TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 4, August 2020, pp. 1917~1925
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i4.14389

Single document keywords extraction in Bahasa Indonesia using phrase chunking

I Nyoman Prayana Trisna, Arif Nurwidyantoro
Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia

Article Info

Article history:

Received Oct 23, 2019
Revised Mar 5, 2020
Accepted Mar 18, 2020

Keywords:

Keyword candidate
Keyword extraction
Phrase chunking
Phrase pattern
Single document

ABSTRACT

Keywords help readers to understand the idea of a document quickly. Unfortunately, considerable time and effort are often needed to come up with a good set of keywords manually. This research focused on generating keywords from a document automatically using phrase chunking. Firstly, we collected part of speech patterns from a collection of documents. Secondly, we used those patterns to extract candidate keywords from the abstract and the content of a document. Finally, keywords are selected from the candidates based on the number of words in the keyword phrases and some scenarios involving candidate reduction and sorting. We evaluated the result of each scenario using precision, recall, and F-measure. The experiment results show: i) shorter-phrase keywords with string reduction extracted from the abstract and sorted by frequency provides the highest score, ii) in every proposed scenario, extracting keywords using the abstract always presents a better result, iii) using shorter-phrase patterns in keywords extraction gives better score in comparison to using all phrase patterns, iv) sorting scenarios based on the multiplication of candidate frequencies and the weight of the phrase patterns offer better results.

Corresponding Author:

Arif Nurwidyantoro,
Department of Computer Science and Electronics,
Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada,
Bulaksumur, Yogyakarta, Indonesia.
Email: arifn@mail.ugm.ac.id

1. INTRODUCTION

Keywords are a sequence of words or phrases that represent the content of a document [1, 2]. Keywords are included primarily to help readers quickly understand the main idea of the document. Furthermore, keywords can also be utilized to support content analysis of a document, such as content matching [3], document categorization [4], document browsing [5], and document summarization [6]. However, coming up with keywords manually needs considerable time and effort [2, 7]. Moreover, subjectivity is another problem that may arise in manual keywords generation due to the different backgrounds, knowledge, experience, etc. of the authors. Automatic keyword extraction techniques provide a means to address this subjectivity limitation of manual keyword creation [5, 8].

Keywords can be extracted from both a single document or a collection of documents. Extracting keywords from a single document, as the focus of this paper, aims to obtain the essential representation of that document [9]. On the other hand, extracting keywords from a collection of documents is usually intended to compare and classify each document into some categories. Previous research proposed various

methods to extract keywords automatically in a single document. These approaches can be categorized into 5 similar approaches: statistic approach, linguistic approach, graph-based approach, machine learning approach, and the combination of these approaches [10, 11].

Jonathan and Karnalim represented automatic keyword extraction as a classification problem. They utilized statistical features of candidate keywords, such as term frequency (TF), inverse document frequency (IDF), and the combination of both (TF-IDF) [12]. TF-IDF also used as a feature for the classifier by Zhang et al. [13], among other word features. Among all of these features from previous research, TF is the only feature applicable to single document keywords generation because IDF needs a collection of documents. The result from Jonathan and Karnalim [12] shows that using TF gives a decent performance, but not as good as using IDF. We considered TF as a scenario in our study, but we argue that using a classification approach with machine learning in keyword extraction could be troublesome due to the possible imbalance set between keyword and non-keyword candidates.

Another method ranks candidate keywords to determine keywords from a single document based on the statistical features of the document content. For example, Imran et al. [14] proposed a linguistic-based model called visualness with lesk disambiguation (VLD) to gather candidate keywords followed by word sense disambiguation and visual similarity to determine the important keywords. Combining statistical and linguistics approaches, the top candidate of keywords are acquired by measuring how well the candidates describe the document title. Unfortunately, this technique only able to generate single-word keywords, whereas, in reality, it is common for a keyword to have more than one word [15].

Another approach called TextRank is proposed by Mihalcea and Tarau [16]. This method transformed the document text into a graph in which the nodes are the words and the edges are the co-occurrence between corresponding words in several sizes of windows. TextRank is then modified into SingleRank and ExpandRank [17] which could give an improved result over TextRank but were still outperformed by the TF-IDF approach [18]. TextRank is then further improved by Li and Wang [19] with known keywords as domain knowledge to generate the phrase network. The addition of domain knowledge provides a better result than the original TextRank, but domain knowledge of document is not always available. On a similar approach using graph-based, Lahiri et al. [20] examined a number of centrality measures such as degree, weighted degree, and PageRank to determine the rank of keyword candidates. Their result showed that even though there was an improvement, the use of centrality measures could not outperform the use of TF and TF-IDF.

