**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 12, No. 3, September 2023, pp. 1491~1499
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i3.pp1491-1499 

# An adaptable sentence segmentation based on Indonesian rules

**Johannes Petrus[1], Ermatita[2], Sukemi[2], Erwin[2]**
[1]Department of Informatics, Universitas Multi Data Palembang, Palembang, Indonesia
[2]Department of Computer Science, Universitas Sriwijaya, Palembang, Indonesia

**Article Info**

***Article history:***

Received Apr 16, 2022
Revised Sep 10, 2022
Accepted Oct 11, 2022

***Keywords:***

Adaptive
Rule
Segmentation
Sentence boundary
Token

**ABSTRACT**

Sentence segmentation that breaks textual data strings into individual sentences is an important phase in natural language processing (NLP). Each word in the string that is added a punctuation mark such as a period, question mark, or exclamation point, becomes the location for splitting the string. Humans can easily see the punctuation and split the string into sentences, but not machines. Basically, the three punctuation marks also perform other functions so that the sentence segmentation process must really be able to detect whether a word marked with punctuation is a sentence boundary or not. This research proposes a sentence segmentation system called *segmentasi kalimat bahasa Indonesia* (SKBI) or Indonesian language sentence segmentation by applying a set of rules and can be used in Indonesian texts and can be adapted for English. There are 34 rules built with a combination of 27 fairly complete features that contribute to this research. The experimental results for the Indonesian text show that the SKBI is able to achieve an F1-Score of 96.89% and 97.07% for English. Both need to be improved but now better than previous research.



***Corresponding Author:***

Ermatita
Department of Computer Science, Universitas Sriwijaya
Palembang, Indonesia
Email: ermatita@unsri.ac.id

## 1. INTRODUCTION

Text processing is one of the most important parts of the natural language processing (NLP) system [1]. In NLP, machines are trained to understand and manipulate human language text [2]. A sentence is a series of words that express a complete thought [3]. Sentence segmentation is the task of breaking text into individual sentences for further processing which is done by detecting sentence boundaries by determining the beginning and end of the sentence [4], [5]. This task is the first step for several NLP applications such as document summarization, information extraction, machine translation, and syntactic parsing. Generally, activities related to text processing are influenced by the success of identifying sentences or words. Correctly recognizing sentence boundaries can speed up workflows and reduce the number of text preprocessing tasks [1].

It is relatively easy for humans to know sentence boundaries but not for machines or applications [6] so it needs the ability to detect sentence boundaries [3], [7]. The standard pattern of a sentence is beginning with a capital letter and ends with a word accompanied by a punctuation mark such as a period, question mark, or exclamation mark. Finding these punctuation marks in the text is the key to breaking the text into sentences. However, the function of these three does not always mark as the end of the sentence [8]. In most cases the period is also used in abbreviations, in dates, or in e-mail addresses and they are not signified as sentence breaks. Therefore, tokens that meet the pattern are declared as sentence boundary candidates and then processed to determine whether they really are end of sentence (EOS) or not end of sentence (NEOS).

This fact causes the determination of sentence boundaries to be very complex and difficult to complete [8], [9]. Although using neural architecture, the results are still imperfect [10]. Failure to segment sentences due to not being able to properly define sentence boundaries will have a negative impact on the text analysis process [1], [11], [12]. This indicates that sentence segmentation is a very important task [9] and not a trivial thing [13].

Many studies of sentence boundary detection are for English, but they are rarely explored for languages other than English [5], [8] including Indonesian. Research for Indonesian has been done by [14] which presents the development of a training dataset to optimize sentence boundary detection using the Indonesian translation of the Al-Quran with F measure 86.4%, [6] using a rule base by looking for patterns of sentence endings based only on a combination of spaces, capital letters or quotation marks. [15] Perform sentence tokenization to get the boundary of each sentence using deep learning with F1-Score 96.57%. [16] Also uses a deep learning approach to separate each sentence from Indonesian news documents with a better F1 score of 98.49%.

Some research outside Indonesian such as [17] to detect sentence boundaries based on modern standard arabic (MSA) transcript which can predict the boundary automatically. [11] Presented and evaluated a supervised machine learning approach to address abbreviations and sentence formation in German-language medical narratives. [3] Develop guidelines for annotating sentence boundaries in the legal field. [1] Detects sentence boundaries in speech transcripts and speech changes. [4] Provides sentence boundary detection in a mix of different text genres and languages. [18] Using rule-based with 21 features and classification with k-means able to produce an average F1-score of 96.58%. [5] Proposed a multitasking neural model to detect sentence beginnings without relying on punctuation in written texts, obtaining an F1 score of up to 98.07%.

