Building Automatic Mind Map Generator for Natural Disaster News in Bahasa Indonesia

Ranu Yulianto
Sekolah Tinggi Ilmu Statistik
rn.yulianto@gmail.com

Siti Mariyah
The Center for Computational Statistics Study,
Sekolah Tinggi Ilmu Statistik
sitimariyah@stis.ac.id

***Abstract*— the use of the internet today cannot be separated from people's lives. The available information is getting bigger and easier to obtain. Such information can be found in blog articles, news sites, and even statuses on social media. However, the available information sometimes cannot be utilized properly due to lack of a better understanding of the obtained information itself. The purpose of this research is to develop an automatic mind map generator application that can create a mind map from the input of news articles automatically. This is expected to help users understand the contents of the article. The study will take a case study of natural disaster news articles. In its application, researchers used the Support Vector Machine (SVM) method of multi-label classifier one vs rest with linear kernel to classify sentences in the news into the 5W+1H class (what, when, where, who, why, how). Beside classification, this research also includes summarization task as a preliminary task. To get the accurate model, we conducted some experimental study by examining combination of filtering features and candidate features. Our classification model raises F1-scores of 75%. The classification maps the word or phrase into each class, each class is determined as each node in mind map visualization with the root node is the image which shows the title of news article. The usability of our application was evaluated using The System Usability Scale and got the score of 78.5. This mind map generator also provides model evaluation by users, each user can review the classification result and if they agree with the result, they can update the model. By this scheme, the accuracy of our model is getting more accurate and lets us grab new data set automatically.**

Keywords—mind map; summarization; news mining; SVM; multi-label classification.

I. Introduction

The use of internet has been inherent in daily activities. A large amount of information is easily accessed and retrieved using the internet. Information is also available in the blog article, news channel, or in social media and commonly in a passage format. However, the information occasionally cannot be harnessed well since the lack of understanding of the passage itself. Due to the time excuse, people just skimmed through the passage instead notice or highlight the main idea or information from the given passage. Based on [1], the reader should build knowledge prior to reading for getting the comprehensive understanding of passage while this effort takes time.

Many techniques have been developed to ease the reader for getting the comprehensive understanding of a passage quickly, e.g. info graphics and mind map. Mind map was first developed by Tony Buzan in 1960 [4]. Mind map is an easy way to brainstorm thoughts, main idea, or information from a passage without worrying about order and structure. Simply, a mind map is made by making a summary of passage then displayed by drawing it on a piece of paper or other media. However, this way is not effective if the mind map is digitally distributed. Nowadays, there are many tools which can help the mind map creation and distribute the resulted mind map digitally but most help designing side only. The process of reading to understand the content of passage and the relationship between entities stated in passage is still required to make the mind map using these tools. It does not work effectively because it charges the users to read passage wholly. Therefore, it needs the mind map maker (tool) which can make mind map automatically without forcing the user to read the passage before.

This research built a tool called Automatic Mind Map Generator. It is a tool to understand the content of a passage, to extract the main ideas or information from passage, and to make a mind map automatically. The input of this tool is passage(s) and the output is mind map visualization. Most research has been already done to automate mind map creation where the input text is in English [1] [2] while our input text is in Bahasa Indonesia. Building this automatic mind map generator needs accurate classification model that is used to extract main ideas from input. The model has to be able classifying sentences in passage whether it represents information or does not. The information means that information can explain *what, when, where, why, who,* or *how* (5W+1H) aspect. We limited input text to natural disaster news only. We presumed 5W+1H aspect is fully explained over the disaster news passage.

