A Novel Approach: Tokenization Framework based on Sentence Structure in Indonesian Language

Johannes Petrus[1], Ermatita[2*], Sukemi[3], Erwin[4]
Informatics, Universitas Multi Data Palembang, Palembang, Indonesia[1]
Department of Computer Science, Universitas Sriwijaya, Palembang, Indonesia[2, 3, 4]

***Abstract*—This study proposes a new approach in the sentence tokenization process. Sentence tokenization, which is known so far, is the process of breaking sentences based on spaces as separators. Space-based sentence tokenization only generates single word tokens. In sentences consisting of five words, tokenization will produce five tokens, one word each. Each word is a token. This process ignores the loss of the original meaning of the separated words. Our proposed tokenization framework can generate one-word tokens and multi-word tokens at the same time. The process is carried out by extracting the sentence structure to obtain sentence elements. Each sentence element is a token. There are five sentence elements that is Subject, Predicate, Object, Complement and Adverbs. We extract sentence structures using deep learning methods, where models are built by training the datasets that have been prepared before. The training results are quite good with an F1 score of 0.7 and it is still possible to improve. Sentence similarity is the topic for measuring the performance of one-word tokens compared to multi-word tokens. In this case the multiword token has better accuracy. This framework was created using the Indonesian language but can also use other languages with dataset adjustments.**

***Keywords*—Token; tokenization; multi-word; sentence structure; sentence elements**

I. INTRODUCTION

In the current era, the amount of information is increasing very rapidly [1], a lot of information is available in text form from various types of documents such as magazines, e-books, research results, social media, emails, pdf files, video, audio, images, and large amounts of business content. Experts predict the volume of text documents will grow by 80% by 2025. To be useful, text data must be processed into information with text mining techniques [2].

To be processed, text data needs to be prepared at the text-pre-processing stage. This stage is the first important step of any data mining process to achieve better accuracy [3]. This process will change the data from its original form into a form that is easier to observe and explore [4]. One of the activities in pre-processing is tokenization besides case folding, filtering/stop-words removal, lemmatization, stemming [5], [6] including normalization and removing irrelevant words [7]. Stopwords are the least important words in a sentence, and ignoring them can help identify the most important words [8].

Tokenization is a fundamental process in almost all Natural Language Processing applications. The standard approach is single-word tokenization, in which the input string is split word by word using spaces as separators [9]. Most NLP research uses this kind of tokenization technique, such as by [10] in semantic similarity, [4][9] in text classification, [11], [12] in information retrieval, [13], [14] in clustering, [15]–[17] in sentiment analysis, and much more.

Usually tokenization separates each word in a sentence as one token based on the spaces between words, but in fact, not all words in a sentence can be separated. There are words that must remain in pairs so that the meaning of the sentence remains correct. Separating a sentence into its constituent words can result in the meaning of a word deviating far from its actual context [18].

There are several publications that state that tokens are not just one word, but can be several words or even one sentence [10][13][14][19]. There is also research into finding multi-word expressions (MWE) or combinations of words that must be paired to make sense, such as by [20]–[23]. Most of this research was conducted for documents in English and other languages, including languages that do not recognize spaces as separators between words, such as Mandarin, Japanese or Thai. Research on Indonesian language texts is still limited. All the research above is only for finding word pairs and not for tokenization.

Methods that have been used in previous research include statistics, linguistic, dictionaries, and machine learning. The statistical method calculates the frequency of co-occurrence of two words. Linguistic methods match grammatical patterns based on the types of word labels. Searching for word pairs in the dictionary, that's the dictionary method. Machine learning methods use a set of datasets to predict the output.

Tokens consisting of several words are referred to as multi-word tokens. Multi-word tokens must be in the same sentence and same sentence element. In paragraphs that contain many sentences, it is necessary to segment the sentences so that each sentence is separated from each other. In order to segment a sentence, it is very important to know where the sentence boundaries are. It is not easy to find sentence boundaries because there is ambiguity from sentence boundary punctuation.

In Indonesian there are 5 sentence elements, namely Subject (Subjek), Predicate (Predikat), Object (Objek), Complement (Pelengkap) and Adverb (Keterangan) known as SPOK in Indonesia [24]. The subject and predicate elements must be present, while others may or may not be present. Each sentence element contains one or more words as word pairs.

Word pairs can only be formed in the same sentence element. Therefore, it is important to be able to perform sentence structure extraction. This is not taken into account by previous studies. By extracting the sentence structure, each sentence element can be treated as a token, at least for Subject and Object. This paper proposes a new method for sentence tokenization based on sentence structure in Indonesian. This new method of sentence tokenization will generate single-word and multi-word tokens simultaneously. That's our contribution. To our best knowledge, there is no research on this. This research uses Indonesian, but can be adapted to other languages that use spaces by customizing and retraining the dataset.

To find out the effectiveness of single-word and multi-word tokens, a sentence similarity test was carried out on both types of tokens. From the test results, it shows that multi-word tokens are able to determine word similarity better than single-word tokens.

This paper divided into several sections. In Section II, we review the related work on multi-word tokenization including multi-word expression, Section III, we give an overview of the proposed method including sentence segmentation, sentence structure extraction and dataset preparation. Section IV, we provide the result and discussion, and finally, Section V, concludes this paper.

II. Literature Review

This paper is inseparable from the previous studies that have been conducted by researchers. The previous studies are summarized in this section, especially those related to multi-word tokenization. There are several methods used in previous research, such as statistics, linguistics, dictionary, and machine learning. We found two research in Indonesian language, that is [25] which perform 2-word extraction to obtain multi-word expression candidates by applying some rules and filtering using a dictionary. Researcher [26] also used rule-based methods and built two dictionaries (close class tagging and multi-word expression dictionary). This dictionary will store two or more words with POS tags of nouns, verbs and adjectives. The study [27] examines the tokenization process using a phrase detection-based approach.