A different approach is proposed by Bhowmik [21] which generates candidate keywords from the title and the abstract of a document using the Perceptron Training Rule. Bhowmik also used the location of keyword candidates in a paragraph as the weight for selecting those candidate keywords. The result showed that using only the document's abstract is not sufficient for keywords generation. Lu et al. [22] proposed another approach by utilizing the references of the document as the keywords' source using the LocalMax algorithm. Both Bowmik [21] and Lu et al. [22] provided performance evaluations by comparing the generated keywords with the actual keywords from the document dataset. We adopt this approach to evaluate the performance of our keywords generator.

Different from other previous studies, Shukla and Kakkar proposed an approach called Noun-Phrase Chunking [23]. This method only considers noun phrases, which are manually specified, to extract phrases from the text document. This chunking method only extracts candidate keywords with the longest phrase. All extracted candidate keywords were assigned as keywords, without any selections or evaluations.

The phrase chunking method is straightforward because it uses part of speech (POS) patterns to extract the keyword's phrases. This method is also flexible in terms of providing control to determine which phrase patterns used to extract the keywords. The previous phrase chunking method proposed by Shukla and Kakkar [23] used several self-defined patterns. We argue that using self-defined patterns may lead to the limited coverage of phrase pattern possibilities, i.e., some keyword patterns may appropriate but not selected. In this paper, we proposed to address the limitation of this phrase chunking method by automatically generate patterns using a collection of documents. Therefore, the contributions of this research are as follows:

a. We proposed approaches for generating phrase patterns from a collection of documents, as an alternative to self-defined patterns proposed by Shukla and Kakkar [23],
b. We attempted to reduce the number of keywords by conducting a candidate reduction as a part of our experiment,
c. We conducted experiments using several keywords extraction scenarios, such as different sources of keywords, candidate keywords selections, and phrase patterns variations, and provide performance comparisons between those different scenarios.

Our experiment include a replication of the method proposed by Shukla and Kakkar [23], but for Bahasa Indonesia, with several adjustments: 1) we generated phrase patterns from the documents instead of

using self-defined patterns, 2) we reduced the number of yielded keywords from candidate instead of using all candidates as keywords. Compared to their method with the aforementioned adjustments, our result showed an improvement of performance based on the f-measure of the extracted keywords.

2. RESEARCH METHOD

We proposed an automatic keyword extraction technique using phrase chunking. Figure 1 shows the methodology of this research. First, we collected and prepared a collection of documents as our dataset. Then, we labeled each word in every document in the dataset with its corresponding part of speech (POS) automatically by utilizing an automated tagger. Afterward, we developed phrase patterns based on the part of speech of the keywords in the dataset. Then, we employed phrase chunking, i.e., extracting candidate keywords from the documents using the phrase patterns. Lastly, we conducted experiments in keywords selection using a number of scenarios and evaluate the results. The remainders of this section explain each step in more detail.

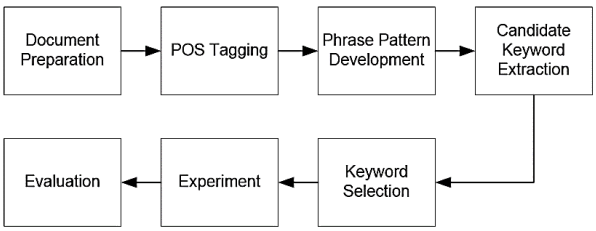

Figure 1. Methodology

2.1. Document collection and preparation

In this research, we used E-Jurnal Akuntansi from Universitas Udayana [24] as our dataset. This journal is an open-access journal in the accounting field written in Bahasa Indonesia. We randomly selected 150 publications from volume 14–2016 to volume 18–2017 of the journal as our dataset. We divided each document into three different parts, namely abstract, keywords, and content text (or the main text of the document). We used these different parts separately in this research. The abstract and content text is used as the source for the keyword extraction, while the keywords are used to generate the phrase patterns. We omitted the title, section headers, and references. After collecting the documents, several pre-processing techniques, namely non-ASCII characters removal, punctuation spacing, newlines removal, and excessive white spaces removal, were conducted. These pre-processing were performed to each part (i.e., abstract, keywords, and content text) of the dataset.