In this study we propose a rule-based sentence segmentation system called SKBI. A rule-based approach is based on a set of predefined rules [19] that can be implemented for many purposes [20]. There were 34 proposed rules that were referred to [6] which succeeded in overcoming the ambiguity in the use of abbreviations, first and last name abbreviations, numbering, common abbreviations and foreign terminology but still cannot detect middle abbreviations of people's names.

Each rule not only checks for punctuation or capitalization of candidate tokens but also considers some features of its neighboring tokens. It is our contribution. The rules refer to the general guidelines for indonesian spelling or *pedoman umum ejaan bahasa Indonesia* (PUEBI). SKBI focuses on written text that includes punctuation marks and assumes that the text is written in a good form according to the rules of the language. In the well-written text, a few rules based are sufficient to successfully detect sentence boundaries.

SKBI is also expected to be used for English texts because there are similar rules regarding sentence boundaries between Indonesian and English. For this purpose, we compared the performance of the SKBI with a pre-existing sentence boundary detection system for English, namely python sentence boundary disambiguation (PySBD) [21] and the vanilla approach. The performance is measured by the confusion matrix with the results of the F1 score for Indonesian is 96.89% and for English is 97.07%. These results indicate that SKBI has a good performance compared to previous studies but still requires improvement.

## 2. METHOD

The SKBI model consists of 3 stages as shown in Figure 1. The first stage is the pre-processing stage which is intended to collect data and initial data processing, the second stage is to detect the status of sentences boundaries and the last stage is to combine all tokens whose status is not the end of the sentence into one sentence. Finally, a list of sentences that have been separated from one another will be obtained. The resulting sentence will be used for the next process.

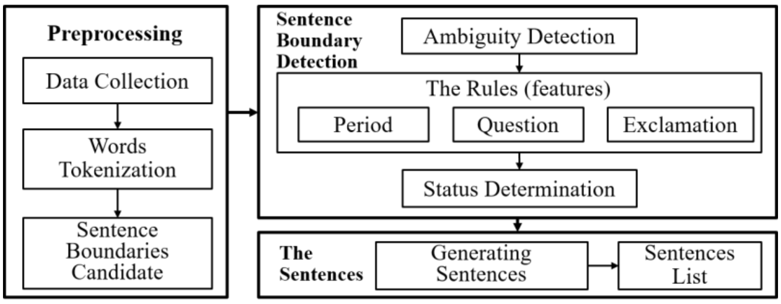



Figure 1. The sentence segmentation model

### 2.1. Preprocessing
#### 2.1.1. Data collection
Data collection is the first step before data processing activities can be carried out. Data in the form of text is collected from sources that are considered to use correct writing rules such as journals and electronic newspapers. Two data sets were collected, in Indonesian and English. In order for the rules to be properly tested, the data collected is limited to only text related to the use of punctuation marks. The statistics of the data collected can be seen in Table 1.

Table 1. The data set

| Data sets | Sources | Sentences | Tokens | Token with punctuation marks | | | |
|---|---|---|---|---|---|---|---|
| | | | | Periods | Questions | Exclamations | Total |
| Indonesian | 66 | 414 | 7,809 | 689 | 36 | 28 | 753 |
| English | 44 | 244 | 5,440 | 408 | 9 | 15 | 432 |

#### 2.1.2. Words tokenization
In NLP, tokenizing is the process of breaking a set of text into unit words [20], [22] or the procedure of splitting sentences into words [23]. These words are called tokens. The standard tokenization approach is word tokenization, i.e. breaking text into its constituent words using spaces as separators [24]. In SKBI, tokens and punctuation marks will be combined as one token. In terms of the use of punctuation marks, periods are the most dominant punctuation marks used for Indonesian and English, respectively 91.50% and 94.44%, question marks as much as 4.78% and 2.08%, and exclamation marks as much as 3.72% and 3.47%. Table 1 shows the statistics of the data set.

#### 2.1.3. Sentence boundaries candidate
Sentence boundaries are usually marked with a period, question mark, or exclamation mark as the last character of a token. However, the token cannot be directly confirmed as EOS, but is designated as a sentence boundaries candidate. This token will be further processed to obtain its actual status as the end of the sentence or not. Tokens whose last character is not one of these punctuation marks are declared immediately as NEOS. Out of 753 tokens in Indonesian, only 544 tokens are sentence boundaries candidates and from 432 English tokens there are 334 tokens for sentence boundaries candidates.