Research in Serbian language with agricultural engineering domain conducted by [28] provides a hybrid approach by combining linguistic and statistical information. The Candidate terms are obtained using the frequency of occurrence of text sequences in the corpus. In an effort to obtain multi-word expressions, the author in [20] examined an implementation in Turkish used four methods: first, statistical methods to calculate high co-occurrence frequencies, second, linguistic methods through POS patterns, third, candidates from idiom dictionaries, and the last is specialized domains such as term dictionaries. Research that presents a method for identification of chemical terms as multi-words was conducted by [23]. In his research, the Multiword Identifying and Representing (MIR) method was implemented to recognize multi-word phrases in chemical literature with an unsupervised data-driven model and the identified phrases were added to the vocabulary. This research uses statistical and linguistic methods without expert annotations. Author in [29] created the MwTExt architecture, for automatic extraction of multi-word terms from unannotated computer science domain English documents. This method uses statistical, linguistic, and logic-based methods and hybrid techniques and focuses only on lexical patterns such as (N P N), (N P N + N), and (N P N P N).

The study [21] built a hybrid approach with the combination of Bi-LSTM + word correlation level and K-Means Clustering to detect MWEs for multiple languages without manual features. Author in [30] proposed a neural network model for learning fixed-size word representations from arbitrary chunks with word embedding. Implementation in French created MWE for Russian dictionary (RuThes). Multi-word expression recognition measure based on similarity of phrase distribution and word components is used for statistical and linguistic methods as well as for word embedding. Author in [31] focus on annotating different types of lexicalized and institutionalized phrases with main goal is to identify MWEs that are perceived as complex by readers and need to be simplified overall. A number of hand-crafted features form the basis for predicting MWE complexity.

From the previous research above, as far as we know, there is no research with a method based on sentence structure as proposed by this research.

III. Proposed Method

The general tokenization process is shown in Fig. 1. This process works by receiving input in the form of sentences and identifying each word as a token by using spaces as separators between words, resulting in single word token. The number of tokens equals the number of words.

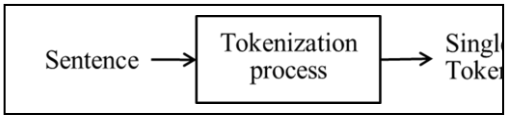

Fig. 1. General tokenization process.

This tokenization method is widely used, but it can also cause inaccuracies, such as:

1) The same word or token, will be considered to have the same meaning even if it is in a different order so that only one token will be used and the other tokens will be ignored [32]. Example :

Token in English : 'sakura','dewi', 'looks', 'at', 'sakura', 'tree', 'in', 'Japan.'

Token in Indonesian : 'sakura', 'dewi', 'memandang', 'pohon', 'sakura', 'di', 'Jepang'

The first token and the fifth token, will be considered to have the same meaning even though they are semantically different. One of them will be ignored.

2) When two or more words are combined and form a whole, a new meaning will be created that is different from each of the constituent words. Example :

Token in English: 'green table'

Token in Indonesian: 'meja hijau'

In Indonesian, ‘meja hijau’ means the court, a place to find the truth. If these two words are separated into ‘meja’ and ‘hijau’ then the meaning becomes different, the first is a piece of furniture that has a flat surface as a table top and legs as a support and the second is one of the base colors.

3) Not only the word meaning problem, but also the Part-of-Speech (POS) ambiguity problem. The POS of a single word token can vary. For example, separating the two words 'memberi makan' (in English: feeding), consists of the word 'memberi' with POS as the verb and the word 'makan' as the noun (since it is something that is given), but in other contexts such as 'kuda makan rumput', the POS of the word 'makan' is as a verb.

From the previous description, it is known that there are words that cannot be separated or must still be combined. Current tokenization methods does not accommodate this.

The main elements of the proposed tokenization framework are shown in Fig. 2. The framework has two stages, namely sentence segmentation and sentence structure extraction.

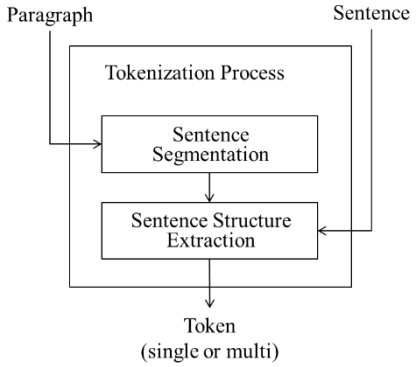

Fig. 2. The tokenization framework block diagram.

The input can be in the form of paragraphs or sentences. If the input is a paragraph, it will go through the sentence segmentation stage. This stage will split the paragraph into separate sentences. These sentences, whether they are new input or output from the first stage will be processed in the sentence structure extraction stage.

The output is a sentence structure with its elements (SPOK). Each sentence element is a token. In other words, sentence structure extraction is a tokenization process. These tokens are then used in natural language processing applications.

A. Sentence Segmentation

The task of sentence segmentation can be performed by detecting sentence boundaries [33]. The general pattern of a sentence is that it begins with a capital letter and ends with a special punctuation mark such as a period, question mark, or exclamation mark. The ability to recognize punctuation is a key requirement for knowing sentence boundaries to divide a paragraph into sentences. In this study the sentence segmentation process is described in Fig. 3.