2.2. POS tagging

After document preparation, we labeled each word of each document in the dataset with its corresponding part of speech, i.e., POS tagging. The labeling process was conducted using the Indonesian POS Tagger tool and the tagset provided by Wicaksono and Purwarianti [25]. In the abstract and the content text, POS tagging was applied directly to the text. Labeling part of speech to the keywords is not trivial since keywords are unlike the abstract or the content text, which consists of sentences. Thus, POS tagging keywords was conducted by utilizing the POS sequence of the keywords found in both the abstract and the content text. For example, to label the keyword "*rasio keuangan*", we used the labeled abstract and content text and search for "*rasio keuangan*" as a phrase in the text, not just a single word of "*rasio*" or "*keuangan*". If the phrase is found in the text, we labeled the keywords phrase with the POS tags of that phrase found in the text. In the case of more than one POS sequence found, because a keyword phrase may occur more than once in the text, the keyword will be labeled with the most frequently occurred POS sequence. We used the POS tagged keywords set in the next step, phrase pattern development, to extract keywords from the tagged abstract and content text.

2.3. Phrase pattern development

Phrase pattern is defined as the POS tag sequence that forms a keyword phrase. Phrase patterns are used to obtain the candidate keywords from the text. For example, for a phrase pattern "NN JJ" (a noun followed by an adjective), we extracted all phrases in the text having the "NN JJ" tag sequence as candidate keywords. In the previous study, Shukla and Kakkar manually defined the phrase patterns [23]. In this

research, we used an automated approach by collecting the phrase patterns from all tagged keywords in the dataset, similarly resembling the knowledge domain from prior research [19]. To select the criteria for the experiment, we also collected the patterns' occurrences in the dataset as the patterns' weight. Table 1 showed examples of phrase patterns found in the dataset. For example, in Table 1, phrase pattern "NN NN" (or a noun followed by another noun) occurred 235 times in the dataset, which then becomes the weight of that pattern (i.e., phrase pattern weight).

Table 1. Some examples of extracted phrase patterns and their weight

Phrase Pattern	Phrase Pattern Weight
NN NN	235
NN NN NN	44
NN JJ	15

2.4. Phrase chunking

After having a list of phrase patterns from the dataset, we extracted candidate keywords from the abstract and the content text using phrase chunking. We collected all phrases from the POS tagged text in the dataset (tagged abstracts and content texts), having the same phrase pattern from our phrase patterns list. Figure 2 shows an example of a phrase chunking process. Using "*NN NN*" as the phrase pattern, we extracted "*kinerja/NN perusahaan/NN*", "*kondisi/NN keuangan/NN*", etc. as the candidate keywords from the tagged content text.

During the process, we found phrases with overlapping words. For example, "*kondisi keuangan*" and "*keuangan perusahaan*" are overlapped with "*kondisi keuangan perusahaan*". We kept all those phrases for candidate selection in the next step. This approach is different from the approach proposed by Shukla and Kakkar [23] because they only consider the longest overlapping words as the candidate. Using this example, Shukla and Kakkar's approach will only consider "*kondisi keuangan perusahaan*" but not "*kondisi keuangan*" and "*keuangan perusahaan*" as the candidates. We argue that our approach will be beneficial since a shorter phrase still can become a keyword. Table 2 shows several examples of candidate keywords extracted from the dataset. We recorded the occurrences of each candidate in the dataset as the candidate weight along with the phrase pattern weight found in the previous step (section 2.3.). We used these weights in the keywords selection process.

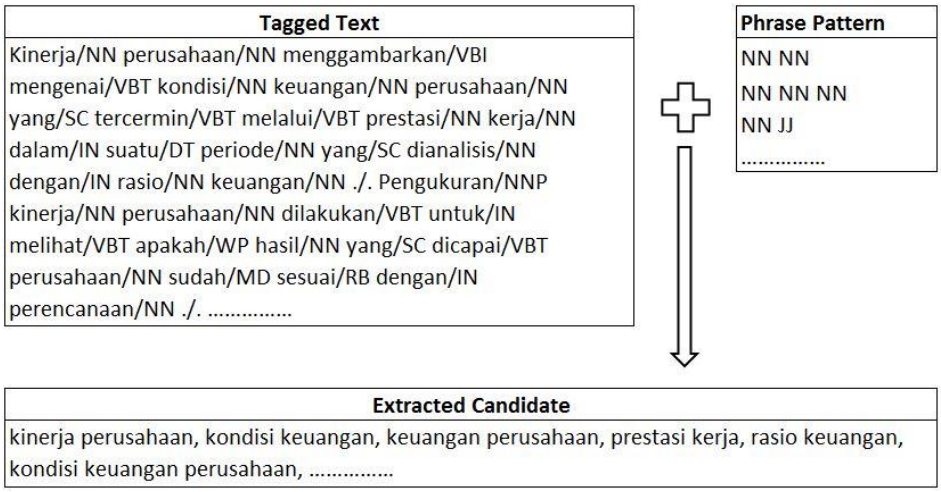

Figure 2. Illustration of the phrase chunking process

Table 2. Example of extracted candidate keywords

Candidate	Candidate Weight	Phrase Pattern	Phrase Pattern Weight
kinerja perusahaan	2	NN NN	235
kondisi keuangan	1	NN NN	235
keuangan perusahaan	1	NN NN	235
prestasi kinerja	1	NN NN	235
rasio keuangan	1	NN NN	235

