IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 July 2023, accepted 16 August 2023, date of publication 23 August 2023, date of current version 5 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307712

RESEARCH ARTICLE

SPECIL: Spell Error Corpus for the Indonesian Language

YANFI YANFI[1], (Graduate Student Member, IEEE), REINA SETIAWAN[1,2], HARYONO SOEPARNO[1], AND WIDODO BUDIHARTO[2]

[1]Computer Science Department, BINUS Graduate Program—Doctor of Computer Science Program, Bina Nusantara University, Jakarta 11480, Indonesia
[2]Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding author: Yanfi Yanfi (eufrasia.yan.fi@binus.ac.id)

This work was supported in part by the U.S. Department of Commerce under Grant BS123456, and in part by Bina Nusantara University.

ABSTRACT In this study, we present the first spell error corpus for the Indonesian Language (SPECIL). This corpus provides a comprehensive resource for researchers and practitioners to detect and correct spelling errors in *Bahasa Indonesia* (Indonesian). It should be emphasized that currently, there is no recognized corpus for identifying spelling mistakes in the Indonesian language that has been officially released or made accessible. This study also provides a systematic literature review to identify resources and methodologies for building a corpus for spelling error detection and correction in Indonesia. A corpus was created using a combination of manual and automatic methods. The results of this study are a review of publications relating to corpora and spelling, the novel algorithm of six types of spelling errors, and the production of a corpus comprising over 180,000 tokens in 21,500 sentences, including non-word, real-word, and punctuation errors. Using the developed corpus, various Natural Language Processing (NLP) models, including spell checkers and language models, can be trained and tested to enhance their accuracy and effectiveness in identifying and rectifying errors in Indonesian texts.Moreover, the corpus can be used to develop and evaluate new algorithms and techniques for spelling error detection and correction in Indonesia. The SPECIL corpus is publicly available and accessible.It is expected that SPECIL will inspire further research in this area and facilitate the development of more accurate and effective spelling error detection and correction tools in Indonesian language.

INDEX TERMS Corpus, Indonesian language, natural language processing, spell.

I. INTRODUCTION

A corpus is a structured collection of text or spoken language data selected and gathered for linguistic analysis. It can include a variety of written or spoken sources, such as books, newspapers, transcribed conversations, or online content. Corpus is widely used in computational linguistics and natural language processing to develop and test algorithms [1], analyze language patterns and structures [2] and explore language use in different contexts [3].

A corpus in the Indonesian language can be utilized for various purposes, such as studying grammar, syntax, and discourse [4], developing machine translation systems, text-to-speech systems, or other language technologies. They can be constructed manually or automatically and annotated with linguistic data, such as part-of-speech tags, semantic labels, or syntactic structures, to enable further in-depth studies.

Utilizing publicly accessible corpora, such as the Indonesian new corpus and Indonesian Wikipedia corpus, research on grammatical errors in the Indonesian language has been conducted [5]. However, it is crucial to note that no officially published or openly available corpus identifies spelling errors in Indonesia.

Developing a particular corpus for detecting errors in the Indonesian language is an essential step toward enhancing the quality of error detection tools, such as grammar checkers or proofreading software. Having a specified corpus improves the accuracy and relevance of mistake detection, allowing developers to improve and develop better error detection tools continuously.

The associate editor coordinating the review of this manuscript and approving it for publication was Alicia Fornés.

Sentences are the basic units of a language. They are composed of words arranged in a specific order to convey complete thoughts and ideas. A sentence can be simple or complicated and consists of one or more clauses. Sentences are essential for effective communication because they allow speakers and writers to express their thoughts and ideas clearly and in an organized manner. Studying syntax and semantics is vital for understanding the structure and meaning of sentences.

This study consisted of six parts. Section I introduces the analysis. Section II provides general information on syntax and semantics in Indonesia. Section III contains the research methodology, such as literature review, corpus creation, and the method of summarizing corpus creation. Section IV provides a review of the literature. Section V presents the results of corpus creation, followed by a summary. Finally, conclusions are provided in Section VI.

II. SENTENCE STRUCTURE IN THE INDONESIAN LANGUAGE

Bahasa Indonesia(Indonesian) is a fascinating language spoken by millions across the Indonesian archipelago and the world.

Sentences are used to communicate ideas or thoughts to others in Indonesia. Primary and derivative sentences were presented in [6]. The characteristics of basic Indonesian sentences are as follows:

1. It has only one clause.
2. Although some sentences include an object, information or description, and complement, they primarily consist of a subject and a verb.

"*Saya pergi.*" (I go.) can be extended to "*Saya pergi ke took.*" (I go to the store.).

3. The arrangement is not inversion.

"*Ke toko, saya pergi.*" (To the store I am going.) is not an introductory sentence but a derivative sentence.

4. It has never experienced a substitution process, for example, "*Saya mencari Dino*." (I looked for Dino.) is a basic sentence, not replaced by "*Saya mencarinya*." (I looked for him.).
5. It has a transitive verb that the existence of an object will help understand the meaning.

"*Saya makan nasi*" (I eat rice) as active sentence. An active sentence can be a derivative sentence when it is changed to a passive sentence, e.g., "*Nasi dimakan saya*" (Rice is eaten by me).

6. It is no nominalization. "*Ibu pergi tadi pagi*" (Mother left this morning) is a basic sentence, but "*Perginya tadi pagi*" (Left this morning) is a derivative sentence.

Based on these characteristics, it is necessary to understand Indonesian sentences with sufficient knowledge of syntax and semantics. The syntax is the study of rules governing the structure of sentences and phrases in a language. To determine whether a sentence meets the requirements of grammatical rules, it is necessary to pay attention to the completeness of its elements, such as *subyek* (subject), *predikat* (verb), *obyek* (object), *pelengkap* (compliment), *keterangan* (adverb) [7].

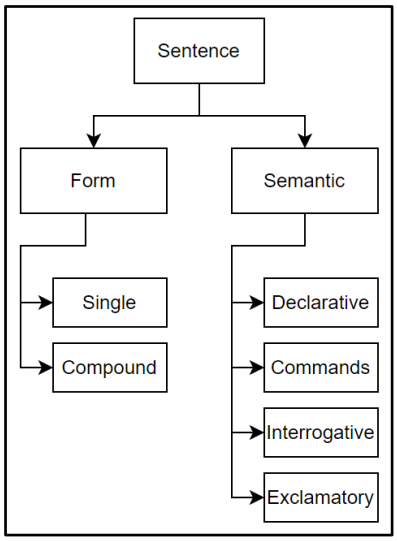

FIGURE 1. Division of sentences based on form and meaning in Indonesian language.

The pattern in the simple Indonesian language [8] is presented in Table 1.

Sentences can be divided according to their form of communication and semantic (meaning) or value [9], as shown in Figure 1. Based on this form, sentences were divided into single and compound sentences.

1. Single sentence

A single sentence consists of one clause. This sentence consists of a subject and verb with or without an object, compliment, or adverb and has the potential to become a sentence.

2. Compound sentence

A sentence form can be determined as a compound if it can be split into two or more clauses without changing the information or message. For example, *"Amy membaca buku dan Doni menyanyikan sebuah lagu."* (Amy reads the book, and Doni sings a song.) can be divided into two clauses: *"Amy membaca buku."* (Amy reads the book.) and *"Doni menyanyikan sebuah lagu."*(Doni sings a song.). It is because their meanings are identical.