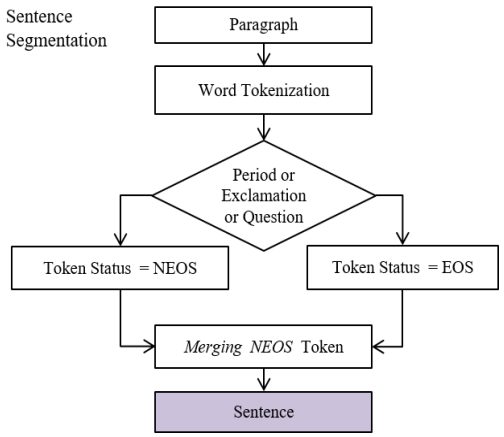

Fig. 3. Sentence segmentation diagram.

1) Word Tokenization, is a tokenization process as commonly used, breaking text data into words [8][34]. If there are punctuation marks then they will be attached to this token.

2) Punctuation Checking, is the process of checking the punctuation attached to the token, one of which is a period, question mark, or exclamation mark.

3) If the punctuation on the token is one of the three sentence-ending punctuation marks, the token will be assigned EOS status. Otherwise, it will be assigned NEOS status.

4) Combining NEOS Tokens. All NEOS will be combined into one sentence after finding EOS.

All tokens with NEOS status are combined into one new sentence and tokens with EOS status become the last word in the sentence. The next token will be the first word of the next sentence. This sentence will be used as input for the next process.

B. Sentence Structure Extraction

There are five sentence elements in Indonesian, namely Subject (Subjek), Predicate (Predikat), Object (Objek), Complement (pElengkap) and Adverb (Keterangan). Each sentence consists of at least a Subject and a Predicate and these two elements are arranged sequentially. The sentence elements Object, Complement and Adverb can be used or not used. The combination of these sentence elements forms a sentence structure pattern like SP, SPO, SPOK, SPOE, SPK, SPE, SPEK and SPOEK. Each word or words in each element of the sentence is a unit. Words or tokens that are in different sentence elements cannot be combined into one unit.

The sentence extraction process will identify sentence elements and classify each word in each sentence element.

This will facilitate the tokenization process, especially in determining multi-word tokens. The sentence structure extraction method in this study is as shown in the Fig. 4.

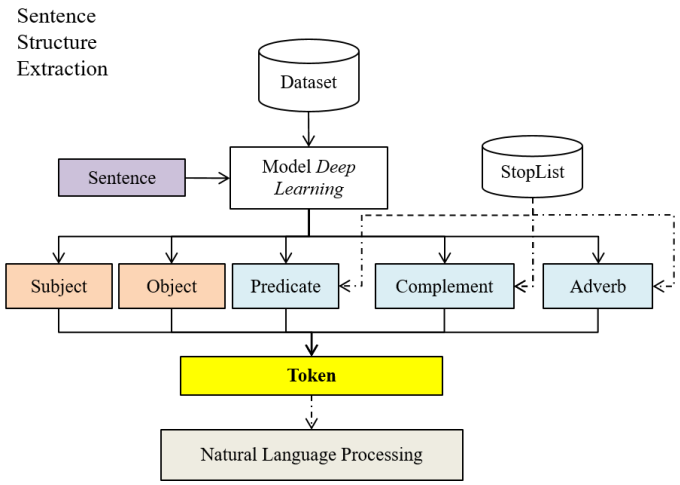

Fig. 4. Sentence structure extraction diagram.

The stages of the extraction process are as follows:

1) The process will accept input in the form of simple and active sentences.

2) A pre-trained deep learning model will predict sentence structure of the input sentence. The model has been trained using a dataset containing a collection of simple and active sentences in Indonesian, complete with labels. The embedded label is the identity of the sentence structure in the BIO tagging format. Label B (for "beginning") indicates as part of a multi-word token with position as the first word. The label I (for "inside") also indicates as part of a multi-word token with the position as the next word and the label O (for "outside") indicates as a stand-alone token or single word token. The dataset is in csv file format with an example as shown in Fig. 5.

This dataset contains 45,079 tokens from 4,740 sentences in Indonesian, with a minimum token range of 2 words and a maximum of 17 words per sentence. The distribution of each sentence element contained in the dataset is shown graphically in Fig. 6.

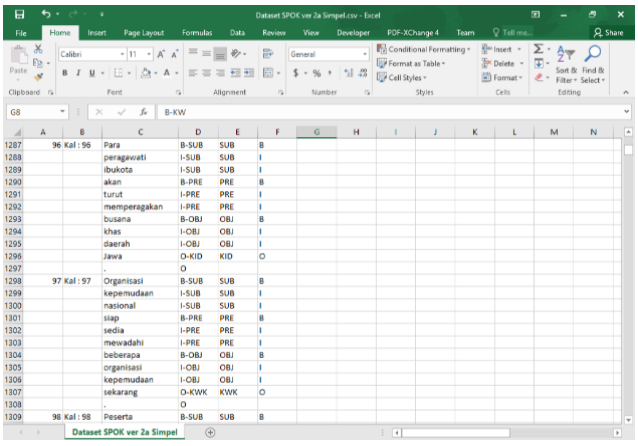

Fig. 5. Sentence structure dataset.

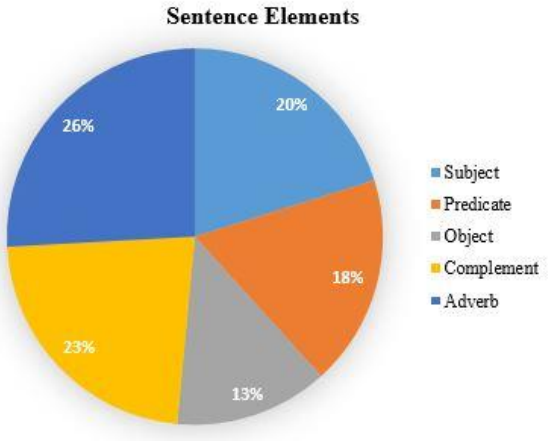

Fig. 6. Distribution of sentence elements in the dataset.

