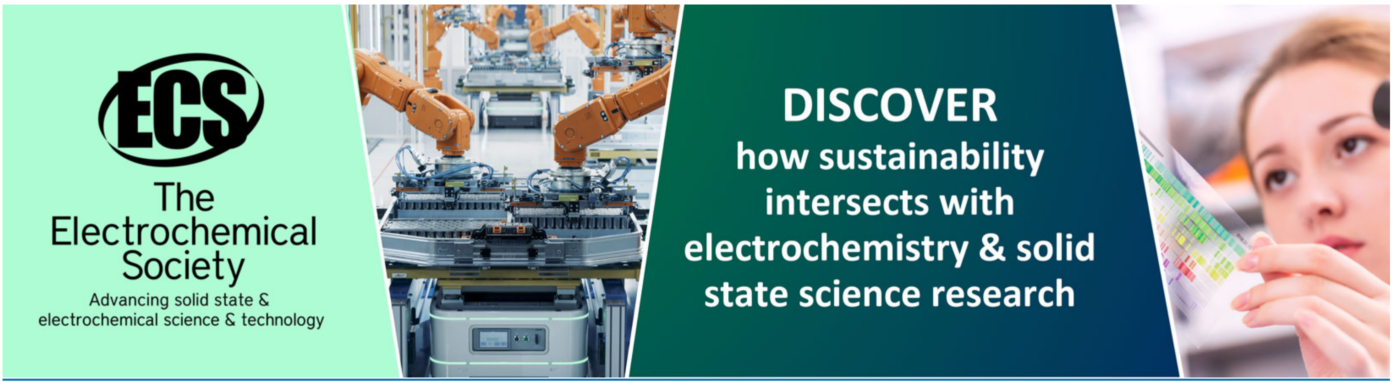
Journal of Physics: Conference
Series

PAPER • OPEN ACCESS

IndoAcro: An Indonesian Acronym and Expansion Repository with Data Auto-Update Implementation

To cite this article: T F Abidin *et al* 2020 *J. Phys.: Conf. Ser.* **1566** 012100


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

IndoAcro: An Indonesian Acronym and Expansion Repository with Data Auto-Update Implementation

T F Abidin[1], R Ferdhiana[2], M Iqbal[1], D Syaputra[1], T W A Putera[1], M Z Aksana[1]

[1]Department of Informatics, Universitas Syiah Kuala, Banda Aceh, Indonesia
[2]Department of Statistics, Universitas Syiah Kuala, Banda Aceh, Indonesia

E-mail: taufik.abidin@unsyiah.ac.id

Abstract. IndoAcro is an Indonesian acronym and expansion repository created using machine learning and big data technology. The repository can be publicly accessed from www.indoacro.cs.unsyiah.ac.id. Six important steps of IndoAcro have been developed and implemented, which consists of (1) data crawling, (2) data cleaning, (3) generating candidate pairs of acronym and expansion, (4) generating numerical features, (5) classifying the candidate pairs, and (6) filtering the classification results. In this study, we introduce and analyze the implementation of data auto-update for IndoAcro. Since it was developed, IndoAcro has 2,232 pairs of acronym and expansion, collected from more than 50 thousand online news articles. Because no auto-update approach has been implemented previously, the number of acronym and expansion pairs in the database is monotonous, dull, and static. In this study, we introduce and analyze the implementation of data auto-update for IndoAcro. We have analyzed and evaluated the data auto-update process for 180 days, each process consists of 2 days interval. We found that the data auto-update approach has successfully implemented and updated the data for IndoAcro. We collected 1,639 pairs of acronym and expansion in the first run, 343 and 224 pairs in the second and third runs.

1. Introduction

Acronyms are abbreviated forms of phrase used to shorten entities' long forms [1]. Acronyms can be formed from a combination of all uppercase letters, a combination of uppercase and lowercase letters, and a sequence of speech sounds [2]. In a paragraph, when an acronym is firstly mentioned, its meaning or expansion is usually written on the left or right side of the acronym [3]. Therefore, automatically recognizing the correct pairs of acronym and expansion by computers from a large set of data texts is not an easy task, especially when applied to specific domains like biomedical texts [1] and Wikipedia [4].