According to their meaning, sentences can be divided into news or declarative, command or imperative, interrogative, and exclamatory or emphatic.

1. Declarative sentence

A declarative sentence is a sentence whose content conveys a statement addressed to another person so that the other person is expected to respond through a response that can be reflected in a glance or expression and is sometimes accompanied by a nod or a yes.

"Siska akan melanjutkan kuliah." (Siska will continue her studies.).

"*Kamu harus berhati-hati setibanya di Jakarta.*" (You have to be careful when you arrive at Jakarta.).

The declarative sentence must end with a full stop or a dot.

2. Command or imperative sentence

An imperative sentence asks the listener or reader to take action. This imperative sentence can be a command, appeal, or prohibition.

TABLE 1. The pattern of simple Indonesian language.

No	Type	Function				
		Subyek **Subject (S)**	***Predikat*** **Verb (V)**	***Obyek*** **Object (O)**	***Pelengkap*** **Complement (C)**	***Keterangan*** **Adverb (A)**
1	S-V	*Saya* (I)	*pergi* (go)			
2	S-V-O	*Saya* (I)	*mencari* (look for)	*Doni* (Doni)		
3	S-V-C	*Saya* (I)	*menjadi* (become)		*manajer* (manager)	
4	S-V-A	*Saya* (I)	*pergi* (go)			*ke toko* (to the store)
5	S-V-O-C	*Kamu* (You)	*mengirimi* (send)	*saya* (me)	*sebuah topi* (a hat)	
6	S-V-O-A	*Kamu* (You)	*membeli* (buy)	sebuah topi *(a hat)*		di pasar (in the market)

"*Jagalah kebersihan!*" (Keep it clean!)

"*Dilarang merokok!*" (No smoking!)

The imperative sentence must end with an exclamation mark (!).

3. Interrogative sentence

Interrogative sentences are those in which verbal answers are to be expected. The answer can be in the form of yes or no or in the form of a long explanation.

"*Apa Ahmad pergi ke pasar?*" (Did Ahmad go to market?)

"*Mengapa kamu tidak belajar?*" (Why do you not study?)

Indonesian also uses certain particles.

a. The particle "*kah*" is used to indicate a question, e.g., "*Anda sudah makankah?*" (Have you eaten it?) or "*Saya datang terlambatkah?*" (Did I come late?).
b. The particle "ya" is used as a tag question, e.g., "*Besok kita pergi ke took, ya?*" (Let's go to the store tomorrow, okay?) or "*Kamu sudah makan, ya?*" (You have eaten, have not you?).

The interrogative sentence must end with a question mark (?).

4. Exclamatory sentence

An exclamatory sentence expresses emotions such as admiration, surprise, amazement, astonishment, anger, sadness, exasperation, disappointment, and dislike. This sentence is composed of a clause with exclamation words such as "*wah*" (wow), "*nah*" (well), "*aduh"*(ouch), ah, and "*alangkah"*(how).

"*Wah, cantik sekali!*" (Wow, how beautiful!)

Exclamation sentences must end with an exclamation mark (!).

III. METHODOLOGY

Figure 2 shows the block diagram of the study. The flow begins with a systematic literature review, followed by the construction of the corpus, and concludes with a summary of the corpus creation.

A. SYSTEMATIC LITERATURE REVIEW

This study analyzes the corpus in Natural Language Processing for spells from the literature review. The papers used in this study were collected from the bibliographic database, Scopus.com. Therefore, this research takes article sources with no time scale using the following keywords: "Corp*" AND "natural language processing" AND "spell"

Next, we created inclusion and exclusion criteria to obtain the articles. The inclusion criteria were as follows:

1. The article type is a conference paper or journal.
2. The article is in English or Indonesian language.
3. The article is in the field of computer science.

The study is limited to computer science because Natural Language Processing (NLP) is in the field of computer science. In addition, we conducted a literature review to provide background information to prove that this study has a gap.

The exclusion criteria are:

1. The article type is not a conference paper or journal (lecturer notes, literature review, theses, or dissertations).
2. The article is not in English or Indonesian language.
3. The article is not in the field of Computer Science.

The following are the query results from Scopus.com (https://www.scopus.com/search/) after adding the inclusion and exclusion criteria.

TITLE-ABS-KEY (corp* AND natural AND language AND processing AND spell) AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOCTYPE, "ar")) TITLE-ABS-KEY (corp* AND natural AND language AND processing AND spell) AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOCTYPE "ar")) AND (LIMIT-TO (LANGUAGE, "English")).

The analysis was performed using the VOSviewer (https://www.vosviewer.com/), including the trend of publications and citations as well as the languages, mapping research corpus by year, and mapping spell checker.

B. CORPUS CREATION

Non-word and real-word errors are the most common causes of spelling errors in Indonesians. Real-word error means that the word is correct but used in the wrong