This dataset was trained using the pre-trained Bidirectional Encoder of Transformers (BERT) model. By dividing 80% as training data and 20% as test data and 10 epochs, an F1-score of 0.7 was obtained. These results show that the model and dataset that have been built are good enough, but need to be improved in the future.

3) The output of this process is a sentence structure prediction with sentence elements, namely Subject (SUB), Predicate (PRE), Object (OBJ), Complement (PEL), and Adverb (KET). There are nine types of adverbs in the dataset so there are thirteen sentence elements as listed in Table I.

Each token or word must be a member of one of the sentence elements. Each sentence element can consist of one or more than one word.

The output of the predicted sentence element will be written in the format of a BIO-tag label and the abbreviation of the sentence element, e.g. 'O-SUB' consists of the label O and the abbreviation SUB which means the word has no word pairs and with the Subject role.

TABLE I. SENTENCE ELEMENTS

No.	Sentence element		Abbreviation
1.	Subject	Subject	SUB
2.	Predicate	Predicate	PRE
3.	Object	Object	OBJ
4.	Complement	Complement	PEL
5.	Adverbs	Adverbs of time	KWK
6.		Adverbs of place	KTM
7.		Adverbs of purpose	KGU
8.		Adverbs of situation	KKD
9.		Adverbs of manner	KCR
10.		Adverbs of tools	KAL
11.		Adverbs of identity	KID
12.		Adverbs of participant	KPE
13.		Adverbs of condition	KSY

For sentence elements with more than one word, the first word will be labeled B ('beginning') and the remaining words will be labeled I ('inside' in BIO tags), e.g. 'B-SUB', 'I-SUB', 'I-SUB' which means there are three words that have the role of Subject and as one unit or one token. Such tokens are referred to as multi-word tokens. These tokens are then used in the NLP process.

IV. Result and Discussion

The experimental results of the proposed tokenization framework are quite good. In this section, the output will be discussed and sentence similarity tests will be conducted based on single word tokens and multi-word tokens.

A. The Output

As mentioned earlier, the outputs of this tokenization framework are sentence structures and sentence elements. Each sentence element can consist of a single word called a single-word token or multiple words called a multi-word token. One word means one token, multiple words also means one token. The number of sentence elements indicates the minimum number of tokens. Table II shows an example.

The first sentence consists of two words, the prediction results show that the first word is the Subject (O-SUB) and the second word is the Predicate (O-PRE). Both are independent because they are labeled O. Then each word is a single word token.

The second sentence consists of seven words. The first word 'Tim' is labeled 'B-SUB' and the second word 'Argentina' is labeled 'I-SUB' which indicates that both are in the same group which is Subject (SUB). So both should remain as one with the meaning of a group of soccer players from Argentina. Separating the two words will lose the original meaning. That is, the Subject is a combination of the words 'Tim' and 'Argentina' to become 'Tim Argentina'. This is a multi-word token.

Likewise, the fourth to seventh words are adverb groups (KTM), so these four words are a single unit. In this second sentence, there is also a word labeled 'O-PRE', namely 'win'. This means that the word 'win' has the role of a Predicate that stands alone, and is a single word token.

Therefore, it can be seen that the second sentence only has three tokens for Subject, Predicate, and Adverb. More details in Table III.

In the third sentence, there are three groups of sentence elements consisting of more than one word, namely words labeled Predicate (PRE), Object (OBJ), and Complement (PEL). Only the subject (SUB) stands alone because it is labeled O. The complete information can be seen in Table IV.

In Table IV, it is clear that the Subject is a one-word token 'Prajurit', the Predicate is a multi-word token 'mulai memasuki', on the Object there are two words 'area pertempuran' as multi-word tokens and the Complement consists of three words 'dengan senjata lengkap' as multi-word tokens.

B. Sentence Elements as Token

As explained earlier, a sentence element can be a token. A sentence extraction result that produces three sentence elements means it has three tokens. A sentence will have at least two tokens. Tokens can be single-word tokens or multi-word tokens.

TABLE II. Input and Prediction of Sentence Elements

No	Lang	Input Sentence	Output Prediction	
			Tokens	***Sentence Elements***
1.	INA	Amir mandi	[‘Amir’, ‘mandi’]	[‘O-SUB’, ‘O-PRE’]
	EN	Amir takes a bath		
2.	INA	Tim Argentina menang di Piala Dunia 2022	[‘Tim’, ‘Argentina’, ‘menang’, ‘di’, ‘Piala’, ‘Dunia’, ‘2022’]	[‘B-SUB’, ‘I-SUB’, ‘O-PRE’, ‘B-KTM’, ‘I-KTM’, ‘I-KTM’, ‘I-KTM’]
	EN	The Argentina team won in the 2022 World Cup		
3.	INA	Prajurit mulai memasuki area pertempuran dengan senjata lengkap.	[‘Prajurit’, ‘mulai’, ‘memasuki’, ‘area’, ‘pertempuran’, ‘dengan’, ‘senjata’, ‘lengkap’]	[‘O-SUB’, ‘B-PRE’, ‘I-PRE’, ‘B-OBJ’, ‘I-OBJ’, ‘B-PEL’, ‘I-PEL’, ‘I-PEL’]
	EN	Soldiers began to enter the battle area with full weapons.		

