that conforms to the UD v2 annotation guidelines," in *2020 International Conference on Asian Language Processing (IALP)*, Dec. 2020, pp. 86–91, doi: 10.1109/IALP51396.2020.9310490.

[23] L. Xue *et al.*, "mT5: a massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498, doi: 10.18653/v1/2021.naacl-main.41.

[24] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 5485–5551, 2020.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 2002, pp. 311–318, doi: 10.3115/1073083.1073135.

[26] C.-Y. Lin and F. J. Och, "ORANGE: a method for evaluating automatic evaluation metrics for machine translation," in *Proceedings of the 20th international conference on Computational Linguistics (COLING)*, 2004, pp. 501–507, doi: 10.3115/1220355.1220427.

BIOGRAPHIES OF AUTHORS



Michael Felix Haryono received his B.Sc. degree in Computer Science from Universitas Indonesia, Indonesia, in 2023. He developed a synthetic GEC dataset for Indonesian for his undergraduate thesis. He is keenly interested in further advancing his research in NLP and machine learning. He can be contacted at email: michael.felix@ui.ac.id.



Rico Tadjudin received his B.Sc. in Computer Science from Universitas Indonesia in 2023. His studies spanned diverse fields within computer science, with a keen interest in machine learning topics such as NLP and image processing. His undergraduate thesis explored solutions to the challenge of limited training resources for GEC for Indonesian. He can be contacted at email: rico.tadjudin@ui.ac.id.



Jeremy Victor Andre Napitupulu received the B.Sc. degree in Computer Science from Universitas Indonesia, Indonesia, in 2023. His academic focus centered on NLP, with his undergraduate thesis exploring the GEC methods for Indonesian. Beyond his academic studies, he is keenly interested in social informatics, aiming to explore the intersection of technology and societal impact. He can be contacted at email: jeremy.victor@ui.ac.id.



Mirsa Salsabila holds a bachelor's degree in Computer Science from Universitas Indonesia in 2023. During her studies, she focused on data science and web development, with a particular interest in NLP. Her undergraduate thesis explored GEC methods. She works as a software engineer, primarily specializing in front-end web development. She can be contacted at email: mirsa.salsabila@ui.ac.id.



Ika Alfina received the B.S., M.Sc., and Dr. degrees in Computer Science from Universitas Indonesia, Depok, Indonesia, in 2000, 2007, and 2021, respectively. She has been a faculty member of the Faculty of Computer Science, Universitas Indonesia, since 2007. She has authored/co-authored more than 30 publications in peer-reviewed journals and conference proceedings in the area of NLP. Her research interests include morphological analysis, syntactic parsing, and GEC. She can be contacted at email: ika.alfina@cs.ui.ac.id.



Arlisa Yuliawati received a bachelor's degree (S.Kom) in Informatics Engineering from Institut Teknologi Sepuluh Nopember Surabaya in 2011 and a master's degree (M.Kom) in Computer Science from Universitas Indonesia in 2013. She is currently pursuing her doctoral degree in Computer Science at Universitas Indonesia. She has been a faculty member of the Faculty of Computer Science, Universitas Indonesia, since 2013. Her research interests include machine translation, text summarization, and NLP for low-resource language. She can be contacted at email: arlisa@cs.ui.ac.id.