2.5. Keyword extraction

Before conducting keywords selection, we removed keywords from the list of candidate keywords that occurred only once in the source text. We argue keywords that occurred only once in a document are not sufficient to represent the main idea of that document. Afterward, we sorted the remaining candidates by their candidate weight and phrase pattern weight. In keywords selection, we sorted the candidate keywords using several scenarios of multiplication between the candidate weight and phrase pattern weight of each candidate keyword (as the sorting base). The top five candidates based on the sorting base are taken as the keywords. In another case, if other candidates have the same sorting base value as in the fifth position of sorted keywords, those candidates are also taken as keywords. For example, if the sixth and seventh position candidates have the same sorting base value as the fifth position keywords, both candidates are taken as keywords as well. Another case in which there are several candidates with the same sorting base value, the sorting is conducted based on the length of the phrase: the longer phrase sorted higher than the shorter one. For example, if both candidate keywords "*kinerja keuangan perusahaan*" and "*kinerja keuangan*" have the same sorting base value, the keyword "*kinerja keuangan perusahaan*" will be positioned higher than the keyword "*kinerja keuangan*".

2.6. Experiment

In this section, we explained the scenarios that we used in the keyword extraction experiments. Our dataset consists of abstracts and content texts that have been labeled with the part of speech. The dataset was then divided into training data and testing data. The training data was used as the source to develop phrase patterns that were used to extract keywords in the testing data. We then compared the keywords generated using several keyword selection scenarios with the actual keywords from the documents in our dataset and evaluate each scenario using precision, recall, and f-measure. From 150 documents in our dataset, we use 5-fold cross-validation. Thus, we have 120 documents as the training data and 30 documents as the testing data for each fold.

In the experiment, we evaluated several scenarios of our method, which are based on 4 parameters:

- (P1) Source of the keywords extraction

In this parameter, we specified the source for the candidate keywords extraction. The values for this parameter are either the abstracts or the content texts from the dataset as the source.

- (P2) Phrase patterns

This parameter specifies the phrase patterns used in the experiments. The values defined for this parameter are short phrase patterns (i.e. phrases that consist of only 2 or 3 words) or all phrase patterns. The objective of using this parameter was to determine whether a shorter phrase pattern gives better performance over all phrase patterns, as used in the previous study [23], in generating keywords.

- (P3) Candidate keywords reduction

Candidate keywords reduction is the removal of some keywords from the candidate keywords list. A keyword is removed if there is a sub-string or super-string of the keyword that has a higher-order based on the sorting base value (see parameter P4). For example, a keyword "*rasio keuangan*" will be removed from the candidate keyword list if "*rasio*" has a higher weight, as a sorting base value, in the candidate keywords list. In this parameter, we specified whether we utilize candidate reduction or not in the experiments. We were inspired to use candidate reduction from prior research [23], such that there are no overlapping keywords.

- (P4) Candidate keywords sorting base

In this parameter, we specified sorting scenarios to select the keywords from the list of candidate keywords. We used candidate keywords' weights and phrase patterns' weight as the sorting base. The sorting scenarios of the experiments are as follows:

A: Descending sort based only on the candidate keywords weight, without considering the phrase pattern weight.
B: Descending sort based on the multiplication of the candidate weight and the phrase pattern weight.
C: Descending sort based on multiplication of the candidate weight and the sub-linear TF normalized phrase pattern weight.
D: Descending sort based on the multiplication of the candidate weight and the maximum TF normalized phrase pattern weight.

In the last two scenarios, the normalization is carried out to address the imbalance between each phrase pattern weight, as exemplified by Table 2.

All of the parameters are combined as the experiment scenarios. The top 5 of phrase patterns in each fold are shown in Table 3, meanwhile all 32 scenarios and their parameters in the experiment are shown in Table 4. Each scenario is evaluated by comparing the keywords generated with the actual keywords from the dataset and evaluate them using precision, recall, and f-measure. These metrics are commonly used to measure the performance of automatic keyword extraction [26].