### 2.2. Sentence boundary detection
#### 2.2.1. The ambiguity detection
The general pattern of sentence boundaries is marked by the presence of a token with a period or exclamation mark or question mark then a space and the first letter of the next token is written in capital letters, as in the following example: "*Ibu pergi ke pasar. Ayah pergi ke kantor*." (Mother goes to the market. Father goes to the office.) According to the pattern above, *"pasar."* (market.) is a sentence boundary. So, the text can be segmented into two sentences, namely *"Ibu pergi ke pasar."* (Mother goes to the market.) as the first sentence and the second sentence is *"Ayah pergi ke kantor."* (Father goes to the office.).

But this pattern doesn't always work that way. For example, the text *"Alamat rumah Prof. Dr. Ratna Juwita terletak di jln. Veteran no. 10 Palembang".* (Home address of Prof. Dr. Ratna Juwita is located at Jl. Veteran no. 10 Palembang). There are 3 tokens that meet the general pattern criteria, namely "Prof.", "Dr." and "jln.", but these three tokens do not act as sentence boundaries. This ambiguity makes sentence segmentation complicated.

In this study, we detect ambiguity in obtaining segmented sentences using a rule-based approach based on Indonesian rules. The success of this task is very dependent on compliance in the use of punctuation. Other things that affect the results are the features of the candidate token and also the neighboring candidate tokens that precede or follow it. We propose more complete feature as our contribution.

#### 2.2.2. The rules
The term rule-based refers to any schemes using IF-THEN rules [25]. The advantage of this system is that the process is traceable and can add a number of new rules to get good results [20]. The rules that are used as learning representations are coded into the system, therefore the order of execution of the rules needs to be considered. The first rule that satisfies will be set as the output result. There are 34 rules are shown in Table 2, with the scope:

- Regarding the abbreviation indicating the region (example: jln., kel., kec., no., rt., rw.)
- Academic degree either before or after the person's name.
- Abbreviations of people's names, countries (example: A.H. Nasution, A.A. Navis, E.U.)

– Abbreviations for units of measure (example: kg., cm.)
– Abbreviations indicating personnel (example: a.n., u.p., d.a.)

Table 2. The rules in SKBI

| Rule # | Token precedes T(t-1) | Candidate token T(t) | Token following T(t+1) | Status | Rule # | Token precedes T(t-1) | Candidate token T(t) | Token following T(t+1) | Status |
|---|---|---|---|---|---|---|---|---|---|
| 01 | | F13, F25 | | NEOS | 17 | | F2, F12, F13 | F2 | NEOS |
| 02 | | F13, F10 | | NEOS | | F5 | | | EOS |
| | F1 | | | NEOS | 18 | F2 | F3, F12 | | EOS |
| 03 | | F13, F11 | | EOS | | | | F18 | NEOS |
| 04 | | F13, F18, F19 | | NEOS | 19 | F7 | F2, F12, F13 | F2 | NEOS |
| | | F13 | F2 | EOS | | | F17 | | EOS |
| 05 | | F2, F16, F12, F13 | F2 | NEOS | 20 | | F7, F13, F24 | F2 | EOS |
| | | F17 | | EOS | 21 | F6 | F7, F4, F12, F13 | F2 | EOS |
| 06 | F2, F3 | F2, F12, F13 | F2 | NEOS | 22 | | F7, F4, F12, F13 | F2 | NEOS |
| 07 | | F13 | F18, F26 | EOS | 23 | | F12, F13 | F6 | NEOS |
| | | F15 | | NEOS | 24 | | F13 | F2 | EOS |
| 08 | | F14, F13, F12, F7 | F2 | NEOS | 25 | | F13 | F6 | EOS |
| 09 | F8 | F13, F21 | F2, F16, #F27 | EOS | 26 | | F13 | F7 | NEOS |
| 10 | | F14, F13, F20 | F2 | EOS | 27 | | F13 | F27, F2 | EOS |
| 11 | | F14, F13 | F2 | NEOS | 28 | | F9, F11 | | EOS |
| 12 | | F14, F13 | F7 | NEOS | 29 | | F9, !F11 | F2 | EOS |
| 13 | | F13, F23 | | NEOS | 30 | | F9, !F11 | F7 | NEOS |
| 14 | F4, F7 | F2, F12, F13 | F15 | NEOS | 31 | F1 | F22 | | NEOS |
| 15 | F4, F18 | F2, F12, F13 | F2 | EOS | 32 | | F22, F5 | | NEOS |
| 16 | F4 | F2, F12, F13 | F2 | NEOS | 33 | | F22, !F11 | F2 | EOS |
| | | | | | 34 | | F22, !F11 | F7 | NEOS |

Notes: , = and, # = or, ! = not. For rules that are written in 2 lines, it means that there is a multilevel IF.