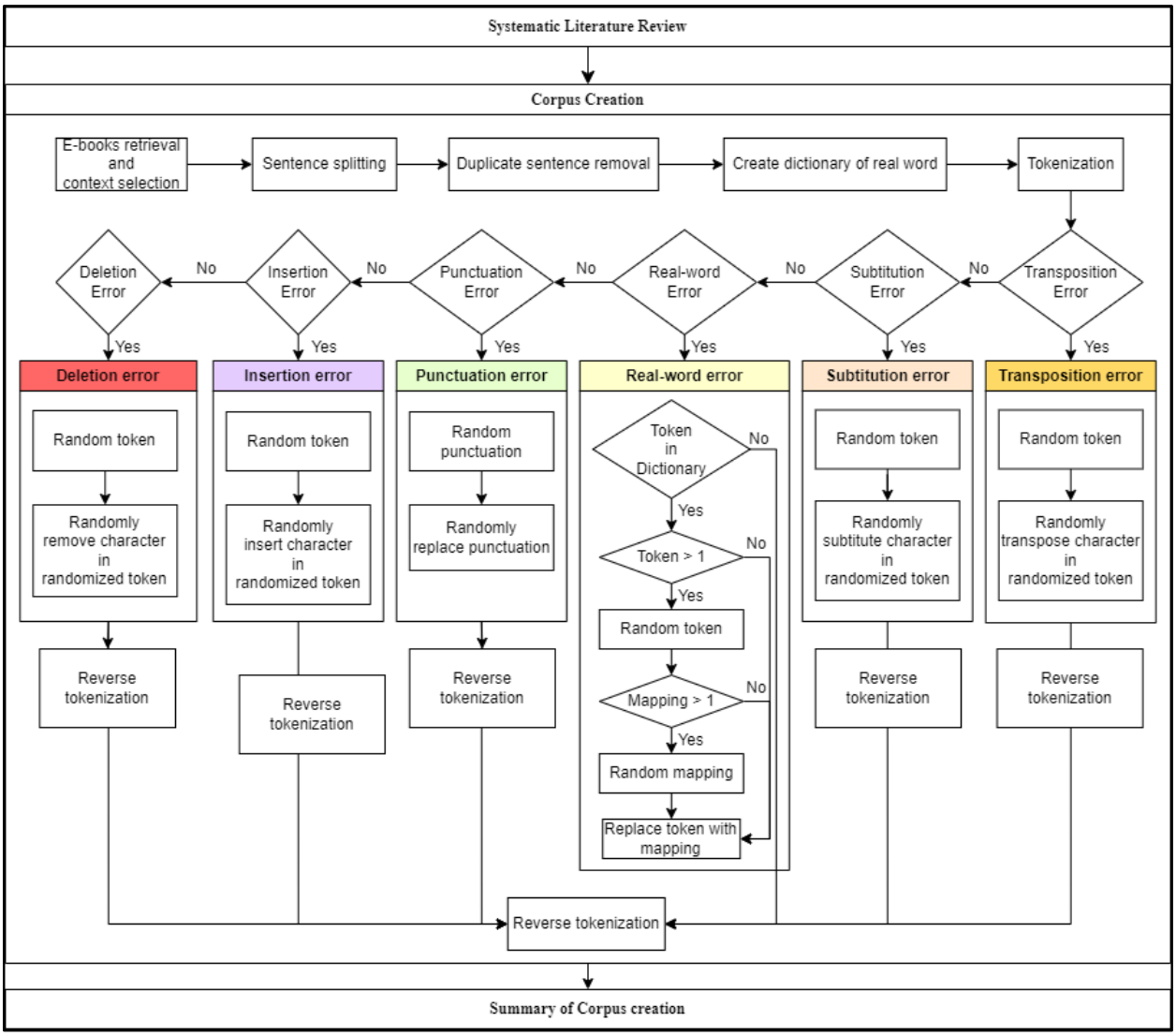

FIGURE 2. Block diagram of SPECIL corpus creation.

context, whereas non-word error means no meaning in the dictionary [10]. Non-word errors include transposition, substitution, insertion, and deletion [11]. Indonesian punctuation is used, such as in periods (.), commas (,), semicolons (;), colons (:), dashes (-), question marks (?), and exclamation points (!) [12]. Hence, we created six types of errors: transposition, substitution, insertion, deletion, real-word, and punctuation errors.

Our corpus consisted of correct sentences, incorrect sentences, and types of errors. Correct sentences were derived from data sources about the Indonesian language, Natural Sciences, and Social Sciences.

Real-word error is referred to as dictionary, which consists of 843 manually created words. Some words in the dictionary of real-word, such as "bisa" (can), "bisa" (poison), "busa" (foam), "makan" (eat), "maka" (so), "kapur" (chalk), "kasur" (mattress), "rumah" (house), "ruah" (abundant), etc. Incorrect sentences were created using the proposed algorithm.

C. SUMMARIZING CORPUS CREATION

In the last phase of this study, as a method for summarizing the corpus creation, we calculated different words and sentences to understand the results.

Each word in $\boldsymbol{w}$ appears in the beginning sentence $\boldsymbol{S}$ is determined as the below Equation (1) describes:

$$Score(S_n) = \sum Count_{unmatch}(w') \tag{1}$$

where,

$Score(S_n)$ = Evaluation score obtained at set sentenceS_n.
S = Set of beginning sentence
w = Set of word
$\sum Count_{unmatch}(w')$ = Count of all such words from wrong sentences S'not in the beginning sentence$S$.

Each word that appears in the wrong sentence is presented in Equation (2):

$$Score(S'_n) = \sum Count_{unmatch}(S') \tag{2}$$

where

$Score(S'_n)$ = Evaluation score obtained at set sentenceS'_n.
S' = Set of wrong sentence
$\sum Count_{unmatch}(S')$ = Count of all such words from wrong sentences S'not in the beginning sentence$S$.

We added the scores of all the evaluation sets to calculate the total number of different words. This is computed in Equation (3):

$$Total\_Different\_Word = \sum_{n=1}^{N} Score(S_n) \tag{3}$$

where,

$Total\_Different\_Word$ = The sum of evaluation scores obtained from all evaluation sets Sn.
N = Denote the total number of sentences sets.

The sum of each difference sentence in S is calculated in Equation (4):

$$Total\_Different\_Sentence = \sum_{n=1}^{N} Score(S'_n) \tag{4}$$

where, the total_different_sentence is the sum of the evaluation scores obtained from all evaluation sets S'n.

IV. LITERATURE ANALYSIS

There were 38 clean papers among the 51 papers discovered using the Scopus search. Figure 3 shows the year distribution for nine papers of the journal type and 29 papers of the conference type.

The VOSviewer tool was used to conduct the literature review to aid in the analysis and visualization of this study.

Corpus research cannot be separated from other variables that support the corpus concept. Figure 4 shows a chart of each keyword, with a connection site to the corpus. This corpus is closely related to natural language processing and received excellent attention in 2016. Furthermore, the Corpus is linked to a spell checker and error. When we plotted the keywords associated with spell checker (see Figure 5), we discovered another term related to error and error. Therefore, it is essential to note that the spell checker is inseparable from the error detection and error correction terms.

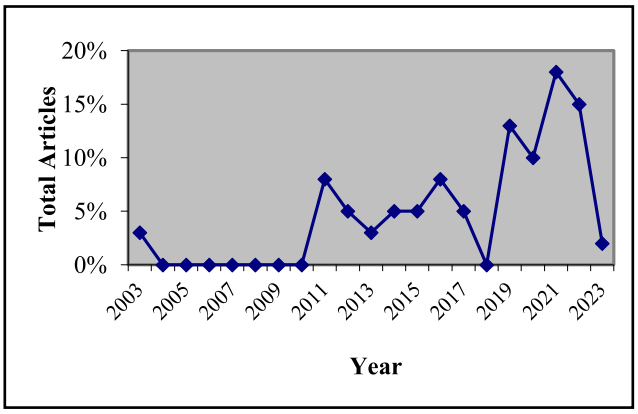

FIGURE 3. Distribution of papers related to NLP and spell by year.

TABLE 2. Research related to NLP and spell by languages.

Language	Total Document	Reference
Arabic	6	[13]–[18]
Bangla	6	[19]–[24]
English	6	[25]–[30]
Persian	3	[31]–[33]
Russian	2	[34], [35]
Spanish	2	[36], [37]
Albanian language	1	[38]
Amazigh	1	[39]
Burmese	1	[40]
Chinese	1	[41]
Croatian	1	[42]
Finnish	1	[43]
Kazakh	1	[44]
Mixed language	1	[45]
Moroccan Arabic	1	[46]
Myanmar	1	[47]
Setswana African language	1	[48]
Sinhala	1	[49]
Urdu	1	[50]
Indonesian language	-	

Table 2 illustrates some interesting facts about languages related to NLP and spell. Arabic, Bangla, and English have the most articles on corpora for spell checkers, followed by Persian, Russian, Spanish, Albanian, Amazigh, Burmese, Chinese, Croatian, Finnish, Kazakh, mixed language, Moroccan Arabic, Myanmar, Setswana African, Sinhala, and Urdu. However, there is no article on the corpus relating to spelling in the Indonesian language is found.

Besides, we also search in the website Google Scholar (https://scholar.google.com/) with same query since 2023, and got 4 articles on corpora for spell in NLP. The languages researched related to spell and NLP are Sindhi language [51], French-speaking Belgium [52], English for welding fabrication [53], and Kurdish language [54]. Since there is no corpus related to spells in the Indonesian language, we cannot make a comparison. Nevertheless, we conducted another review to identify the corpus available in Indonesian. There was no limited period.