3. RESULTS AND ANALYSIS

In the experiment, the phrase patterns were extracted from each fold of the training set. Table 3 shows the top five phrase patterns found in the training set as examples. From the table, one can see that "*NN NN*" occurred significantly higher (denoted by the pattern's weight) than the other patterns. To minimize the differences, a normalization technique is employed as the sorting scenario's parameters of the experiment. The experiment result for all scenarios is shown in Table 4. From this result, scenario 13 gives the best f-measure. That means the best result was obtained if we used the abstract as the source of the keyword extraction, used only short phrase patterns with candidate reduction, and sorted the candidate keywords based on their weights. In the next subsection, we discussed the performance of each parameter.

Table 3. Top 5 phrase pattern found in each fold

Fold 1		Fold 2		Fold 3		Fold 4		Fold 5	
Pattern	Weight	Pattern	Weight	Pattern	Weight	Pattern	Weight	Pattern	Weight
NN NN	235	NN NN	241	NN NN	241	NN NN	254	NN NN	249
NN	89	NN	82	NN	82	NN	69	NN	84
NN NN	44	NN NN	43	NN NN	43	NN NN	38	NN NN	39
NN		NN		NN		NN		NN	
NN JJ	15	NN JJ	21	NN JJ	21	NN JJ	16	NN JJ	16
NN NN	9	NN VBI	8	NN VBI	8	NN NN	9	NN NN	7
NN NN		NN		NN		NN NN		NN NN	

Table 4. Result of all extraction scenarios

Scenario	Parameter				Result		
	P1	P2	P3	P4	Precision	Recall	F-measure
1	Abstract	All Pattern	No	A	0.16	0.397	0.218
2	Abstract	All Pattern	No	B	0.314	0.483	0.372
3	Abstract	All Pattern	No	C	0.199	0.324	0.241
4	Abstract	All Pattern	No	D	0.221	0.354	0.267
5	Abstract	All Pattern	Yes	A	0.134	0.266	0.174
6	Abstract	All Pattern	Yes	B	0.317	0.489	0.375
7	Abstract	All Pattern	Yes	C	0.153	0.239	0.184
8	Abstract	All Pattern	Yes	D	0.184	0.282	0.219
9	Abstract	Short Pattern	No	A	0.368	0.58	0.424
10	Abstract	Short Pattern	No	B	0.411	0.528	0.446
11	Abstract	Short Pattern	No	C	0.403	0.53	0.44
12	Abstract	Short Pattern	No	D	0.398	0.52	0.434
13	Abstract	Short Pattern	Yes	A	0.458	0.584	0.492
14	Abstract	Short Pattern	Yes	B	0.416	0.526	0.447
15	Abstract	Short Pattern	Yes	C	0.415	0.526	0.447
16	Abstract	Short Pattern	Yes	D	0.414	0.527	0.446
17	Content Text	All Pattern	No	A	0.11	0.146	0.124
18	Content Text	All Pattern	No	B	0.285	0.371	0.318
19	Content Text	All Pattern	No	C	0.14	0.184	0.157
20	Content Text	All Pattern	No	D	0.179	0.234	0.2
21	Content Text	All Pattern	Yes	A	0.038	0.046	0.041
22	Content Text	All Pattern	Yes	B	0.292	0.374	0.324
23	Content Text	All Pattern	Yes	C	0.066	0.081	0.071
24	Content Text	All Pattern	Yes	D	0.111	0.142	0.123
25	Content Text	Short Pattern	No	A	0.359	0.484	0.407
26	Content Text	Short Pattern	No	B	0.341	0.443	0.38
27	Content Text	Short Pattern	No	C	0.37	0.477	0.412
28	Content Text	Short Pattern	No	D	0.372	0.479	0.414
29	Content Text	Short Pattern	Yes	A	0.36	0.473	0.403
30	Content Text	Short Pattern	Yes	B	0.344	0.446	0.382
31	Content Text	Short Pattern	Yes	C	0.366	0.468	0.405
32	Content Text	Short Pattern	Yes	D	0.37	0.475	0.411

3.1. Source of keyword extraction

Figure 3 compares the experiment using the abstract and the content text as the source of the keywords extraction. This figure shows that using abstract as the source gave a better result than using the content text. It means, although not substantial, extracting keywords from shorter and brief text like abstract is better than long text like the content text in our dataset.