In the last decades, many prominent approaches have been introduced to recognize pairs of acronym and expansion, such as a supervised learning with SVM [5], recognizing acronyms and expansions in specific languages [6, 7], and specific domains [1, 4] to name a few. In this study, we introduce and analyze the implementation of data auto-update for IndoAcro, an Indonesian acronym and expansion repository that is publicly accessible from www.indoacro.cs.unsyiah.ac.id as depicted in Figure 1. There are six important steps of IndoAcro which have been developed and evaluated in our previous study [8]. It consists of (1) data crawling, (2) data cleaning, (3) generating pairs of acronym and expansion, (4) generating numerical features, (5) classifying pairs of acronym and expansion, and (6) filtering the results. However, no data auto-update has been implemented. As a result, the number of acronym and expansion pairs in the IndoAcro database is monotonous, dull, and static. The auto-update diagram proposed for IndoAcro is illustrated in Figure 2.

Since it was developed, IndoAcro has 2,232 pairs of acronym and expansion in its database, collected from more than 50 thousand online news articles. Table 1 summarizes the number of acronym and expansion pairs for each alphabet. Through the implementation of data auto-update, the number of acronym and expansion pairs in the database can be updated regularly. Hence, the main contribution of this study is to introduce the data auto-update approach for IndoAcro that can periodically and automatically carry out the six important steps of IndoAcro and update the database. Our contributions are:

(1) We record all URLs that have been downloaded and analyzed in the database, and then, only download news articles that have never been processed before.

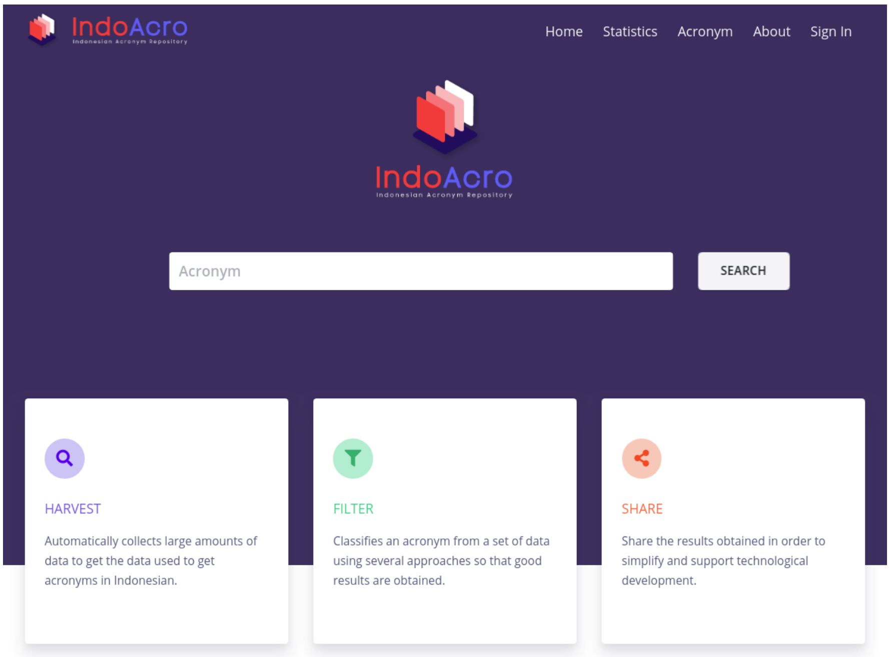
(2) We manage efficiently the process of determining candidate pairs of acronym and expansion, generating features, classifying, and filtering the results and record their elapsed time.

(3) We score the correct pairs of acronym and expansion and insert them into the database when URL supports meet the threshold.

The rest of the paper is organized in several sections. In Section 2, we discuss the methodology and in Section 3, we present the results and discussion. Finally, in Section 4, we conclude the work and state our future plans.

2. Methodology

IndoAcro automation consists of six main stages: (1) data crawling, (2) data cleaning, (3) generating pairs of acronym and expansion, (4) generating numerical features, (5) classifying pairs of acronym and expansion, and (6) filtering the results to meet a given threshold [8]. Each



Automatically collects large amounts of data to get the data used to get acronyms in Indonesian.

Classifies an acronym from a set of data using several approaches so that good results are obtained.

Share the results obtained in order to simplify and support technological development.

Figure 1. IndoAcro repository main page at www.cs.indoacro.unsyiah.ac.id

stage is executed with two days time interval. We created a monitoring table to record the information of each stage. The table has the following attributes:

(1) id (int) is the primary key of each date interval.