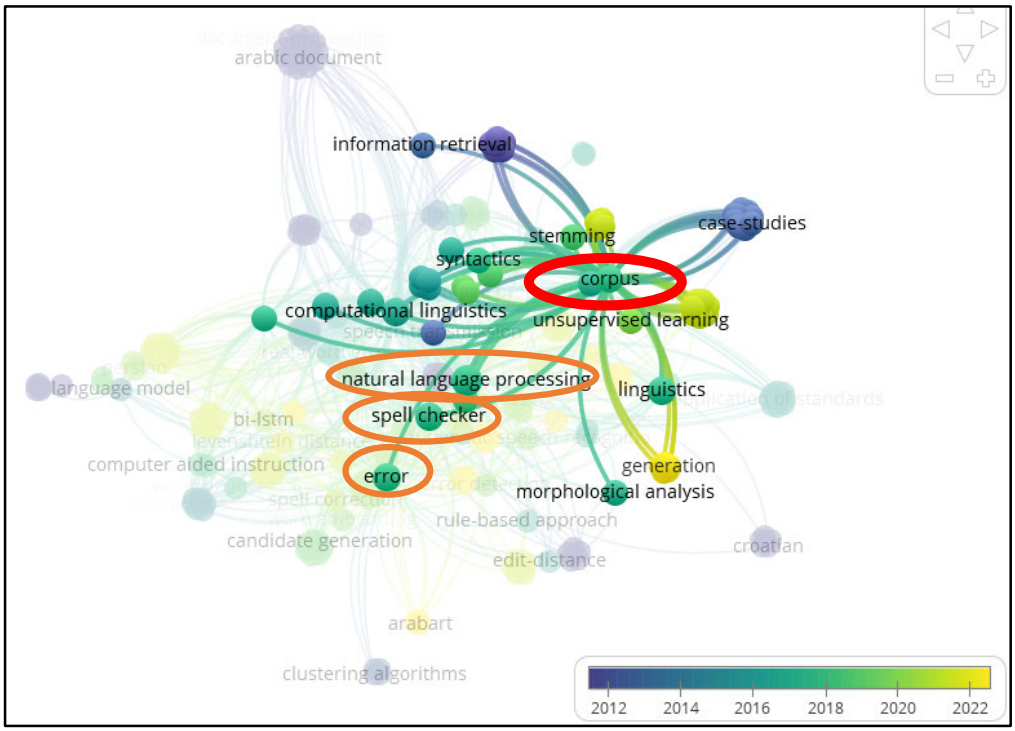

FIGURE 4. Mapping research corpus by year created in VOSviewer.

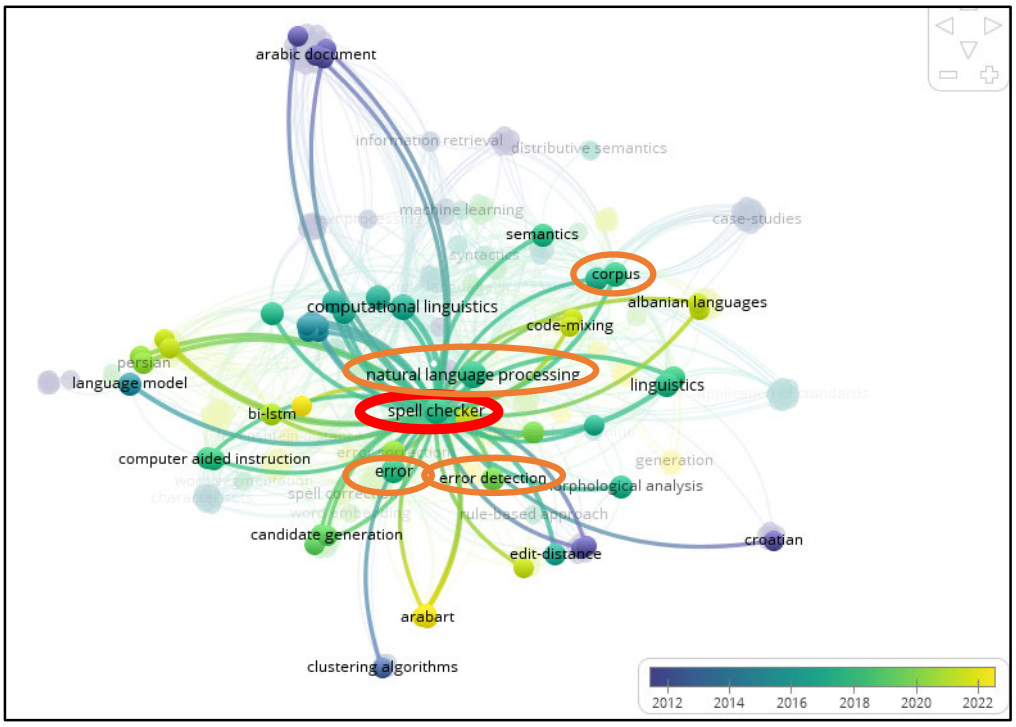

FIGURE 5. Mapping of spell checker keyword research by year created in VOSviewer.

The following search keywords within the paper title resulted in five papers collected from Scopus.com.

(TITLE (“develop*” OR “build*”) AND TITLE (corpus) AND TITLE (indonesia)) AND (LIMIT-TO (SRCTYPE, “j”) OR LIMIT-TO (SRCTYPE, “p”)) AND (LIMIT-TO (DOCTYPE, “ar”) OR LIMIT-TO (DOCTYPE, “cp”)) Table 3 presents some corpora available in Indonesian language; unfortunately, this is not the case in this study.

TABLE 3. Corpus available in the Indonesian language.

No.	Topic corpus	Author
1	Cyber bullying	Jambak, M.I., Setiawan, P.S. [55]
2	Management system	Uliniansyah, T., Riza, H., Riandi, O.[56]
3	Speech	Cahyaningtyas, E., Arifianto, D. [57]
4	Pornography	Gunawan, D., Lubis, S.A.H., Rahmat, R.F., Hizriadi, A. [58]
5	Pornography	Chandra, R., Sucipta Iskandar, M.A., Yuniar Banowosari, L., Suhendra, A., Prihandoko, P.[59]

V. RESULT OF CORPUS CREATION

Figure 2 shows that the creation of the SPECIL corpus started by retrieving e-books from three subjects: the Indonesian language, natural science, and social science. E-books were retrieved by manual downloading from http://buku.kemdikbud.go.id and https://annibuku.com/. Then, sentences are manually divided into rows with due observance of the rules, as stated in Section II. Then, redundant sentences were eliminated to ensure only one for each sentence. Moreover, a dictionary of real terms was created, the text was tokenized into words, and the type of error determined the process. Finally, a novel algorithm related to spelling with six kinds of errors in Indonesian was developed.

A. NON-WORD ERROR

Non-word errors include type errors, such as insertion, substitution, transposition, and deletion, as well as the block diagram shown in Figure 2.

1) INSERTION ERROR

Tokenizing the text into words was the first step, followed by selecting a random word from the tokens, selecting a random character from the alphabet, inserting the random character into the random word, and reverse tokenizing by replacing the original word with the final word in the text and capitalizing on the first letter of the final text.