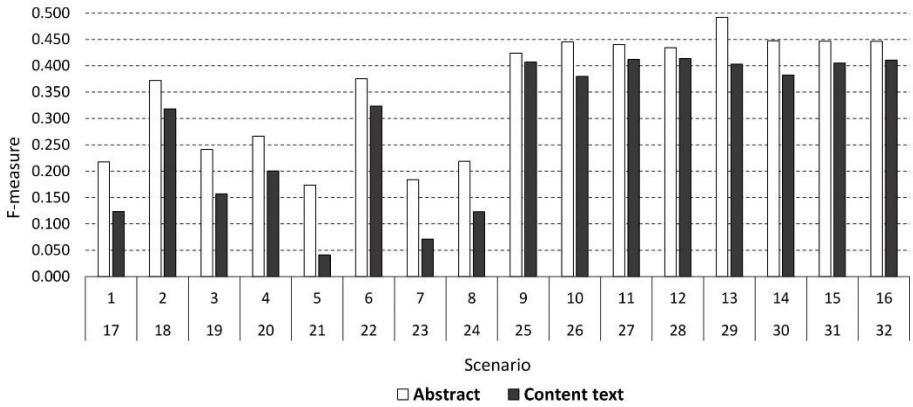

Figure 3. Comparison of the scenarios with abstract and content text as the reference

3.2. Phrase patterns

Figure 4 shows the comparison between using all phrase patterns and using only short phrase patterns. From the figure, we could see that keywords extraction using short phrase patterns gave a better result compared to using all phrase patterns. Table 4 also shows that short phrase patterns always gave better precision and recall. We observed that this is because using all phrase patterns include using single word candidates. A single word candidate occurred more than a phrase candidate such that it has a higher rank than phrase candidates if we used the number of occurrences as the base for candidate sorting. This higher rank causes single word candidates to be selected as keywords, even if they are not suitable keywords. We argued that this can be avoided if the candidate sorting is based on not only the occurrences of the patterns. The result proves the utilization of short patterns is better than using all patterns used in previous work [23].

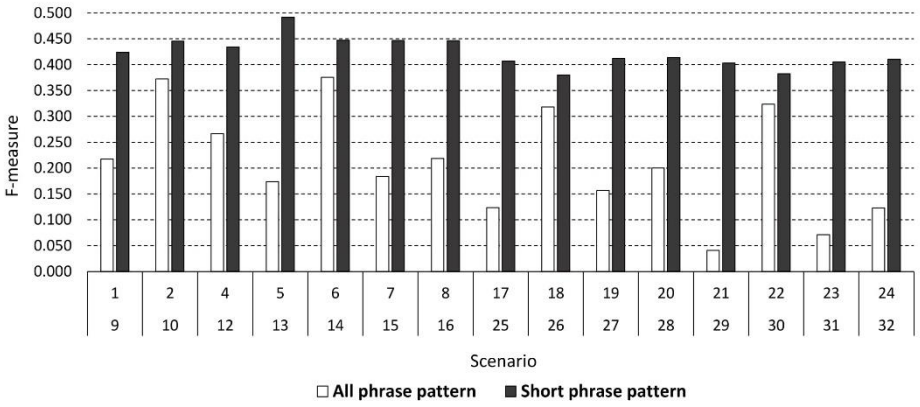

Figure 4. Comparison of the scenarios based on phrase pattern used

3.3. Candidate keyword sorting

Figure 5 shows the experiment result using different candidate keywords sortings. The figure shows that the sorting scenario B, i.e., descending sort based on the multiplication of the candidate weight and the phrase pattern weight, has the highest f-measure score. In Figure 5, there are 5 group columns of comparison: the first, second, third, fifth, and sixth columns, which have Sorting B as the best method of sorting. The first, second, fifth, and sixth group columns are extraction with all phrase patterns. As we stated before, if all phrase patterns were used, then the best sorting method is using value based on multiplication of candidate weight and phrase pattern weight. For other comparison groups (i.e., fourth, seventh, and eighth column), the best results are varied, but there were no substantial differences between them. From this experiment, normalization reduces the significant differences of phrase patterns weights of each candidate keyword as shown in Table 2. However, the normalization of the phrase pattern weights (i.e., scenario C and D) did not give substantially better results.