(2) date_range (varchar) is the attribute that holds the date range of data execution in the format of start date to end date, for example, 20190101-20190102.

(3) crawl_time (timestamp) is the field that records the crawling time for a particular date range (default value: current timestamp).

(4) is_clean (tinyint) is the field that holds true (1) or false (0) that indicates whether the crawled articles have been cleaned and no failures occurred during the cleaning process (default value: 0).

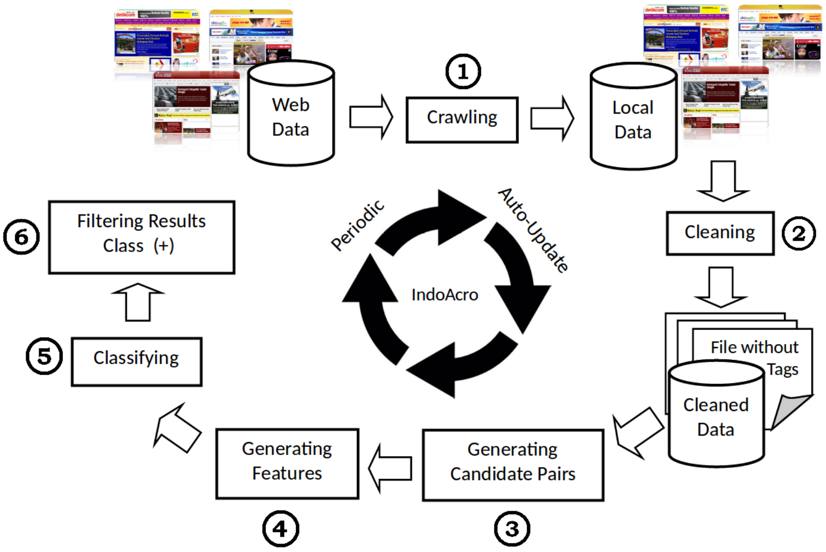


Figure 2. IndoAcro auto-update stages

Table 1. The number of acronym and expansion pairs for each alphabet

Alphabet	Total	Alphabet	Total
A	135	N	37
B	191	O	25
C	49	P	351
D	141	Q	4
E	16	R	92
F	67	S	161
G	53	T	92
H	57	U	70
I	121	V	7
J	62	W	55
K	244	X	0
L	68	Y	5
M	127	Z	2

(5) clean_time (timestamp) is the attribute that holds the completion time of the cleaning process for a certain date range (default value is null).

(6) is_candidate_generated (tinyint) is the field that holds true (1) or false (0) that indicates whether the cleaned files have been processed or not, the candidate pairs of acronym and expansion have been generated, and no failures occurred during the process (default value is 0).

(7) generate_candidate_time (timestamp) is the attribute that records the completion time of the candidate generation process for a certain date range (default value is null).

(8) is_feature_generated (tinyint) is the attribute that holds true (1) or false (0) that indicates whether the candidate pairs have been processed or not and there were no failures occurred during the feature generation stage (default value is 0).

(9) generate_feature_time (timestamp) is the attribute that records the completion time of the feature generation process for a certain date range (default value is null).

(10) is_feature_classified (tinyint) is the attribute that holds true (1) or false (0) that indicates whether candidate pair of acronym and expansion with numerical features attached have been classified and there no failures occurred during the classification (default value is 0).

(11) feature_classification_time (timestamp) is the field that holds the completion time of the classification process for a certain date range (default value is null).

(12) is_final_expansion_extracted (tinyint) is the attribute that holds true (1) or false (0) that indicates whether the final classification file has been filtered to get the correct pairs of acronym and expansion or not (default value is 0).

(13) final_expansion_extraction_time (timestamp) is the attribute that holds the completion time of the filtering process for a certain date range (default value is null).

Crawling is the initial process of IndoAcro. During the crawling stage, news articles were collected from eight Indonesian news portals, namely Viva, Detik, Liputan6, Kompas, Sindo News, Tribune, and JPNN. The process begins by determining the crawling date interval, for example, the date interval starts from January 10 to 11, 2018, and therefore, the date interval is formatted as 20180110-20180111. After the date interval is obtained, a *data_crawl_20180110-20180111* folder will be created and the date interval is inserted into the monitoring table in the IndoAcro database.