Due to the difference characteristics of English and Bahasa Indonesia, we built a suitable text processing pipeline. Text preprocessing pipeline is applied to construct corpus which is used later in feature selection and feature extraction stages. We created some feature combinations and applied the best feature combination in order to produce accurate classification model. The extracted features are used to train the algorithm SVM one vs rest. Classification model learns characteristics owned by every sentence and decide which sentences that can explain

978-1-5386-3100-3/17/$31.00 ©2017 IEEE

5W+1H aspect. Our experiments show that a sentence can tell more than one important aspects, e.g. a sentence can tell both what and where a natural disaster happened. The chosen sentences are analyzed using named entity recognition to get the word or phrase in the sentence that describes an entity. The entities resulted are displayed in mind map visualization where the center of visualization is the image of disaster. Our tool is successful to process the news and to display the mind map of it with F-1 scores over 70% and processing time about 30 seconds. The remainder of paper is organized as follows: section 2 shows the literature review and section 3 describes the research problems and proposed solutions. Section 4 explains research methodology. Experimental study is described in section 5 and conclusion is delivered in section 6.

II. LITERATURE REVIEW

Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things [15]. It is intended to facilitate communication between human languages to computer so computer can understand what human languages mean. Example studies in this field are summarization and classification which used together to make long text more understandable for the reader. There are studies about Automatic Mind Map Generator in the past that can help the reader easily understand the content of the text.

Mind map can help us in many areas. It can represent a document summary; can help filter search results in a better way; speed up research; and help to get more relevant information. In the cognitive area, there is research where the use of mind map can help us in fostering trust, help creating conceptual design to fully develop designer's potential [17].

Several studies related to this research include the research of Mohamed Elhoseiny and Ahmed Elgammal [4]. In this research, mind map generator is developed using Multi Level Meaning Representation algorithm which is the development of Detailed Meaning Representation Algorithm. By this algorithm, visual representation of text input is grouped by theme and transformed into mind map. However, their research is suitable to be applied to biographical texts, which theme grouping among personal life side, career achievement, and politics. The difference with this research lies in grouping the nodes of the mind map. Their study grouped the text based on themes surrounding one's life (biography) whereas our research grouped the news based on the 5W+1H aspect.

This research includes summarization which is part of natural language processing task. There are some references in summarization topic that we used in this research. The first one is the research of Niladri Chatterjee and Shiwali Mohan [18] about extraction-based summarization for single document. In this research they use semantic similarity between sentences to remove the redundancy from the text. The next is Aristoteles, et.al. [3] who studied highest weighting in summarization case. Method used in this research to search for the weight of the features is genetic algorithm. The features that used in this research are sentences position (F1), positive keywords in sentence (F2), negative keywords in sentence (F3), sentence centrality (F4), sentence resemblance to the title (F5), sentence inclusion of name entity (F6), sentence inclusion of numerical data (F7), sentence relative length (F8), bushy path of the node (F9), summation of similarities for each node (F10), and latent semantic feature (F11).

In the field of text visualization itself, there are almost similar research and technology, including Wordle from IBM which is a tool for web-based text visualization. Wordle will produce a set of words that resemble clouds with attention to aspects of typography, color, and decomposition. Each of these aspects provides a visual effect that enables users to know which words those frequently appear in the source text. Although it looks similar, Wordle only provides visual information about which words those frequently appear, but not to the meaning and purpose of the text.

III. PROBLEMS DEFINITION AND PROPOSED SOLUTIONS

Some problems are found in this research. The first problem is how to preprocess Bahasa Indonesia text with the purpose of readiness to mine. The second is how to select best features that can first-rate sentence or words representing main ideas or information from the passage. The last is how to validate the output of this application and make this app more accurate to visualize the mind map of new disaster news.