The steps for the ''insertion_error'' function is summarized within the pseudocode in Algorithm 1.

Algorithm 1 Insertion Error Pseudocode

Input: selected subject datasets as beginning sentences.
Output: datasets of wrong sentences due to insertion error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
    tokenize (sentence)
    Choose a word randomly.
    For *each selected word in the sentence* **do**
        Add a letter randomly.
    End
    Reverse word in the sentence.
End
End

2) SUBSTITUTION ERROR

The process begins with tokenization, which involves selecting a random word from the tokens. Choose another random word if it has only one character. Then, a random character not appearing in the random word is selected. Finally, reverse tokenization was performed by replacing the original term with the final word in the text and capitalizing on the first letter of the final text.

The steps for the ''substitution_error'' function are illustrated in the pseudocode in Algorithm 2.

Algorithm 2 Substitution Error Pseudocode

Input: selected subject datasets as beginning sentences.
Output: datasets of wrong sentences due to substitution error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
    Tokenize (sentence).
    Choose a word randomly.
    For *each selected word in the sentence* **do**
        Substitute a letter randomly with another letter that is not in the word.
    End
    Reverse word in the sentence.
End
End

3) TRANSPOSITION ERROR

The process begins with tokenizing the text into words, selecting a random word from the tokens to swap, swapping characters, and reverse tokenization by replacing the original word with the final word in the text and capitalizing on the first letter of the final text.

Two functions were used for creating a corpus of transposition errors. The ''transposition_error()'' function generates a transposition error for a given text. In contrast, the ''get_transposition_error(()'' function is used to create a transposition error until the resulting text differs from the original text. The pseudocode for these functions is presented in Algorithm 3.

4) DELETION ERROR

The process started with tokenization: choosing a random word from the tokens, choosing a random character from the selected word, deleting the random character from the random word, and reversing tokenization by replacing the original word with the final word in the text and capitalizing on the first letter of the final text. The steps of the deletion error are illustrated in the pseudocode in Algorithm 4.

The corpus includes the Indonesian language, natural science, and social science. Each participant contained 21,500 sentences. Non-word errors comprise four types of errors: insertion error, substitution error, transposition error, and deletion error.

Table 4 presents the sample results for sentences with non-word errors and their calculations. The first sentence

Algorithm 3 Transposition Error Pseudocode

Input: selected subject datasets as beginning sentences.
Output: datasets of wrong sentences due to a transposition error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
  Tokenize (sentence).
  Choose a word randomly.
  For *each selected word in the sentence* **do**
    Transpose between two letters randomly.
  End
  Reverse word in a sentence.
End
End

Algorithm 4 Deletion Error Pseudocode

Input: selected subject datasets as beginning sentences.
Output: datasets of wrong sentences due to deletion error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
  Tokenize (sentence).
  Choose a word randomly.
  For *each selected word in sentence* **do**
    Delete a letter randomly.
  End
  Reverse word in sentence.
End
End

is "*Saya bisa belajar di rumah.*" (I can study at home.), containing five words. The same word is calculated by checking the total of the same words appearing in the sentence, while the different words are calculated by checking the total of the different words appearing in a sentence, as formulated in Equation 2.

A summary of these non-word errors is presented in Table 5. We remarked that each sentence had its own errors. The non-word error generated 86,106 different words for the Indonesian language, 86,091 different words for Natural Science, and 86,198 different words for Social Science.

B. REAL-WORD ERROR

The real-word error process, as shown in Figure 2, begins with tokenization and checking of tokens in the dictionary. Randomize the token if it is more than one. Then, we perform mapping based on the selected token. We then substitute the chosen token with the mapping result. Finally, reverse tokenization was completed.

There are three functions: subtitute_*kalimat_awal* (beginning sentence), get_error_result, and save_output. The steps of these functions are presented in the pseudocode of Algorithm 5.

TABLE 4. Result of non-word error: insertion error, substitution error, transposition error, and deletion error.

Kalimat awal (Beginning sentence)	*Kalimat salah* (Wrong sentence)	Total difference sentence	Total same word	Total different word
Type of error: Insertion				
Saya bisa belajar di rumah. (I can study at home.)	*Saya bisa belajar di* ***rumahh****.* (I can study at **homee**.)	1	4	1
Type of error: Substitution				
Saya bisa belajar di rumah. (I can study at home.)	*Saya bisa belajar di* ***rumsh****.* (I can study at **hoje**.)	1	4	1
Type of error: Transposition				
Saya bisa belajar di rumah. (I can study at home.)	*Saya bisa belajar di* ***ruamh****.* (I can study at **hoem.**)	1	4	1
Type of error: Deletion				
Saya bisa belajar di rumah.	*Saya bisa belajar di* ***ruma****.*	1	4	1
(I can study at home.)	(I can study at **hom**.)			
Summary of non-word error		4	16	4

TABLE 5. Summary of non-word error on three subjects: Indonesian language, natural science, and social science.

Subject	Total difference sentences	Total same words	Total different words
Indonesian Language	86,000	869,907	86,106
Natural Science	86,000	669,987	86,091
Social Science	86,000	750,381	86,198

TABLE 6. Results of real-world error on three subjects: Indonesian language, natural science, and social science.

Kalimat awal Beginning sentence	*Kalimat salah* Wrong sentence
Saya bisa belajar di rumah. (I can study at home.)	*Saya bisa belajar di* ***ruah****.* (I can study at **hope**.)

TABLE 7. Summary of real-word error on three subjects: Indonesian language, natural science, and social science.

Subject	Total same sentences	Total difference sentences	Total same words	Total different words
Indonesian Language	1,375	20,125	216,251	22,583
Natural Science	1,486	20,014	167,188	21,755
Social Science	1,933	19,567	187,066	21,973

The real-word error occurred when the word was correct but was used in the wrong context, so it changed the meaning. Example sentence for real-word errors is provided in Table 6.

A summary of real-word errors is presented in Table 7. The real-word error generated 22,583 different words for the Indonesian language, 21,755 different words for Natural Science, and 21,973 different words for Social Science.

In this result, there are still the same sentences because no words are found in the dictionary of real-word errors. As mentioned in Section III, the dictionary of manually created real words contained 843 words. Therefore, future studies should be conducted.

Algorithm 5 Real-Word Error Pseudocode

Input: selected subject datasets as beginning sentences, dictionary.
Output: datasets of wrong sentences due to real-word error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
    Tokenize (sentence).
    Find the word in dictionary.
    If *the word is in the dictionary > 1,* **then**
    Randomize word.
    End
    For*each selected word in sentence* **do**
    Find the mapped word in the dictionary.
    End
    If*mapped word > 1* **then**
        Choose a mapped word randomly.
    End
    Reverse word in sentence.
End
End

C. PUNCTUATION ERROR

Tokenization begins the real-word error process, as shown in Figure 2. Then, we select only one punctuation mark to be substituted at random. Finally, reverse tokenization is performed. The steps for punctuation errors are illustrated in the pseudocode in Algorithm 6.