TABLE III. Sentence Structure for Example No. 2

Source	Tim Argentina menang di Piala Dunia 2022						
Initial Token	Tim	Argentina	menang	di	Piala	Dunia	2022
Output Labels	B-SUB	I-SUB	O-PRE	B-KTM	I-KTM	I-KTM	I-KTM
Sentence Elements	Subject		Predicate	Adverb of Place			
Proposed Token	‘Tim Argentina’		‘menang’	‘di Piala Dunia 2022’			
	Multi-word		Single word	Multi-word			

TABLE IV. Sentence Structure for Example No. 3

Source	Prajurit mulai memasuki area pertempuran dengan senjata lengkap.							
Initial Token	Prajurit	mulai	memasuki	area	pertempuran	dengan	senjata	lengkap
Output Labels	O-SUB	B-PRE	I-PRE	B-OBJ	I-OBJ	B-PEL	I-PEL	I-PEL
Sentence Elements	Subject	Predicate		Object		Complement		
Proposed Token	‘Prajurit’	‘mulai memasuki’		‘area pertempuran’		‘dengan senjata lengkap’		
	Single word	Multi-word		Multi-word		Multi-word		

However, not all multi-word tokens derived from sentence elements can be assigned as end tokens. The contents of multi-word tokens can be words that do not provide important information.

In the second sentence above, there is the word 'di' in the adverb of place with a multi-word token. The multi-word token

in the third sentence contains the word 'mulai' in the Predicate and the word 'dengan' in the Complement. These words can be ignored and have no effect on the token. Such words are known as stopwords.

From the example sentences above, stopwords can appear in Predicate, Complement, or Adverb. There are almost no stopwords in Subject and Object. Therefore, multi-word tokens in Predicate, Complement, and Adverb need to be filtered first. These unnecessary words will be removed before providing tokens. Filtering is done by comparing the contents of the multi-word tokens of the three sentence elements with a database containing words that fall into the category of stopwords.

C. Evaluation

The outputs of this framework are single word tokens and multi-word tokens. To get an overview of the two types of tokens, the following is an evaluation of both in determining sentence similarity.

The evaluation is done using the token lexical similarity method. Overlap Coefficient, Jaccards Index, Jaccards Distance, Dice Coefficient and Cosine Similarity methods will be used for single word tokens, while Dice-Index Coefficient for multi-word tokens.

Some of the stages of evaluation are as follows:

1) Defines a set of single-word tokens and multiple-word tokens in sentences.

2) Perform statistical calculations:

a) For single word token.

- Counts the number of tokens in the sentence, which is mathematically symbolized as $| K_1 |$.
- Counts the number of tokens that appear in both sentences, symbolized as $| K_1 \cap K_2 |$.
- Counts the number of tokens derived from the two sentences, and is symbolized as $| K_1 \cup K_2 |$.

b) For multi-word tokens:

- Counts the core (head) token on each token, symbolized as $| h_1 |$ and $| h_2 |$. Head is a word whose meaning is included in the meaning of another word.
- Perform token combinations according to the token order.
- Counts the number of core tokens (head) present in both multi-word tokens, symbolized as $| h_1 \cap h_2 |$.
- Sum the core (head) tokens, symbolized by $| h_1 | + | h_2 |$.
- Counts the number of tokens present in both multi-word tokens and is symbolized as $| M_1 \cap M_2 |$.
- Counts the number of tokens from both multi-word tokens, symbolized as $| M_1 | + | M_2 |$.

c) Measuring sentence similarity

Measuring the similarity between sentence1 and sentence2 basically determines how many similarity tokens there are in each sentence divided by the normalization factor.

The sentence similarity measurement function used is as follows:

- Overlap Coefficient: is the size of the overlap of the sets K_1 and the sets K_2 divided by the smallest size between K_1 and K_2.

$$OC(K_1, K_2) = \frac{| K_1 \cap K_2 |}{\min(| K_1 |, | K_2 |)} \quad (1)$$

- Jaccard Index: is the Intersection over Union size of the sets K_1 and K_2.

$$JI(K_1, K_2) = \frac{|K_1 \cap K_2|}{|K_1 \cup K_2|} = \frac{|K_1 \cap K_2|}{|K_1| + |K_2| - |K_1 \cap K_2|} \quad (2)$$

- Jaccard Distance: Measures the degree of difference of the two sets, or by subtracting 100% with the Jaccard Index.

$$JD(K_1, K_2) = 1\text{-}JI(K_1, K_2) = \frac{|K_1 \cup K_2| - |K_1 \cap K_2|}{|K_1 \cup K_2|} \quad (3)$$

- Dice Coefficient: measures two times the number of tokens shared in both sentences divided by the total number of tokens in both sentences.

$$DC(K_1, K_2) = \frac{2 | K_1 \cap K_2 |}{| K_1 | + | K_2 |} \quad (4)$$

- Cosine Similarity, with the formula:

$$CS(K_1, K_2) = \frac{K_1 \cdot K_2}{\| K_1 \| \cdot \| K_2 \|} \quad (5)$$

The following three sentences are used as test data.

1) K_1 = "walikota solo memberikan apresiasi kepada Agnes." (The mayor of solo city gave his appreciation to Agnes.).

2) K_2 = "agnes monica adalah penyanyi solo wanita berbakat." (agnes monica is a talented female solo singer.).

3) K_3 = "pemerintah kota solo mendapat hibah dari pangeran arab saudi." (she solo city government received a grant from the prince of saudi arabia.)

By using the formula described above, the calculation results are as follows in Table V. From the table, it can be concluded that the first sentence is more similar to the second sentence.

Meanwhile, the proposed tokenization process generates tokens according to the sentence structure as follows:

For K_1, S="walikota solo", P="memberikan", O="apresiasi", C="kepada agnes".