Each URL of an article is then double-checked with the records in the *url_acronym* table in the IndoAcro database to know if the article has been downloaded before or not. If the URL is not found in the table, then the URL is inserted into the queue for further processing. After all URLs from one news portals have been extracted and checked, the URLs in the queue are downloaded and stored in the crawling folder. The URL is also inserted into the *url_acronym* table so that the same URL will not be downloaded again later. The process is repeated until all dates are processed. The crawling ends by writing the crawling logs into the times-logs.txt file which stores the start time, end time, total time in hours, minutes, seconds, and the number of articles successfully downloaded.

After the crawling process is completed, the articles are further cleaned by removing the HTML tags and special symbols in the articles. The process begins by checking the record with the ID equal to the date interval in the monitoring table that has *is_clean value* equal to 0. The value indicates that the articles have not previously been cleaned. If there is no record with date interval key has the value *is_clean* equal to 0, then the process will stop. However, if there is at least one record with date interval key has *is_clean* equal to 0, then the process continues and the files in the *data_crawl* folder are cleaned. All cleaned files are stored in *data_clean* folder. We used HTML::ExtractContent Perl module specifically designed to extract web content[9] and heuristically scored the blocks of HTML based on the number of punctuation marks and the lengths of non-tag texts in the paragraph [10].

After all crawled articles are cleaned, the monitoring table is updated by setting the field *is_clean value* to true, field *clean_time* to *current_timestamp*, and writing the logs into the times-logs.txt file that contains the start time, end time, total time in hours, minutes, seconds, and the number of articles that were successfully cleaned.

Generating pairs of acronym and expansion is started after the cleaning process is completed. This process starts by checking the date interval key in the monitoring table that has value *is_candidate_generated* equal to 0. If at least one date interval key in the monitoring table has *is_candidate_generated* equal to 0, the files are then read from *data_clean* folder and the candidate pairs of acronym and expansion will be generated. The candidate pairs are written into a file with the same name as the date range key and placed in the *data_candidate* folder.

The classification is done using SVM-Light model [11] and started by checking the date interval key in the monitoring table and the value of *is_feature_classified* must be equal to 0. Files are then read from folder *data_candidate_f8* which contains pairs of acronym and expansion and their numerical features. After the classification results are obtained, the prediction results are written to the prediction file, stored in the temporary folder. The prediction results are further examined. Only the results with a predictive value greater than 0 (positive) will be collected. The process ends by updating field *is_feature_classified* to 1, field *feature_classification_time* to the value of *the current_timestamp*, and writing logs into times-logs.txt file which includes the start time, the end time, total time in hours, minutes, seconds, and the number of results with positive predictive values.

Finally, the filtering process begins by checking the date interval in the monitoring table that has *is_final_expansion_extracted* equal to 0. When finished, the correct pairs of acronym and expansion are written into the file with the same name as the date interval in the following format:

$$[acronym]+ : [expansion]+ : [8features]$$

for pairs of acronym and expansion that are considered as acronyms of type syllable, and

$$[acronym] :: [expansion] :: [8features]$$

for pairs of acronym and expansion that are considered as acronyms with a combination of uppercase letters.

The filtering process ends by updating field *is_final_expansion_extracted* to 1 and field *final_expansion_extraction_time* to *the current_timestamp*. The log of this stage is stored in times-logs.txt file that stores the start time, end time, total time in hours, minutes, seconds, and the number of correct pairs of acronym and expansion obtained.

3. Results and Discussion

We show and discuss the results of data auto-update implementation for IndoAcro in this section. The auto-update has been implemented to process news articles published in news portals such as Viva, Detik, Liputan6, Kompas, Sindo News, Tribune, and JPNN as shown in Figure 3. The patterns show that all processes have a stationary time series which means that the statistical properties are all constant over time. The crawling process consumes a lot of time when compared to the other processes as shown in Figure 4. On average, the crawling process requires 6,860.78 seconds to crawl the data in two days period. The second most time consuming is the generating features process. On average, it needs 2,581.54 seconds to complete. Moreover, the time of the generating features process is around 1/3 the time of crawling.