All the rules are formed based on the features on the candidate tokens and their neighbor either following or preceding. These features are the key. The contribution of our research lies in the creation of a set of sentence segmentation rules based on a number of token features as shown in Table 3. There are 27 features spread across 34 rules where 9 features are for tokens that precede candidate token T(t-1), 22 features are for candidate tokens T(t), and 9 features are for tokens that follow candidate token T(t+1). The rules will be tested sequentially from the first to the last rule. The first rule annotation that satisfies will be assigned to the token and the rest of the rules will be ignored.

Table 3. The features list

| | Description | | Description |
|---|---|---|---|
| F1 | Status as EOS | F15 | Starts with an opening parenthesis "(" |
| F2 | Starts with a capital letter | F16 | Second character is capital letter |
| F3 | Ends with a comma | F17 | The third character is a capital letter |
| F4 | Token length is more than 2 characters | F18 | Starting with a number |
| F5 | Maximum token length is 1 character | F19 | The second character is a period |
| F6 | In the form of numbers | F20 | Longer than 4 digits |
| F7 | Starting with lowercase | F21 | Ends with a closing parenthesis and a period |
| F8 | Does not end with a period | F22 | Ends with an exclamation mark |
| F9 | Ends with a question mark | F23 | Title in front of a person's name |
| F10 | As the first token | F24 | Maximum length 3 characters |
| F11 | As the last token | F25 | Only 1 character long |
| F12 | Maximum length 4 characters | F26 | Ends with a closing parenthesis “)” |
| F13 | Ends with a period | F27 | Starting with quotation marks |
| F14 | The number of periods is more than 1 | | |

The most widely used rules are those that conform to the general pattern of sentence boundaries, i.e. tokens are marked with a period followed by a space and the first capital letter of the next token. The usage of these rules reached 44.92% for Indonesian and 45.23% for English. Figure 2 shows the graph.

### 2.2.3. Status determination

After the sentence boundary candidate tokens are tested based on predefined rules, the status of each token will be obtained. There are only 2 statuses, namely EOS or NEOS. This status is very important for the next process. The data will be separated on tokens with EOS status.

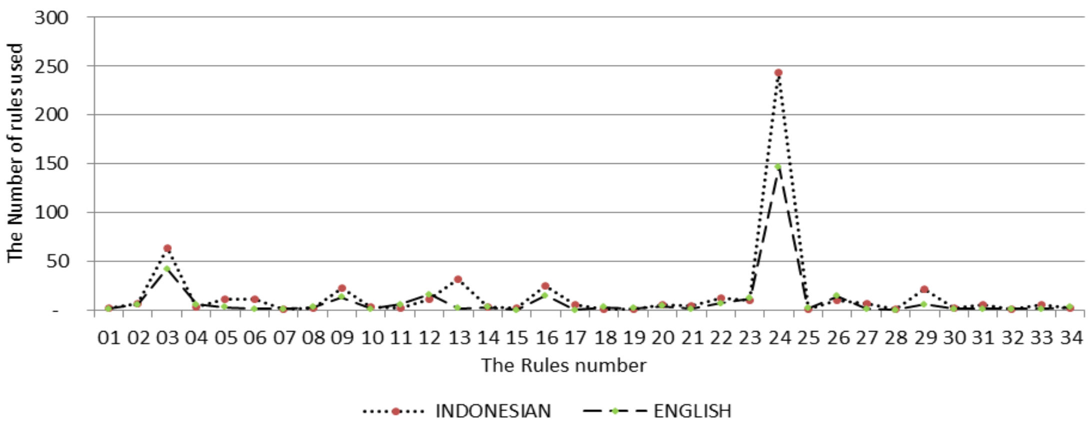



Figure 2. Graph of usage of each rule for Indonesian and English text

### 2.3. The sentences

#### 2.3.1 Generating sentences

Individual sentences are formed by concatenating all NEOS tokens and ending with EOS tokens. The next token marks the start of a new sentence. From the Indonesian language data set, SKBI was able to correctly predict 394 sentences out of 414 sentences and 20 sentences incorrectly. As for 244 sentences in English, 235 sentence boundaries were predicted correctly, only 9 sentences were still wrong. The success rate of SKBI in segmenting sentences is listed in Table 4.