For K_2, S="agnes monica", P="adalah", O="penyanyi solo wanita", A="berbakat".

TABLE V. SENTENCE SIMILARITY FOR SINGLE WORD TOKEN

	K_1	**K_2**	**K_3**	**K_1, K_2**	**K_1, K_3**	**K_2, K_3**
$\| K_n \|$	6	7	9			
$\| K_x \cap K_y \|$				2	1	1
$\| K_x \cup K_y \|$				11	14	15

Overlap Coefficient	0.3333	0.1667	0.1429
Jaccard Index	0.1818	0.0769	0.0714
Jaccard Distance	0.8182	0.9231	0.9286
Dice Coefficient	0.3077	0.1429	0.1333
Cosine Similarity	0.3086	0.1443	0.1336

For K_3, S="pemerintah kota solo", P="mendapat", O="hibah", A="dari pangeran arab saudi".

Multi-word token similarity measurement uses the concept of lexical similarity based on identifying the common sequence of each token. It is based on the hypothesis that the head is a hyponym of the same term, which is denoted as hn. The visualization of the hyponyms of the multi-word tokens in the above three sentences is shown in the Fig. 7 below.

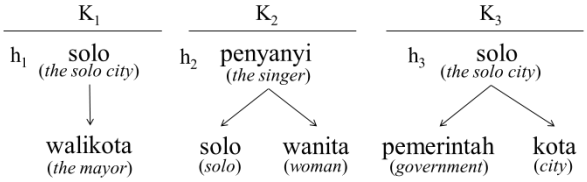

Fig. 7. Hyponyms referring heads.

The word sequence of the multiword token P(t) references the set of all sequences in t. The lexical similarity between multi-word tokens t_1 and t_2 is measured based on the Dice-like coefficient formula as follows:

$$KMK(M_1, M_2) = \frac{|P(h_1) \cap P(h_2)|}{|P(h_1)| + |P(h_2)|} = \frac{|P(t_1) \cap P(t_2)|}{|P(t_1)| + |P(t_2)|} \quad (6)$$

The numerator in the formula indicates the set of shared constituents (constituents present in both tokens), while the denominator refers to the total number of constituents.

The multi-word token obtained from sentence structure extraction are shown in Table VI.

TABLE VI. MULTI-WORD TOKEN

Stc	Multi-word Token	Core Token (head)		Constituent order	
		P(h)	|P(h)|	P(t)	|P(t)|
K_1	'walikota solo'	'solo'	1	{walikota, solo, walikota solo}	3
K_2	'agnes monica'	-	-	-	-
	'penyanyi solo wanita'	'penyanyi'	1	{penyanyi, solo, wanita, penyanyi solo, solo wanita, penyanyi solo wanita}	6
K_3	'pemerintah kota solo'	'solo'	1	{pemerintah, kota, solo, pemerintah kota, kota solo, pemerintah kota solo}	6

TABLE VII. SENTENCE SIMILARITY LEVEL FOR MULTIWORD TOKEN

Formula	Description	K_1, K_2	K_1, K_3	K_2, K_3
$\|P(h_x) \cap P(h_y)\|$	The sum of the same terms in both heads	0	1	0
$\|P(h_x)\| + \|P(h_y)\|$	The total number of terms on each head	2	2	2
$\|P(t_x) \cap P(t_y)\|$	The sum of the same terms in both constituents	1	1	1
$\|P(t_x)\| + \|P(t_y)\|$	The total number of terms on each constituents	9	9	12
Similarity		0.11	0.61	0.08

By using the Dice-like coefficient formula, the level of similarity of multiword tokens is obtained as shown in Table VII.

From the table above, the multi-word tokens in the first sentence are similar to the third sentence compared to the second sentence, and the multi-word tokens in the second sentence are very different from the third sentence.

From the similarity measurement of the two sentences above, there is a difference in results between single word tokens and multi-word tokens. The measurement with single word tokens concludes that the first sentence and the second sentence are more similar than the other sentences.

While the measurement with multi-word tokens states that the first sentence and the third sentence are more similar than the first and second sentences. Both have the same measurement result, that the second and third sentences are least similar.

In human judgment, the first and third sentences are similar, just like the measurement results of multi-word tokens. This shows that multi-word tokens also have advantages and can help NLP work.

D. Performace

To evaluate the quality of the proposed method, we conducted a manual evaluation of 100 sentences. The evaluation was done by checking the supposed multi-word tokens and then compared with the multi-word tokens extracted by the proposed method, with the results as shown in the Table VIII.

From the table, we can calculate Precision and Recall using the following formula:

$$P = \frac{\text{Correctly extracted multi-word tokens}}{\text{Total extracted multi-word tokens}} \quad (7)$$

$$R = \frac{\text{Correctly extracted multi-word tokens}}{\text{Multi-word tokens should be}} \quad (8)$$

TABLE VIII. EXTRACTED MULTI-WORD TOKEN

Number of sentences	Number of tokens	Correctly extracted sentence structure	Extracted		
			Correctly extracted multi-word tokens	Total extracted multi-word tokens	Multi-word tokens should be
100	709	84	204	221	237

And the results are P = 0.92 and R = 0.86. The success of extracting multi-word tokens correctly is quite dominant, out of 221 multi-word tokens extracted, 204 of them are correct. While the R value has a value of 0.86 which is obtained from 204 correct multi-word tokens out of 237 multi-word tokens that can be generated. These results provide information that

the proposed method is able to extract sentence structure and at the same time produce multi-word tokens that are quite accurate.