Figure 5 shows the boxplot for the other four processes. It is clear that the classifying process is the least time consuming when compared to the others, even though it is not significantly

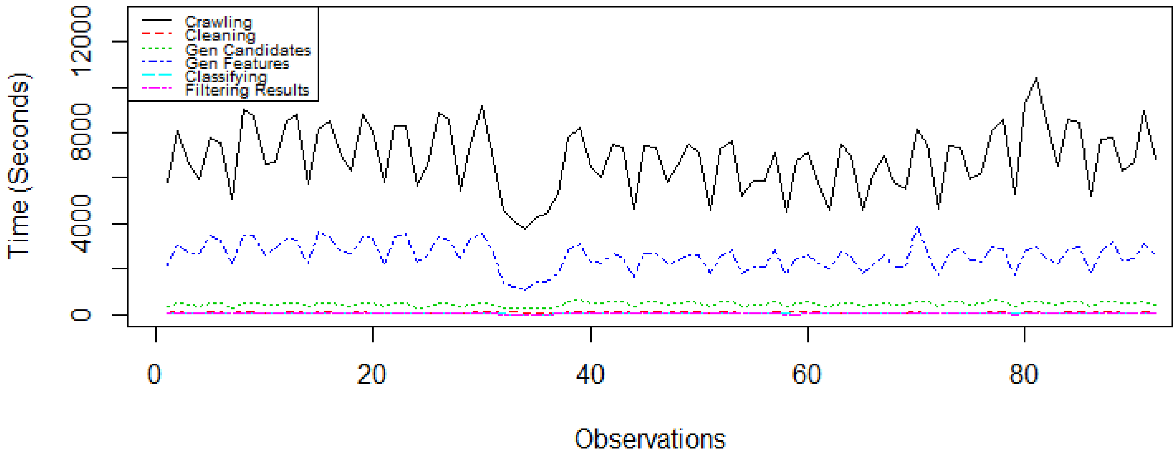


Figure 3. Time pattern of IndoAcro automation time for the six processes

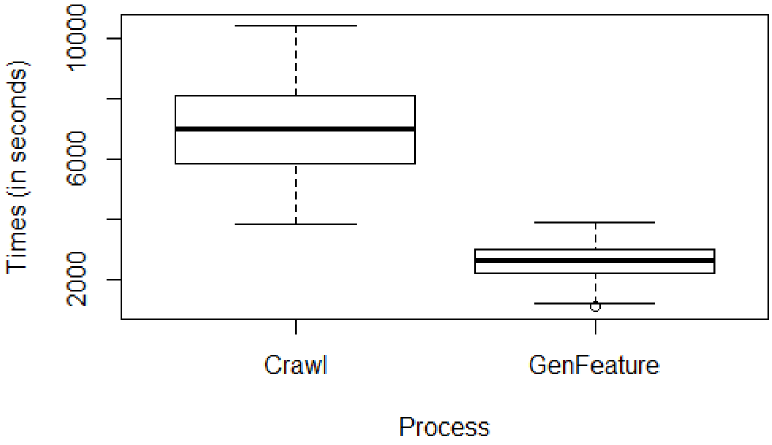


Figure 4. Box plot of crawling and generating feature

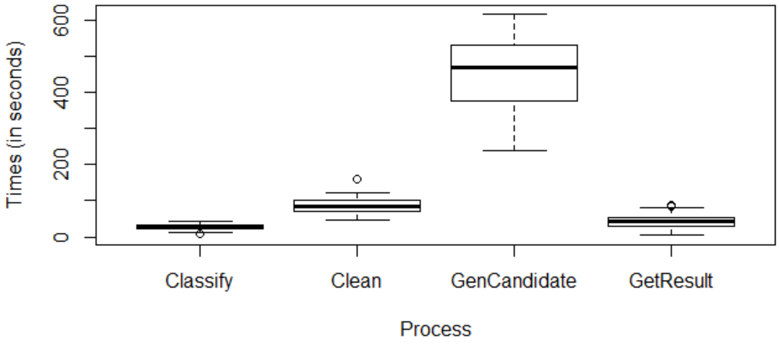


Figure 5. Box plot of the other four processes

different from the time of the cleaning and get result processes. In addition, among those four processes, the generating candidate process is the most time consuming, although it is not as much as the crawling and generating feature processes. The maximum time of the generating candidate process is around 600 seconds, but it is faster than the minimum time of the generating feature process which is approximately 1,050 seconds.