Algorithm 6 Punctuation Error Pseudocode

Input: selected subject datasets as beginning sentences.
Output: datasets of wrong sentences due to punctuation error.
Begin
For *each sentence in selected subject datasets as beginning sentences* **do**
    Tokenize (sentence).
    Choose punctuation randomly.
    For each *selected punctuation in a sentence* **do**
        Replace a punctuation randomly.
    End
    Reverse word in sentence.
End
End

A summary of the punctuation errors is listed in Table 8. The punctuation error generated 21,500 different words for the Indonesian language, 21,500 different words for Natural Science, and 21,500 different words for Social Science. We confirmed that each sentence contained punctuation errors.

As there is no official corpus for identifying spell errors in Indonesia, we would like to highlight SPECIL, which has been made publicly available on Kaggle.com to improve upon various methodologies and use for future research efforts.

TABLE 8. Summary of punctuation error on three subjects: Indonesian language, natural science, and social science.

Subject	Total difference sentences	Total different marks
Indonesian language	21,500	21,500
Natural science	21,500	21,500
Social science	21,500	21,500

VI. CONCLUSION AND FURTHER WORK

This paper provides a review of publications relating to corpora and spelling and introduces the Spell Error Corpus for the Indonesian Language (SPECIL), which serves as a comprehensive resource for detecting and correcting spelling errors in Indonesian. SPECIL addresses the existing gap in this field by providing the first officially released corpus specifically designed for spelling error detection and correction in Indonesia. It has a vast collection of more than 180,000 tokens across 21,500 sentences, encompassing diverse types of errors, namely, non-word errors (deletion, substitution, insertion, transposition), real-word errors, and punctuation errors.

The SPECIL corpus has been made publicly available and will be of use to any researcher and anyone who needs it. This will save future researchers a lot of time and effort so that they can focus on their methodology instead of having to develop a corpus.

The corpus can be utilized in additional research to train and test different natural language processing (NLP) models, including spell checkers and language models, to enhance their precision and efficacy in identifying and correcting errors in Indonesian texts.

The creation of this error corpus, SPECIL, is an important step towards enabling the training and evaluation of NLP models, ultimately enhancing the accuracy of identifying and rectifying spelling mistakes in Indonesian texts. The availability of SPECIL to the research community and its potential to inspire further advancements make it a valuable contribution that deserves recognition and inclusion in this journal.

ACKNOWLEDGMENT

The authors would like to thank those who contributed to the success of this study and also would like to thank the readers. Without generous contributions, this study would not have been possible. They hope that you find this article valuable to your research and work.

REFERENCES

[1] N. Thalji, N. Adilah, Y. Yacob, and S. Al-Hakeem, "Corpus for test, compare and enhance Arabic root extraction algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 5, pp. 1–8, 2017. [Online]. Available: https://www.ijacsa.thesai.org

[2] C. Rosa Caldas-Coulthard and R. Moon, "Curvy, hunky, kinky': Using corpora as tools for critical analysis," *Discourse Soc.*, vol. 21, no. 2, pp. 99–133, Mar. 2010, doi: 10.1177/0957926509353843.

[3] A. C. Charles, L. Ruback, and J. Oliveira, "Fakepedia corpus: A flexible fake news corpus in Portuguese," *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13208. Cham, Switzerland: Springer, 2022, pp. 37–45, doi: 10.1007/978-3-030-98305-5_4.

[4] F. Koto, A. Rahimi, J. Han Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," 2020, *arXiv:2011.00677*.

[5] R. Mahendra. *IndoNLI: A Natural Language Inference Dataset for Indonesian*. Accessed: Mar. 1, 2023. [Online]. Available: https://www.babbel.com/en/magazine/

[6] H. Alwi, D. Sugono, and A. M. Moeliono, *Telaah Bahasa dan Sastra: Persembahan Kepada prof. Dr. Anton M. Moeliono* (Pusat Pembinaan dan Pengembangan Bahasa). Indonesia: Yayasan Obor, Departemen Pendidikan dan Kebudayaan, 1999.

[7] D P Nasional. (2008). *Pusat Bahasa*. Kamus Besar Bahasa Indonesia, no. 31. Accessed: Jul. 17, 2023. [Online]. Available: https://core.ac.uk/download/pdf/227147524.pdf

[8] N. Widyaningsih. (2016). *Kalimat Dalam Bahasa Indonesia*. Ukdw. p. 15, Accessed: Jul. 17, 2023. [Online]. Available: http://pustaka.unpad.ac.id/wp-content/uploads/2010/03/kalimat_dalam_bahasa_indonesia.pdf

[9] W. Tarmini and R. Sulistyawati. (2019). *Sintaksis Bahasa Indonesia*. [Online]. Available: https://www.academia.edu/download/60532213/SINTAKSIS-Rev-ok_edu20190909-54833-f9ee9h.pdf

[10] T. M. Fahrudin, I. Sa'diyah, L. Latipah, I. Z. Atha Illah, C. C. Bey Lirna, and B. S. Acarya, "KEBI 1.0: Indonesian spelling error detection system for scientific papers using dictionary lookup and Peter norvig spelling corrector," *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol. 12, no. 2, p. 78, Aug. 2021, doi: 10.24843/lkjiti.2021.v12.i02.p02.

[11] P. Santoso, P. Yuliawati, R. Shalahuddin, and A. P. Wibawa, "Damerau Levenshtein distance for Indonesian spelling correction," *J. Informatika*, vol. 13, no. 2, pp. 11–15, 2019, doi: 10.26555/jifo.v13i2.a15698.

[12] H. Harzoni, S. Suhirman, and I. Eliya, "Indonesian spelling errors in the description text," *Jadila, J. Develop. Innov. Lang. Literature Educ.*, vol. 2, no. 3, pp. 283–291, Feb. 2022, doi: 10.52690/jadila.v2i3.213.

[13] S. B. Aichaoui, N. Hiri, A. H. Dahou, and M. A. Cheragui, "Automatic building of a large Arabic spelling error corpus," *Social Netw. Comput. Sci.*, vol. 4, no. 2, p. 108, Mar. 2023, doi: 10.1007/S42979-022-01499-X.

[14] M. Alkhatib, A. A. Monem, and K. Shaalan, "Deep learning for Arabic error detection and correction," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 5, pp. 1–13, Sep. 2020, doi: 10.1145/3373266.

[15] A. Yahya. *Enhancement Tools for Arabic Web Search*. Accessed: Mar. 23, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5893871/

[16] M. Al-Jefri, S. Mahmoud, M. M. Al-Jefri, and S. A. Mahmoud, "Context-sensitive Arabic spell checker using context words and N-gram language models," in *Proc. Taibah Univ. Int. Conf. Adv. Inf. Technol. Holy Quran Sci.*, 2013, pp. 258–263, doi: 10.1109/NOORIC.2013.59.

[17] Z. Althafir and R. Ghnemat, "A hybrid approach for auto-correcting grammatical errors generated by non-native Arabic speakers," in *Proc. Int. Conf. Emerg. Trends Comput. Eng. Appl. (ETCEA)*, Nov. 2022, pp. 1–6, doi: 10.1109/ETCEA57049.2022.10009874.

[18] S. B. Aichaoui, N. Hiri, and M. A. Cheragui, "SPIRAL: SPellIng eRror parallel corpus for Arabic language," in *Proc. Int. Conf. Intell. Syst. Pattern Recognit.*, in Communications in Computer and Information Science, vol. 1589, 2022, pp. 248–259, doi: 10.1007/978-3-031-08277-1_21.