Table 4. Sentence predictions and the percentage of success rates

| Data sets | Tokens | | Total Actual Sentences | Sentences | | Success Rate (%) |
|---|---|---|---|---|---|---|
| | Total | With punctuations | | Predicted Correctly | Predicted Wrongly | |
| Indonesian | 7,809 | 753 | 414 | 394 | 20 | 95.17% |
| English | 5,440 | 432 | 244 | 235 | 9 | 96.31% |

#### 2.3.2. Sentences list

The final result is the correct individual sentences generated from the data text. Each token with NEOS status will unite to form a sentence. The token with the EOS status becomes the last word in a sentence and the next token becomes the starting word for a new sentence. The data text as the input string is finally split into several sentences.

### 2.4. The algorithm

The process of detecting sentence boundaries begins by separating sentences for each word or token. The last character of the token is checked if it is one of the punctuation marks indicating the end of the sentence. Tokens are tested using existing rules to get token status. This activity is illustrated in the algorithm below. In this algorithm, string operations such as length of the string, and string's part extraction applied. Data Source: string text.

```
Count text lengths
Set positionStart
For each character do
Extract token
Append token in Array_list
End for
For each token do
Identification punct in token
If punct is period then CheckAmbiguity(period)
If punct is exclamation then CheckAmbiguity(exclamation)
If punct is question then CheckAmbiguity(questionMark)
List token, tokenStatus
End for
Append token (NEOS) in Sentences
List Sentences
Return
```

## 3. RESULTS AND DISCUSSION

The experimental results show that the rules in the SKBI are able to group sentences with fairly reliable results. Table 5 shows the statistical comparison between the actual data and the predicted data. The difference between the actual data and the predicted data shows that there are still inaccuracies in determining the status of tokens as either EOS or NEOS.

Table 5. Actual and predicted data

| Data sets | Number of data sources | Number of Tokens | Number of sentence boundaries candidate | Actual | | Predicted | |
|---|---|---|---|---|---|---|---|
| | | | | NEOS | EOS | NEOS | EOS |
| Indonesian | 66 | 7,809 | 544 | 130 | 414 | 150 | 394 |
| English | 44 | 5,440 | 334 | 90 | 244 | 99 | 235 |

### 3.1. Evaluation

The success level of the experimental results needs to be evaluated. Evaluation is used for improvement in future research. There are 2 evaluation methods used in this study, quantitative and qualitative.

#### 3.1.1. Quantitative evaluation

Quantitative evaluation was carried out based on the Confusion matrix with four parameters. The four parameters are: true positive (TP), false positive (FP), false negative (FN), and true negative (TN) [26]. An explanation with experimental data is shown in Table 6.

Table 6. Confusion matrix parameters

| Data sets | TP | FP | FN | TN |
|---|---|---|---|---|
| | When an EOS is correctly predicted. | When an EOS is wrongly predicted as NEOS. | When an NEOS is wrongly predicted as EOS. | When an NEOS is correctly predicted. |
| Indonesian | 390 | 21 | 4 | 126 |
| English | 232 | 11 | 3 | 85 |

SKBI performance is measured using confusion matrix. An NxN size table will be used to evaluate the performance of the SKBI. The matrix will compare the actual target value with the predicted one. Computed performance data in terms of accuracy, precision, recall, and F-1 score using the following formula [26]–[28]. SKBI performance results are shown in Table 7.

- Accuracy: To calculate the proportion of correct predictive value (TP+TN) among the total number of measured cases.

  Accuracy = (TP+TN) / (TP+FP+FN+TN) (1)

- Precision: precision to calculate the proportion of correct positive values to the total number of positive values both correctly and incorrectly predicted.

  Precision = TP / (TP+FP) (2)

- Recall: to calculate the proportion of correctly predicted positive values to the total of positive values that are correctly predicted and negative cases that are incorrectly predicted.

  Recall = TP / (TP+FN) (3)

- F1-Score: is a mean of an individual's performance, based on two factors i.e. precision and recall.

  F1-Score = (2*Precision*Recall) / (Precision + Recall) (4)

Table 7. SKBI performance for Indonesian and English texts

| Language | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Indonesian | 95.38% | 94.89% | 98.98% | 96.89% |
| English | 95,77% | 95,47% | 98,72% | 97,07% |