The total new URLs obtained during the six month observations are 25,975 with 2,206 unique acronym and expansion pairs have supporting URLs above the given threshold (20 URLs). There are 28,992 acronym and expansion pairs with supporting URLs below the threshold. The acronym and expansion pairs are inserted into the IndoAcro database, however, they are not active until their supporting URLs reach the threshold. Table 2 summarizes the finding of each two-month executions (batch) and Table 3 lists the additional acronyms for each alphabet. These results have confirmed that the auto-update approach for IndoAcro has been successfully implemented to auto-update the IndoAcro database. Examples of new acronyms found during the six month executions are:

Table 2. The number of acronym and expansion pairs found in each batch

Batch	Acronym below Threshold	Acronym above Threshold	Total URLs
One	11,899	1,639	11,240
Two	8,106	343	7,075
Three	8,987	224	7,660
Total	28,992	2,206	25,975

- GPS stands for *Global Positioning System*
- IDC stands for *International Data Corporation*
- KIPP stands for *Komite Independen Pemantau Pemilu* (Independent Election Monitoring Committee)
- APTRI stands for *Asosiasi Petani Tebu Rakyat Indonesia* (Indonesian Sugarcane Farmers Association)
- Jubir the abbreviation of *Juru Bicara* (Spokesman)
- Monas, the abbreviation of *Monumen Nasional* (National Monument)
- Jamsos, the abbreviation of *Jaminan Sosial* (Social Security)
- Paspampres, the abbreviation of *Pasukan Pengamanan Presiden* (President's Security Forces)
- Kapuspenkum, the abbreviation of *Kepala Pusat Penerangan Hukum* (Head of the Legal Information Center)

4. Conclusion

We have developed and evaluated the data auto-update for IndoAcro. The results show that the automation was running well on the six stages of IndoAcro. Among those six stages, crawling and generating features are the two stages that consume a lot of time. For six month observations, 2 days period for each observation, we found that on average it takes 6,860.78 seconds to crawl and 2,581.54 seconds on average to generate the eight numerical features. We also discovered that in the first batch, 1,639 pairs of acronym and expansion were found and met the threshold of supporting URLs whereas, in the second and third runs, we found 343 and 224 pairs of acronym and expansion respectively. We will prune and optimize the process of generating candidates and numerical features in our future work.

Table 3. The number of acronym and expansion pairs for each alphabet added for each batch

Alphabet	Baseline	Batch 1	Batch 2	Batch 3
A	135	88	17	11
B	191	144	25	12
C	49	58	15	14
D	141	80	23	9
E	16	14	1	3
F	67	38	11	10
G	53	39	8	8
H	57	37	10	5
I	121	96	24	9
J	62	47	11	5
K	244	147	27	20
L	68	62	12	7
M	127	67	19	7
N	37	33	4	5
O	25	24	5	0
P	351	247	50	38
Q	4	0	0	1
R	92	87	19	13
S	161	118	24	18
T	92	90	18	9
U	70	64	8	13
V	7	5	0	1
W	55	49	12	6
X	0	0	0	0
Y	5	3	0	0
Z	2	2	0	0
Total	2,232	1,639	343	224

4.1. Acknowledgments

This study was supported by Universitas Syiah Kuala through Lektor Kepala Research Grant 521/UN11/SPK/PNBP/2019. We would like to thank the members of Data Mining and IR Research Group, Department of Informatics for their valuable thought and discussion.

References

[1] Sanchez D and Isern D 2011 *Journal of Applied Intelligence* **34** 2 p 311-327.
[2] Senthilkumar R M and Jayanthi V E 2018 *Proc. of the 2nd International Conference on SCI* p 121-133.
[3] Park Y and Byrd R J 2001 *Proc. of Conference on Empirical Methods in NLP* p 126-133.
[4] Choi D and Kim P 2015 *Software: Practice and Experience* **45** p 1073-1086.
[5] Xu J and Huang Y 2007 *Soft Computing* **11** 4 p. 369-373.
[6] Jacobs K, Itai A and Wintner S 2018 *Annals of Mathematics and Artificial Intelligence.*
[7] Wahyudi J and Abidin T F 2011 *Prosiding SNATIKA* p 115-119.
[8] Abidin TF, Adriman R and Ferdhiana R 2018 *Proc. of the 3rd Int. Conf. on IT, IS, and EE* p 189-193.
[9] Abidin TF, Hasanuddin M and Mutiawani V 2017 *Proc. of the Int. Conf. on EE and Informatics* p 324-327.
[10] CPAN 2019 cpansearch.perl.org/src/TARAO/HTML-ExtractContent-0.12/README.md.
[11] T. Joachims 1999 *Making large-scale SVM learning practical* Advances in kernel methods - support vector learning B Scholkopf, C Burges and A Smola ed (Massachusetts: MIT Press).