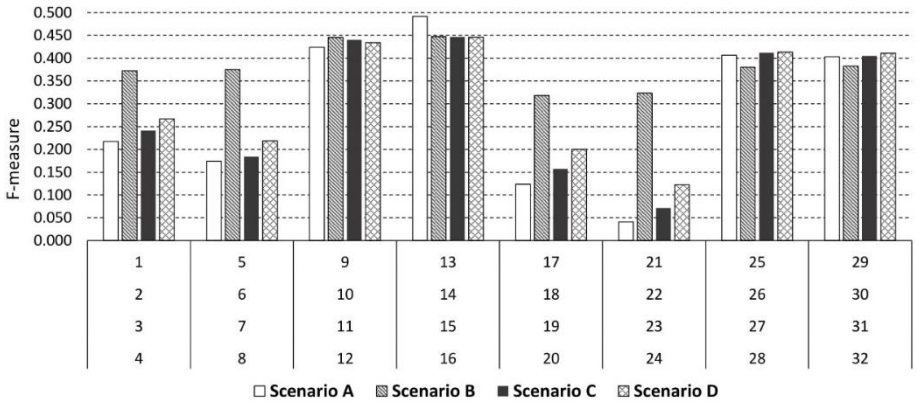

Figure 5. Comparison of the scenarios based on the sorting

3.4. Candidate keyword reduction

Candidate reduction in our proposed method did not always give a better result. Figure 6 shows only 7 out of 16 scenario comparisons where using candidate reduction provides a better result, namely, scenario 2, 9, 10, 11, 12, 18, and 26. From Table 4, we can see that scenario 2, 10, 18, and 26 are the scenarios which used Sorting B as the sorting method, while scenario 9, 10, 11, and 12 are scenarios that used only the shorter-phrase patterns from the abstract. In this case, we could see that candidate reduction gave a better result when the candidate sorting was based on the multiplication of candidate weight and phrase pattern weight without normalization, or using the shorter-phrase patterns from the abstract. The best result from this experiment was obtained using candidate reduction. This result also implicates that the utilization of candidate reduction or similarly regular expression like previous research [23] does not always improve keywords extraction results.

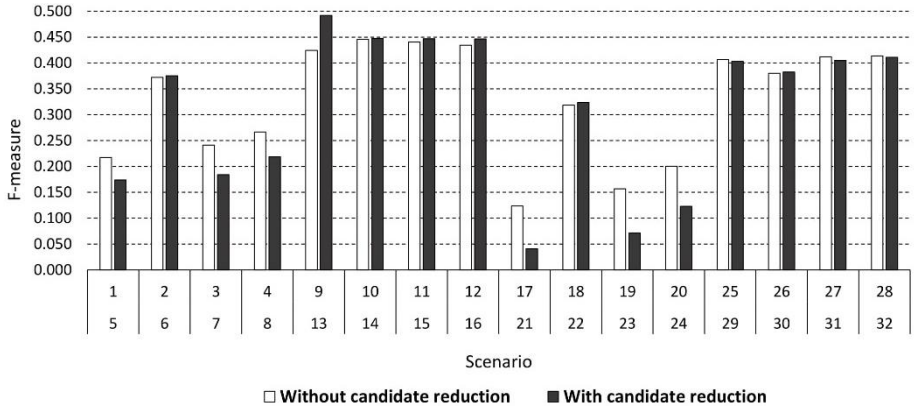

Figure 6. Comparison of the scenarios without and with the reduction

4. CONCLUSION

This research focused on automatic keywords extraction using phrase chunking for a single document written in *Bahasa Indonesia*. We used phrase patterns from actual keywords in our dataset for the phrase chunking. Then, we experimented using several scenarios defined from several parameters to generate the keywords. The best result was gained when the extraction was conducted using the abstract with shorter-phrase patterns (i.e., consists of only 2-3 words), candidate sorting based on the occurrence frequency of each candidate, and applying candidate reduction.

From this research, we concluded that using the abstract gave a better result of keyword extraction than content text. Our result also shows that the extraction using shorter-phrase patterns gave a better result than using all phrase patterns. This result is aligned with the fact that most of the keywords in papers written in Bahasa Indonesia are in phrases form. Candidate reduction in this research did not always give better results in all scenarios, except when the sorting method was based on candidate weight and phrase pattern weight without any normalization or if shorter-phrase patterns were used. Our adjustments in using generated

patterns and reducing the number of selected keywords also showed substantially better results than prior research, which used all phrase patterns and candidate reduction. Future research of keywords extraction with phrase chunking can use other selection methods for candidate selection combined with other methods of candidate weighting. Another experiment can also be carried out by combining the abstract and the content text as the source of keyword extraction algorithms.

REFERENCES

[1] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic Keyword Extraction from Individual Documents," Text Mining: Applications and Theory, John Wiley & Sons, Ltd, pp. 1–20, 2010.

[2] J. Mothe, F. Ramiandrisoa, and M. Rasolomanana, "Automatic Keyphrase Extraction using Graph-based Methods," *Proceeding of 33rd Annual ACM Symposium on Applied Computing*, pp. 728-730, 2015.