[19] S. Ismail and M. S. Rahman, "Bangla word clustering based on N-gram language model," in *Proc. Int. Conf. Electr. Eng. Inf. Commun. Technol.*, Apr. 2014, pp. 1–5, doi: 10.1109/ICEEICT.2014.6919083.

[20] T. Ahmed, S. Hossain, Md. S. Salim, A. Anjum, and K. M. Azharul Hasan, "Gold dataset for the evaluation of Bangla stemmer," in *Proc. 5th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2021, pp. 1–6, doi: 10.1109/EICT54103.2021.9733662.

[21] T. T. Urmi, J. J. Jammy, and S. Ismail, "A corpus based unsupervised Bangla word stemming using N-gram language model," in *Proc. 5th Int. Conf. Informat., Electron. Vis. (ICIEV)*, May 2016, pp. 824–828, doi: 10.1109/ICIEV.2016.7760117.

[22] T. Mittra, S. Nowrin, L. Islam, and D. C. Roy, "A Bangla spell checking technique to facilitate error correction in text entry environment," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–6, doi: 10.1109/ICASERT.2019.8934461.

[23] A. Husna, M. Mostofa, A. Khatun, J. Islam, and Md. Mahin, "A framework for word clustering of Bangla sentences using higher order N-gram language model," in *Proc. Int. Conf. Innov. Eng. Technol. (ICIET)*, Dec. 2018, pp. 1–6, doi: 10.1109/CIET.2018.8660791.

[24] M. N. Hoque and M. H. Seddiqui, "Bangla parts-of-speech tagging using Bangla stemmer and rule based analyzer," in *Proc. 18th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2015, pp. 440–444, doi: 10.1109/ICCITECHN.2015.7488111.

[25] D. Yang, X. Sun, and P. Wang, "Using neural machine translation for detecting and correcting grammatical errors," in *Proc. 20th Int. Conf. WWW/Internet Appl. Comput.*, 2021, pp. 11–18, doi: 10.33965/icwi_ac2021_2021091002.

[26] C. Wang and R. Zhao, "Multi-candidate ranking algorithm based spell correction," in *Proc. CEUR Workshop*, vol. 2410, 2019, pp. 1–8.

[27] S. Seneff, G. Chung, and C. Wang, "Empowering end users to personalize dialogue systems through spoken interaction," in *Proc. 8th Eur. Conf. Speech Commun. Technol. (Eurospeech)*, Sep. 2003, pp. 749–752, doi: 10.21437/EUROSPEECH.2003-312.

[28] R. C. De Amorim and M. Zampieri, "Effective spell checking methods using clustering algorithms," in *Proc. Recent Adv. Natural Lang. Process.*, 2013, pp. 172–178, Accessed: May 8, 2022. [Online]. Available: https://aclanthology.org/R13-1023.pdf

[29] D. Micol, R. Muñoz, and Ó. Ferrández, "Investigating advanced techniques for document content similarity applied to external plagiarism analysis," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process.*, 2011, pp. 240–246.

[30] H. Dutta and A. Gupta, "PNRank: Unsupervised ranking of person name entities from noisy OCR text," *Decis. Support Syst.*, vol. 152, Jan. 2022, Art. no. 113662, doi: 10.1016/J.DSS.2021.113662.

[31] F. H. Kermani and S. Ghanbari, "A partitional clustering approach to Persian spell checking," in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Feb. 2019, pp. 297–301, doi: 10.1109/KBEI.2019.8734932.

[32] T. Mosavi Miangah, "FarsiSpell: A spell-checking system for Persian using a large monolingual corpus," *Literary Linguistic Comput.*, vol. 29, no. 1, pp. 56–73, Apr. 2014, doi: 10.1093/llc/fqt008.

[33] M. S. Sartakhti, M. J. M. Kahaki, S. V. Moravvej, M. j. Joortani, and A. Bagheri, "Persian language model based on BiLSTM model on COVID-19 corpus," in *Proc. 5th Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Apr. 2021, pp. 1–5, doi: 10.1109/IPRIA53572.2021.9483458.

[34] A. S. Fenogenova, I. A. Karpov, V. I. Kazorin, and I. V. Lebedev, "Comparative analysis of Anglicism distribution in Russian social network texts," in *Proc. Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2017, pp. 65–74, Accessed: Mar. 23, 2023. [Online]. Available: https://www.dialog-21.ru/media/4114/fenogenova.pdf

[35] A. Fenogenova. *A General Method Applicable to the Search for Anglicisms in Russian Social Network Texts*. Accessed: Mar. 23, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7891854/

[36] J. López-Hernández, "Analysis and classification of errors for a proposal to improve medical reports in Spanish," in *Proc. CEUR Workshop*, vol. 2633, 2020, pp. 38–43.

[37] M. Melero, M. R. Costa-jussà, P. Lambert, and M. Quixal, "Selection of correction candidates for the normalization of Spanish user-generated content," *Natural Lang. Eng.*, vol. 22, no. 1, pp. 135–161, Jan. 2016, doi: 10.1017/S1351324914000011.

[38] D. N. Mati, M. Hamiti, B. Selimi, and J. Ajdari, "Building spell-check dictionary for low-resource language by comparing word usage," in *Proc. 44th Int. Conv. Information, Commun. Electron. Technol. (MIPRO)*, 2021, pp. 229–236, doi: 10.23919/MIPRO52101.2021.9597183.

[39] Y. Chaabi and F. A. Allah, "Amazigh spell checker using Damerau–Levenshtein algorithm and N-gram," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6116–6124, Sep. 2022, doi: 10.1016/j.jksuci.2021.07.015.

[40] E. P. P. Mon, Y. K. Thu, T. T. Yu, and A. Wai Oo, "SymSpell4Burmese: Symmetric delete spelling correction algorithm (SymSpell) for burmese spelling checking," in *Proc. 16th Int. Joint Symp. Artif. Intell. Natural Lang. Process. (iSAI-NLP)*, Dec. 2021, pp. 1–6, doi: 10.1109/ISAI-NLP54397.2021.9678171.

[41] Y.-M. Hsieh, M.-H. Bai, S.-L. Huang, and K.-J. Chen, "Correcting Chinese spelling errors with word lattice decoding," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 14, no. 4, pp. 1–23, Nov. 2015, doi: 10.1145/2791389.