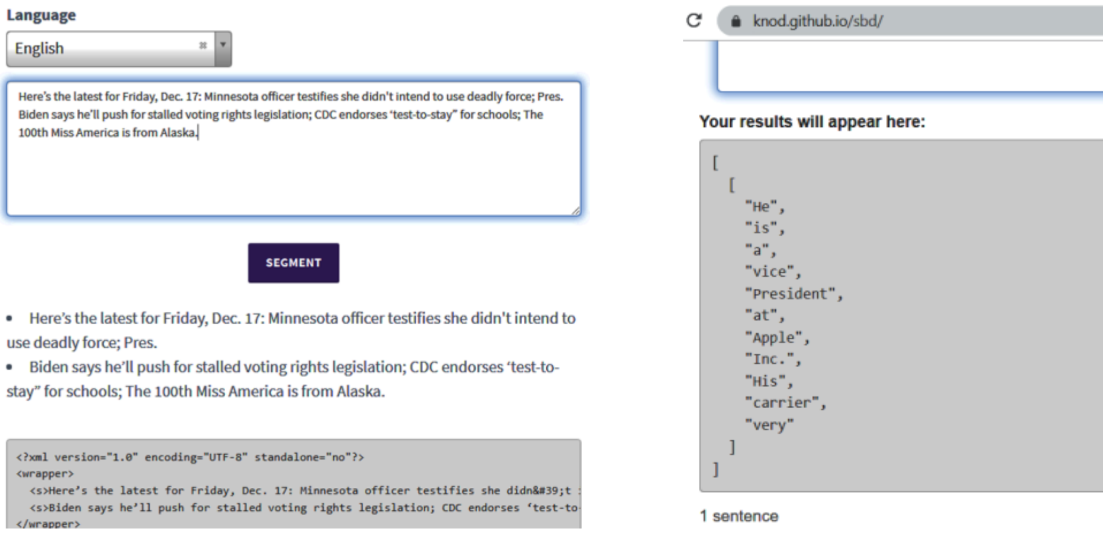
#### 3.1.2. Observation and findings

Apart from Indonesian, the SKBI can also be adapted for use in English texts. Therefore, the capabilities of the SKBI need to be compared with other similar systems. Another intended system is pySBD [21] and the vanilla approach which can be accessed at https://knod.github.io/sbd/.

One of PySBD's faults is that it doesn't assign EOS state to candidate tokens representing city or country names, even if subsequent tokens start with a lowercase letter. For example, in the text “They share in conversation while outside the U.S. Department of justice”. Abbreviation "U.S." with a period at the end, it's not really a sentence boundary because it's part of the "U.S. Department of Justice". PySBD predicts "U.S." as sentence boundaries.

In other texts such as “Minnesota officer testified that he had no intention of using lethal force; Pres. Biden says he will push for stalled voting rights laws;”. The period on “Pres. Biden” is still detected as a sentence boundary so the text is split into 2 sentences as shown in Figure 3(a).

Likewise with the vanilla approach, there are also errors in determining sentence endings, such as the text “He is a vice president at Apple Inc. His carrier very …”. The period on "Inc." is not detected as a sentence boundary, so the text continues and connects as one sentence. Figure 3(b) shows the process. Some problems with PySBD and vanilla approach can be handled well by SKBI.




(a) (b)

Figure 3. Error detecting English sentence boundary on (a) pySBD and (b) Vanilla approach

Benchmarking was performed using the same test data as the SKBI. The performance of these three systems is also measured by the confusion matrix. The results of the performance calculations are shown in Table 8.

Table 8. Confusion matrix for text in English

| System | TP | FP | FN | TN | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|---|---|---|
| SKBI | 232 | 11 | 3 | 85 | 95,77% | 95,47% | 98,72% | 97,07% |
| PySBD | 238 | 0 | 13 | 121 | 96,50% | 100% | 94,82% | 97,00% |
| Vanilla | 236 | 2 | 25 | 113 | 92,82% | 99,16% | 90,42% | 94,59% |

#### 3.1.3. Qualitative evaluation

Some mistakes in identifying sentence boundaries were also found. This error occurs because of features from the same context but are also used in different sentence structures. The currently defined rules are for general conditions only. More precise rules are needed.

1) *Guna menunjang kelancaran upaya PT. Garuda memberikan pelayanan…*
(…In order to support the smooth efforts of PT. Garuda provides services ...)
2) *Selaku Wakil Ketua DPR RI. Beberapa Penyempurnaan dalam…*
(... as the Vice Chairman of DPR RI. Some Improvements in ...)

3) Reduce blood sugar levels in mice at a dose of 400 mg/kg BW. Penetration of allicin can be...

In the example (1-3) above, "PT.", "RI.", and "BW." is a sentence boundaries candidate token. SKBI identified the three as part of the sentence (NEOS). In example 1) the candidate token is true as NEOS and categorized as True Positive. In examples 2) to 3) the actual status is sentence boundary (EOS) but predicted as NEOS, so it is categorized as False Positive. The position of the candidate token in example 1) is in the middle of the sentence while the example 2) to 3) is at the end of the sentence.