[3] N. Kamaruddin, A. W. A. Rahman, and R. A. M. Lawi, "Jobseeker-Industry Matching System using Automated Keyword Selection and Visualization Approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 3, pp. 1124-1129, 2019.

[4] S. Menaka and N. Radha, "Text Classification using Keyword Extraction Technique," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 12, 2013.

[5] T. Bohne and U. M. Borghoff, "Data Fusion: Boosting Performance in Keyword Extraction," *Proceedings of the 20th IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS)*, pp. 166–173, 2013.

[6] N. Alami, M. Meknassi, and N. Rais, "Automatic Text Summarization: Current State of the Art," *Journal of Asian Scientific Research*, vol. 5, no. 1, pp. 1-15, 2015.

[7] B. Lott, "Survey of Keyword Extraction Techniques," *UNM Education*, vol. 5, pp. 1-10, 2012.

[8] J. R. Thomas, S. K. Bharti, and K. S. Babu, "Automatic Keyword Extraction for Text Summarization in e-Newspapers," *Proceedings of the international conference on informatics and analytics*, pp. 1-8, 2018.

[9] Y. Qin, "Applying Frequency and Location Information to Keyword Extraction in Single Document," *Proceedings of the 2nd IEEE International Conference on Cloud Computing and Intelligent Systems (CCIS)*, pp. 1398–1402, 2012.

[10] S. K. Bharti and K. S. Babu, "Automatic keyword extraction for text summarization: A survey," arXiv:1704.03242, 2017.

[11] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, "An Overview of Graph-Based Keyword Extraction Methods and Approaches," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 1–20, 2015.

[12] F. C. Jonathan and O. Karnalim, "Semi-Supervised Keyphrase Extraction on Scientific Article using Fact-based Sentiment," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 4, pp. 1771–1778, 2018.

[13] C. Zhang, "Automatic Keyword Extraction from Documents using Conditional Random Fields," *Journal of Computational Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[14] A. S. Imran, L. Rahadianti, F. A. Cheikh, and S. Y. Yayilgan, "Objective Keyword Selection for Lecture Video Annotation," *Proceedings of the 5th European Workshop on Visual Information Processing (EUVIP)*, pp. 1–6, 2014.

[15] S. Siddiqi and A. Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review," *International Journal of Computer Applications*, vol. 109, no. 2, pp. 18-23, 2015.

[16] R. Mihalcea and P. Tarau, "Textrank: Bringing Order into Text," *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[17] X. Wan and J. Xiao, "Single Document Keyphrase Extraction Using Neighborhood Knowledge," *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, vol. 8, pp. 855–860, 2008.

[18] K. S. Hasan and V. Ng, "Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art," *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 365–373, 2010.

[19] G. Li and H. Wang, "Improved Automatic Keyword Extraction based on TextRank using Domain Knowledge," *Natural Language Processing and Chinese Computing, Springer*, pp. 403–413, 2014.

[20] S. Lahiri, S. R. Choudhury, and C. Caragea, "Keyword and Keyphrase Extraction using Centrality Measures on Collocation Networks," arXiv:1401.6571, 2014.

[21] R. Bhowmik, "Keyword Extraction from Abstracts and Titles," *Proceedings of the 2008 IEEE Southeastcon*, pp. 610–617, 2008.

[22] Y. Lu, R. Li, K. Wen, and Z. Lu, "Automatic Keyword Extraction for Scientific Literatures using References," *Proceedings of the 2014 IEEE International Conference on Innovative Design and Manufacturing (ICIDM)*, pp. 78–81, 2014.

[23] H. Shukla and M. Kakkar, "Keyword Extraction from Educational Video Transcripts using NLP Techniques," *Proceedings of the 6th IEEE International Conference on Cloud System and Big Data Engineering (Confluence)*, pp. 105–108, 2016.

[24] Universitas Udayana, "E-Jurnal Akuntansi," Jan. 01, 2012. Accessed on: May 05, 2017. [Online]. Available: http://ojs.unud.ac.id/index.php/Akuntansi

[25] A. F. Wicaksono and A. Purwarianti, "HMM Based Part-of-speech Tagger for Bahasa Indonesia," *Proceedings of the 4th International MALINDO Workshop*, 2010.

[26] Z. Nasar, S. W. Jaffry, and M. K. Malik, "Textual Keyword Extraction and Summarization: State-of-the-Art," *Information Processing & Management*, vol. 56, no. 6, 2019.