[42] Š. Dembitz, M. Randić, and G. Gledec, "Advantages of online spellchecking: A croatian example," *Software, Pract. Exper.*, vol. 41, no. 11, pp. 1203–1231, Oct. 2011, doi: 10.1002/SPE.1037.
[43] J. Kauttonen, "Dialog modelling experiments with Finnish one-to-one chat data," in *Artificial Intelligence and Natural Language* (Communications in Computer and Information Science), vol. 1292. Cham, Switzerland: Springer, 2020, pp. 34–53, doi: 10.1007/978-3-030-59082-6_3.
[44] Z. Yessenbayev, Z. Kozhirbayev, and A. Makazhanov, "KazNLP: A pipeline for automated processing of texts written in Kazakh language," in *Proc. Speech Comput., 22nd Int. Conf. (SPECOM)*, Saint Petersburg, Russia, 2020, pp. 657–666, doi: 10.1007/978-3-030-60276-5_63.
[45] K. S. S. Varma, A. Chaluvadi, and R. Mamidi, "Corpus creation and language identification in low-resource code-mixed Telugu-English text," in *Proc. Conf. Recent Adv. Natural Lang. Process. Deep Learn. Natural Lang. Process. Methods Appl.* Shoumen, Bulgaria: INCOMA, 2021, pp. 744–752, doi: 10.26615/978-954-452-072-4_085.
[46] R. Tachicart and K. Bouzoubaa, "Moroccan Arabic vocabulary generation using a rule-based approach," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8538–8548, Nov. 2022, doi: 10.1016/J.JKSUCI.2021.02.013.
[47] A. M. Mon, "Spell checker for Myanmar language," in *Proc. Int. Conf. Inf. Retr. Knowl. Manage.*, Mar. 2012, pp. 12–16, doi: 10.1109/INFRKM.2012.6204974.
[48] M. A. Dibitso, P. A. Owolawi, and S. O. Ojo, "Part of speech tagging for Setswana African language," in *Proc. Int. Multidisciplinary Inf. Technol. Eng. Conf. (IMITEC)*, 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9015871/
[49] E. Jayalatharachchi, A. Wasala, and R. Weerasinghe, "Data-driven spell checking: The synergy of two algorithms for spelling error detection and correction," in *Proc. Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, 2012, pp. 7–13, doi: 10.1109/ICTer.2012.6422063.
[50] R. Aziz, M. W. Anwar, M. H. Jamal, and U. I. Bajwa, "A hybrid model for spelling error detection and correction for Urdu language," *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14707–14721, Nov. 2021, doi: 10.1007/S00521-021-06110-7.
[51] I. N. Sodhar, S. Sulaiman, and A. H. Buller, "Exploration of Sindhi corpus through statistical analysis on the basis of reality," *Indian J. Sci. Technol.*, vol. 16, no. 12, pp. 924–931, Mar. 2023, doi: 10.17485/IJST/v16i12.236.
[52] F. Meunier, I. Hendrikx, A. Bulon, K. Van Goethem, and H. Naets, "MulTINCo: Multilingual traditional immersion and native corpus. Better-documented multiliteracy practices for more refined SLA studies," *Int. J. Bilingual Educ. Bilingualism*, vol. 26, no. 5, pp. 572–589, May 2023, doi: 10.1080/13670050.2020.1786494.
[53] K. Guan, Z. Li, Y. Zhang, L. Zou, and X. Yang, "Information extraction and application for constructing guidance corpus of welding fabrication," in *Proc. Inst. Mech. Engineers, B, J. Eng. Manuf.*, Jan. 2023, Art. no. 095440542211477, doi: 10.1177/09544054221147705.
[54] M. Azzat, K. Jacksi, and I. Ali, "The Kurdish language corpus: State of the art," *Sci. J. Univ. Zakho*, vol. 11, no. 1, pp. 125–131, Feb. 2023, doi: 10.25271/sjuoz.2023.11.1.1123.
[55] M. I. Jambak and P. S. Setiawan, "The development of Bahasa Indonesia corpora for machine learning model in combating cyber bullying: A case study of the Indonesian 2017 capital city governor election," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 7, pp. 1971–1988, Mar. 2018.
[56] T. Uliniansyah, H. Riza, and O. Riandi, "Developing corpus management system for Bahasa Indonesia the 'Perisalah' project," in *Proc. Int. Conf. Oriental COCOSDA Held Jointly Conf. Asian Spoken Lang. Res. Eval. (O-COCOSDA/CASLRE)*, Nov. 2013, pp. 1–4, doi: 10.1109/ICSDA.2013.6709887.
[57] E. Cahyaningtyas and D. Arifianto, "Development of under-resourced Bahasa Indonesia speech corpus," in *Proc. 9th Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2018, pp. 1097–1101, doi: 10.1109/APSIPA.2017.8282191.
[58] D. Gunawan, S. A. H. Lubis, R. F. Rahmat, and A. Hizriadi, "Building the pornography corpus for Bahasa Indonesia based on TRUST+TM positif database," in *Proc. Int. Conf. ICT Smart Soc. (ICISS)*, Nov. 2019, pp. 1–5, doi: 10.1109/ICISS48059.2019.8969831.
[59] R. Chandra, M. A. Sucipta Iskandar, L. Yuniar Banowosari, A. Suhendra, and P. Prihandoko, "Building corpus in Bahasa Indonesia for pornographic indicated website content," in *Proc. 5th Int. Conf. Comput. Eng. Design (ICCED)*, Apr. 2019, pp. 1–5, doi: 10.1109/ICCED46541.2019.9161141.

YANFI YANFI (Graduate Student Member, IEEE) received the S.Kom. degree in computer science from Mercu Buana University and the M.T.I. degree in computer science from Bina Nusantara University, Indonesia, where she is currently pursuing the Ph.D. degree with the Doctor of Computer Science Program. She is also a Faculty Member of Bina Nusantara University. Her research interests include natural language processing, human–computer interaction, and multimedia.

REINA SETIAWAN received the bachelor's degree in information management from STMIK Bina Nusantara, Jakarta, Indonesia, in 1996, the master's degree in management from Pelita Harapan University, Banten, Indonesia, in 2005, and the Ph.D. degree from the BINUS Graduate Program—Doctor of Computer Science Program, Bina Nusantara University, Jakarta. She is currently a Faculty Member with the Computer Science Department, Bina Nusantara University. Her research interests include information retrieval, data mining, and text processing.

HARYONO SOEPARNO received the bachelor's degree in statistics and computation from IPB University, Bogor, Indonesia, in 1980, the master's degree in computer science from Western Michigan University, Michigan, USA, in 1987, and the Ph.D. degree in computer science from the School of Engineering and Technology, Asian Institute of Technology, Bangkok, Thailand, in 1995. He has been an Associate Professor in computer science with the School of Computer Science, Binus University, Jakarta, Indonesia, since 1984. He was a member of the National Research Council, Ministry of Research, Technology, and Higher Education, Indonesia, from 2011 to 2019, and a reviewer of research and innovation funded by multiple donors. He is also the Head of the Concentration in Computer Science, Doctor of Computer Science Program, Binus University. He is the coauthor of more than three books on interdisciplinary research in computer science with various application domains. His teaching experience and research interests include databases, software engineering, analysis of algorithms, advanced knowledge systems, machine learning, deep learning, and natural language processing.

WIDODO BUDIHARTO received the bachelor's degree in physics from the University of Indonesia, Jakarta, Indonesia, the master's degree in information technology from STT Benarif, Jakarta, and the Ph.D. degree in electrical engineering from the Institute of Technology Sepuluh Nopember, Surabaya, Indonesia. He took the Ph.D. Sandwich Program in robotics with Kumamoto University, Japan, and conducted postdoctoral research in robotics and artificial intelligence with Hosei University, Japan, where he was a Visiting Professor with the Erasmus Mundus French Indonesian Consortium (FICEM), France, and Erasmus Mundus Scholar with EU Universite de Bourgogne, France, in 2007, and in 2017 and 2016, respectively. He is currently a Professor in artificial intelligence with the School of Computer Science, Bina Nusantara University, Jakarta. His research interests include intelligent systems, data science, robotic vision, and computational intelligence.