## 4. CONCLUSION

This study proposes a set of rules for sentence segmentation by considering the features of sentence boundaries candidate and their neighbors. These rules were tested with two different datasets, namely Indonesian and English. The text in the dataset uses punctuation that meets the criteria as sentence boundaries. SKBI achieved excellent sentence segmentation performance. For the dataset in Indonesian, the F1-Score is 96.89% better than the previous work of 86.4% and 97.07% for the English dataset, also better than the previous 96.58%. SKBI shows its reliability for segmenting sentences from English texts and perhaps also for other international languages that have similarities in the use of punctuation marks as sentence boundaries. However, in some cases, SKBI still incorrectly predicts sentence boundaries for candidates which only consist of a maximum of 3 digits and at the end of the sentence. For future research, it is important to learn more about sentence segmentation techniques for tokens that have similar features but different states and also expected to be implemented in many languages.

## REFERENCES

[1] M. S. U. Miah *et al.*, "Sentence boundary extraction from scientific literature of electric double layer capacitor domain: Tools and techniques," *Applied Sciences (Switzerland)*, vol. 12, no. 3, 2022, doi: 10.3390/app12031352.
[2] S. N. Khan *et al.*, "Urdu word segmentation using machine learning approaches," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 193–200, 2018, doi: 10.14569/IJACSA.2018.090628.
[3] J. Savelka, V. R. Walker, M. Grabmair, and K. D. Ashley, "Sentence boundary detection in adjudicatory decisions in the United States," *TAL Traitement Automatique des Langues*, vol. 58, no. 2, pp. 21–45, 2017.
[4] D. F. Wong, L. S. Chao, and X. Zeng, "I sentenizer-: Multilingual sentence boundary detection model," *The Scientific World Journal*, vol. 2014, 2014, doi: 10.1155/2014/196574.
[5] K. Lim and J. Park, "Real-world sentence boundary detection using multitask learning: A case study on French," *Natural Language Engineering*, 2022, doi: 10.1017/S1351324922000134.
[6] S. Raharjo, R. Wardoyo, and A. E. Putra, "Rule based sentence segmentation of Indonesia language," *J. Eng. Appl. Sci.*, vol. 13, pp. 8986–8992, 2018, doi: 10.3923/jeasci.2018.8986.8992.
[7] A. Singh, B. P. Singh, A. K. Poddar, and A. Singh, "Sentence boundary detection for Hindi–English social media text," *Advances in Intelligent Systems and Computing*, vol. 709, pp. 207–215, 2018, doi: 10.1007/978-981-10-8633-5_22.
[8] N. Wanjari, G. M. Dhopavkar, and N. B. Zungre, "Sentence boundary detection for Marathi language," *Physics Procedia*, vol. 78, pp. 550–555, 2016, doi: 10.1016/j.procs.2016.02.101.
[9] D. T. and A. Aghaebrahimian, "The sentence end and punctuation prediction in NLG text (SEPP-NLG) shared task 2021," *CEUR Workshop Proc.*, vol. 2957, no. 2016, 2021, doi: 10.21256/zhaw-23258.
[10] K. Sirts and K. Peekman, "Evaluating sentence segmentation and word tokenization systems on estonian web texts," *Frontiers in Artificial Intelligence and Applications*, vol. 328, pp. 174–181, 2020, doi: 10.3233/faia200620.
[11] M. Kreuzthaler and S. Schulz, "Detection of sentence boundaries and abbreviations in clinical narratives," *BMC Medical Informatics and Decision Making*, vol. 15, no. 2, 2015, doi: 10.1186/1472-6947-15-S2-S4.
[12] A. Michaud, O. Adams, T. A. Cohn, G. Neubig, and S. Guillaume, "Integrating automatic transcription into the language documentation workflow: Experiments with Na data and the Persephone toolkit," *Language Documentation and Conservation*, vol. 12, pp. 393–429, 2018.
[13] D. Griffis, C. Shivade, E. Fosler-Lussier, and A. M. Lai, "A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain," *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, vol. 2016, pp. 88–97, 2016, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/27570656%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5001746.
[14] S. J. Putra, M. N. Gunawan, I. Khalil, and T. Mantoro, "Sentence boundary disambiguation for Indonesian language," *ACM International Conference Proceeding Series*, pp. 587–590, 2017, doi: 10.1145/3151759.3156474.
[15] C. N. Purwanto, A. T. Hermawan, J. Santoso, and Gunawan, "Distributed training for multilingual combined tokenizer using deep learning model and simple communication protocol," *2019 1st International Conference on Cybernetics and Intelligent System, ICORIS 2019*, pp. 110–113, 2019, doi: 10.1109/ICORIS.2019.8874898.
[16] J. Santoso, E. I. Setiawan, C. N. Purwanto, and F. Kurniawan, "Indonesian sentence boundary detection using deep learning approaches," *Knowledge Engineering and Data Science*, vol. 4, no. 1, p. 38, 2021, doi: 10.17977/um018v4i12021p38-48.
[17] C. E. González-Gallardo, E. L. Pontes, F. Sadat, and J. M. Torres-Moreno, "Automated sentence boundary detection in modern standard Arabic transcripts using deep neural networks," *Procedia Computer Science*, vol. 142, pp. 339–346, 2018, doi: 10.1016/j.procs.2018.10.485.
[18] C. G. Chithra and E. Ramaraj, "Heuristic sentence boundary detection and classification," *International Journal on Emerging Technologies*, vol. 7, no. 2, pp. 199–206, 2016.
[19] H. Liu, A. Gegov, and M. Cocea, "Rule based systems for big data: A machine learning approach," *Rule Based Systems for Big Data: A Machine Learning Approach*, vol. 13, pp. 1–121, 2015, doi: 10.1007/978-3-319-23696-4.
[20] A. M. Aubaid and A. Mishra, "A rule-based approach to embedding techniques for text document classification," *Applied Sciences (Switzerland)*, vol. 10, no. 11, 2020, doi: 10.3390/app10114009.