We also conducted a comparison with three other studies on multi-word tokens or similar from [21], [27] and [29]. Methods used by [21] are hybrid to train a multi-word expression detector for multiple languages without any manually encoded features. The methods used by [27] is a rule-based. The methods used by [29] are statistical, linguistic and logic-based methods and hybrid techniques, for the automatic extraction of multi-word terms from unannotated computer science domain English documents.

A comparison between these four methods is shown in Table IX.

TABLE IX. METHOD COMPARISON

Liang et al. [21]	**Putranto et al.** [27]	**Thanawala et al.** [29]	**Propose Method**
a hybrid approach, which combines Bi-directional LSTM (Bi-LSTM), phrase head word expansion and cluster to identify three types of multi-word expression	a rule that contains combinations of word classes that are most likely to form phrases.	Using shallow parsing and syntactic structure analysis and using a rule-based linguistic approach pattern.	Sentence structure extraction
compound nouns, verb construction and idiom.	Verbal, Nominal, Adverbial, Pronominal, Adjectival phrase rule.	Lexical patterns such as (N P N), (N P N + N), (N P N P N).	Dataset model
Sequence features, word correlation degree and three types of multi-word expression	The classification model obtained is more optimal.	Output in various forms of noun phrases	Output in form of sentence element
Precision = 0.92	0.79	0.87	0.92
Recall = 0.92	0.84	1.00	0.86

Each method has advantages and disadvantages. However, by preparing and training the sentence structure dataset, the proposed method is excellent in predicting the sentence structure elements. Each element is a token, either a single token or a multi-word token. Thus, this method does not rely on manually constructed lexical patterns. The method is highly adaptable and evolves as new data becomes available.

V. CONCLUSION

A tokenization process that generates single-word tokens and multi-word tokens simultaneously is possible. This is proposed through this research. To our knowledge, we are the first to propose this tokenization method based on sentence structure, which is expected to inspire new research with new ideas. Providing a complete dataset is a very important factor for successful sentence structure prediction. The predicted sentence element (SPOK) can consist of one or more words, i.e. tokens. Multi-word tokens are more accurate than single-word tokens in terms of sentence similarity.

Multi-word tokens are worthy of further research. In the future, we will enhance the dataset with passive sentences and also apply this approach for use in other types of cases such as NER.

REFERENCES

[1] El Haddadi, A. Fennan, A. El Haddadi, Z. Boulouard, and L. Koutti, "Mining unstructured data for a competitive intelligence system XEW," SIIE 2015 - 6th Int. Conf. "Information Syst. Econ. Intell., pp. 146–149, 2015, doi: 10.1109/ISEI.2015.7358737.

[2] R. Talib, M. Kashif, S. Ayesha, and F. Fatima, "Text Mining: Techniques, Applications and Issues," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 11, pp. 414–418, 2016, doi: 10.14569/ijacsa.2016.071153.

[3] M. M. Samia, A. Rajee, R. Hasan, M. O. Faruq, and P. C. Paul, "Aspect-based Sentiment Analysis for Bengali Text using Bidirectional Encoder Representations from Transformers (BERT)," vol. 13, no. 12, 2022.

[4] H. X. Huynh, L. X. Dang, N. Duong-Trung, and C. T. Phan, "Vietnamese Short Text Classification via Distributed Computation," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 7, pp. 23–31, 2021, doi: 10.14569/IJACSA.2021.0120703.

[5] M. Allahyari et al., "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," arXiv, no. July, 2017.

[6] R. A. Farouk, M. H. Khafagy, M. Ali, K. Munir, and R. M.Badry, "Arabic Semantic Similarity Approach for Farmers" Complaints," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 10, pp. 348–358, 2021, doi: 10.14569/IJACSA.2021.0121038.

[7] W. L. Roldan-Baluis, N. A. Zapata, and M. S. M. Vásquez, "The Effect of Natural Language Processing on the Analysis of Unstructured Text: A Systematic Review," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, pp. 43–51, 2022, doi: 10.14569/IJACSA.2022.0130507.

[8] N. M. Ibrahim, W. M. S. Yafooz, A. M. Emara, and A. Abdel-wahab, "Utilizing Deep Learning in Arabic Text Classification Sentiment Analysis of Twitter," vol. 13, no. 12, pp. 830–838, 2022.

[9] M. Usman, Z. Shafique, S. Ayub, and K. Malik, "Urdu Text Classification using Majority Voting," Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 8, pp. 265–273, 2016, doi: 10.14569/ijacsa.2016.070836.

[10] I. A. Norabid and F. Fauzi, "Rule-based Text Extraction for Multimodal Knowledge Graph," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, pp. 295–304, 2022, doi: 10.14569/IJACSA.2022.0130535.

[11] M. Krallinger, O. Rabal, A. Lourenço, J. Oyarzabal, and A. Valencia, "Information retrieval and text mining technologies for chemistry," Chem. Rev., vol. 117, no. 12, pp. 7673–7761, 2017, doi: 10.1021/acs.chemrev.6b00851.

[12] S. P. Panda, V. Behera, A. Pradhan, and A. Mohanty, "A Rule-based Information Extraction System," Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 9, pp. 1613–1617, 2019, doi: 10.35940/ijitee.i8156.078919.

[13] J. Joseph and J. R. Jeba, "Information Extraction using Tokenization and Clustering Methods," Int. J. Recent Technol. Eng., vol. 8, no. 4, pp. 3690–3692, 2019, doi: 10.35940/ijrte.d7943.118419.

[14] S. A. Fahad, "Design and Develop Semantic Textual Document Clustering Model," J. Comput. Sci. Inf. Technol., vol. 5, no. 2, pp. 26–39, 2017, doi: 10.15640/jcsit.v5n2a4.