[21] N. Sadvilkar and M. Neumann, "PySBD: Pragmatic sentence boundary disambiguation," pp. 110–114, 2020, doi: 10.18653/v1/2020.nlposs-1.15.

[22] A. A. Jalal and B. H. Ali, "Text documents clustering using data mining techniques," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 664–670, 2021, doi: 10.11591/ijece.v11i1.pp664-670.

[23] A. Berrajaa, "Natural language processing for the analysis sentiment using a LSTM model," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 777–785, 2022, doi: 10.14569/IJACSA.2022.0130589.

[24] V. S and J. R, "Text mining: Open source tokenization tools – an analysis," *Advanced Computational Intelligence: An International Journal (ACII)*, vol. 3, no. 1, pp. 37–47, 2016, doi: 10.5121/acii.2016.3104.

[25] A. M. Aubaid and A. Mishra, "Text classification using word embedding in rule-based methodologies: A systematic mapping," *TEM Journal*, vol. 7, no. 4, pp. 902–914, 2018, doi: 10.18421/TEM74-31.

[26] S. Senhadji and R. A. S. Ahmed, "Fake news detection using Naïve Bayes and long short term memory algorithms," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 2, pp. 746–752, 2022, doi: 10.11591/ijai.v11.i2.pp746-752.

[27] N. Seman and N. A. Razmi, "Machine learning-based technique for big data sentiments extraction," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 473–479, 2020, doi: 10.11591/ijai.v9.i3.pp473-479.

[28] A. Raut and R. K. Pandey, "Sentiment analysis using optimized feature sets in different twitter dataset domains," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 3035–3039, 2019, doi: 10.35940/ijitee.K2195.0981119.


## BIOGRAPHIES OF AUTHORS

**Johannes Petrus** is currently pursuing his doctorate in informatics engineering at Universitas Sriwijaya, Palembang. He is interested in machine learning and its application in language to solve everyday problems. He can be contacted at email: johannes@mdp.ac.id.

**Ermatita** is an Associate Professor at the Faculty of Computer Science Universitas Sriwijaya, Indonesia. Received her Doctorate degree in computer science from the Universitas Gadjah Mada, Yogyakarta, Indonesia. Her research interests are in Data Mining, Decision Support System, Knowledge Management. She is currently the Dean of faculty of computer science at Universitas Pembanguan Nasional Veteran, Jakarta. She can be contacted at email: ermatita@unsri.ac.id.

**Sukemi** received his Doctorate degree in computer science from the Universitas Indonesia, Jakarta, Indonesia. His research interests are in Computer Architecture, Operating System. He is currently the Head of Computer System Study Program at Universitas Sriwijaya, Palembang. He can be contacted at email: sukemi@unsri.ac.id.

**Erwin** received his Bachelor of Mathematics from Universitas Sriwijaya, Indonesia, in 1994, and an M.Sc. degree in Actuarial from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2002. Since 2012, he has been with the Department of Computer Engineering, Universitas Sriwijaya. Then, in 2019, he received his Doctorate in Engineering, Faculty of Engineering, Universitas Sriwijaya. His current research interests include image processing, and computer vision. He is also a member of IAENG and IEEE. He can be contacted at email: erwin@unsri.ac.id.