[15] H. Juwiantho, E. I. Setiawan, J. Santoso, and M. H. Purnomo, "Sentiment Analysis Twitter Bahasa Indonesia Berbasis Word2Vec Menggunakan Deep Convolutional Neural Network," J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 1, pp. 181–188, 2020, doi: 10.25126/jtiik.202071758.

[16] E. W. Pamungkas and D. G. P. Putri, "An experimental study of lexicon-based sentiment analysis on Bahasa Indonesia," Proc. - 2016 6th Int. Annu. Eng. Semin. Ina. 2016, pp. 28–31, 2017, doi: 10.1109/INAES.2016.7821901.

[17] H. Sudira, A. L. Diar, and Y. Ruldeviyani, "Instagram Sentiment Analysis with Naive Bayes and KNN: Exploring Customer Satisfaction of Digital Payment Services in Indonesia," 2019 Int. Work. Big Data Inf. Secur. IWBIS 2019, pp. 21–26, 2019, doi: 10.1109/IWBIS.2019.8935700.

[18] A. Hamzah, A. Susanto, F.Soesianto, and J. E. Istyanto, "Perbandingan Feature Kata dan Frasa dalam kinerja Clustering dokumen teks berbahasa Indonesia," in Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2007, no. SNATI, p. B-53-B-58.

[19] U. Rahardja, T. Hariguna, and W. M. Baihaqi, "Opinion mining on e-commerce data using sentiment analysis and k-medoid clustering," Proc. - 2019 12th Int. Conf. Ubi-Media Comput. Ubi-Media 2019, pp. 168–170, 2019, doi: 10.1109/Ubi-Media.2019.00040.

[20] S. K. Metin and M. Taze, "A procedure to build multiword expression data set," 2nd Int. Conf. Comput. Commun. Syst. ICCCS 2017, pp. 46–49, 2017, doi: 10.1109/CCOMS.2017.8075264.

[21] Y. Liang, H. Tan, H. Li, Z. Wang, and W. Gui, "A language-independent hybrid approach for multi-word expression extraction," Proc. Int. Jt. Conf. Neural Networks, vol. 2017-May, pp. 3273–3279, 2017, doi: 10.1109/IJCNN.2017.7966266.

[22] S. Agrawal, R. Sanyal, and S. Sanyal, "Hybrid method for automatic extraction of multiword expressions," Int. J. Eng. Technol., vol. 7, no. 2, pp. 33–38, 2018, doi: 10.14419/ijet.v7i2.6.10063.

[23] L. Huang and C. Ling, "Representing Multiword Chemical Terms through Phrase-Level Preprocessing and Word Embedding," ACS Omega, vol. 4, no. 20, pp. 18510–18519, 2019, doi: 10.1021/acsomega.9b02060.

[24] D. Gunawan, H. P. Siregar, and O. Salim Sitompul, "Identifying Sentence Structure in Bahasa Indonesia by Using POS Tag and LALR Parser," 5th Int. Conf. Comput. Eng. Des. ICCED 2019, 2019, doi: 10.1109/ICCED46541.2019.9161125.

[25] D. Gunawan, A. Amalia, and I. Charisma, "Automatic extraction of multiword expression candidates for Indonesian language," Proc. - 6th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2016, no. November, pp. 304–309, 2017, doi: 10.1109/ICCSCE.2016.7893589.

[26] F. Rashel, A. Luthfi, A. Dinakaramani, and R. Manurung, "Building an Indonesian rule-based part-of-speech tagger," Proc. Int. Conf. Asian Lang. Process. 2014, IALP 2014, pp. 70–73, 2014, doi: 10.1109/IALP.2014.6973521.

[27] H. A. Putranto, O. Setyawati, and W. Wijono, "Effect of Phrase Detection with POS-Tagger on Sentiment Classification Accuracy using SVM," J. Nas. Tek. Elektro dan Teknol. Inf., vol. 5, no. 4, pp. 252–259, 2016, doi: 10.22146/jnteti.v5i4.271.

[28] V. Pajić, S. Vujičić Stanković, R. Stanković, and M. Pajić, "Semi-automatic extraction of multiword terms from domain-specific corpora," Electron. Libr., vol. 36, no. 3, pp. 550–567, 2018, doi: 10.1108/EL-06-2017-0128.

[29] P. Thanawala and J. Pareek, "MwTExt: automatic extraction of multi-word terms to generate compound concepts within ontology," Int. J. Inf. Technol., vol. 10, no. 3, pp. 303–311, 2018, doi: 10.1007/s41870-018-0111-6.

[30] J. Legrand and R. Collobert, "Phrase Representations for Multiword Expressions," Proc. 12th Work. Multiword Expressions, no. 2011, pp. 67–71, 2016, doi: 10.18653/v1/w16-1810.

[31] E. Kochmar, S. Gooding, and M. Shardlow, "Detecting multiword expression type helps lexical complexity assessment," Lr. 2020 - 12th Int. Conf. Lang. Resour. Eval. Conf. Proc., no. May 2020, pp. 4426–4435, 2020.

[32] J. Rabelo, M. Y. Kim, and R. Goebel, "Combining similarity and transformer methods for case law entailment," Proc. 17th Int. Conf. Artif. Intell. Law, ICAIL 2019, pp. 290–296, 2019, doi: 10.1145/3322640.3326741.

[33] K. Lim and J. Park, "Real-world sentence boundary detection using multitask learning: A case study on French," Nat. Lang. Eng., pp. 1–21, 2022, doi: 10.1017/S1351324922000134.

[34] A. A. Jalal and B. H. Ali, "Text documents clustering using data mining techniques," Int. J. Electr. Comput. Eng., vol. 11, no. 1, pp. 664–670, 2021, doi: 10.11591/ijece.v11i1.pp664-670.