English and Bahasa Indonesia have some differences and it affect to different approaches are required to apply. Stemming is a technique which offers a mapping for different morphological variants of words into their base word. Most text mining tasks [2][9][10][11][12] include stemming in text preprocessing stage since terms which have a common stem will usually have similar meaning and it can improve the task performance [5][7][8]. However, there are many problems faced by stemming process in Bahasa Indonesia. The problems comprise [5]:

a. Imbuhan of Bahasa Indonesia is quite complex which consist of prefix, suffix, conflix, infix, imbuhan from foreign language, and the rule of prefix alteration;
b. The word sense ambiguity that means one word has double meaning and comes from different base word, e.g. “beruang” and “berikan”;
c. Over stemming, e.g. “berikan” is stemmed to be “ber-i-kan” and under stemming, e.g. “mengecek” is stemmed to be “meng-ecek”
d. The dictionary (a collection of base words) dependency;
e. The inconsistency of stemming manually;
f. The plural form, e.g “buku-buku”;
g. Acronyms and proper noun, that should not be stemmed.

To handle these problems, we applied stemming algorithm based on Nazief and Adriani algorithm [14], Enhanced Confix Stripping Algorithm [6], and Extended Confix Stripping Algorithm [13]. Applying these algorithms, we can avoid overstemming with basic words, avoid understemming with additional rules, and stem plural words.

The passage can have large feature sets because every token (word) can act as a feature. Feature extraction transforms every token into numeric represented in a matrix. The algorithm should learn and deal with big matrix which contains feature values from whole training documents. Features such as bag of word, term frequency (tf), term frequency inverse document frequency (TF-IDF) and n-grams are used to use in last research. We arrange some feature combinations and conduct feature selection to select the best combination. To avoid the sparsity matrix that would be produced, we conducted two stages. First, we chose the top 50% of sentences from each news article (document). The ranking is measured from the value of positive keyword (F2), sentence centrality (F4) and sentence resemblance to the title (F5). F2, F4, and F5 are the features with highest weighting in summarization case which are studied very well by Aristoteles, et.al. [3]. Then, from the top 50% of sentences, we examine the feature combinations in order to understand which combination results the highest F1-score.

The last problem is output validation. We realize that a good application must qualify. Therefore, we did not only conducting many text mining related experiments but also the white-box testing, black-box testing, and the system usability scale test. Besides that, we provide an application feature that facilitate user to verify the result of text classification (5W+1H classification) and to update the model in order to grab the new dataset automatically.

IV. Methodology

Our experiments combined summarization and classification to form the node in mind map. First, we summarized the passage and filtered the sentences using feature proposed by Aristoteles, et.al. [3]. We adopted the weights of features in summarization and used features with three biggest weights. Then we predicted summarized passage into 5W+1H classes and place it into corresponding node in mind map.

Preprocessing and Corpus Building

First we build a corpus using 76 news articles that we got from several Indonesian news channels, such as detik.com, kompas.com, liputan6.com, etc. From 76 articles we got 1121 sentences that we manually classified into zero or more classes in why, what, where, when, who, how class. Based on our observation to the sentences in the passage, we should apply multi-label classification because one sentence can be a part of one or more class. Manual classification was conducted for creating the labeled corpus. Then we pre-processed the labeled corpus to get a ready mined corpus. Preprocessing included case folding, tokenization, stopword removal and stemming. We used Sastrawi package for stemming process that apply all algorithms which we have already explained in Section III. The preprocessing step is necessary to prepare and produce the input for the next step.

Text Transformation (Feature Extraction)

At this stage, we transformed the labeled dataset into feature matrixes. A matrix contains the collection of feature values owned by dataset. We extracted bag of word (BOW) and term frequency-inverse document frequency (TF-IDF) features combined with filtering features which consist of positive keyword feature / the frequency of keyword appear in corpus (F2), sentence centrality feature / frequency of keyword that appear in document (F4) and sentence resemblance to the title feature (F5). The matrixes will then be used in feature selection and modeling.

Feature Selection

We set F2, F4, and F5 as filtering features that filter the sentence based on the value owned by sentence itself. The higher value represents the more important the sentence. It means that the sentence is considered to be able explaining the 5W+1H aspect. We set F2, F4, F5, BOW, and TF-IDF as candidate features.

We used a filter to select 50% of the sentences with the highest value in the passage as a form of summarization. We extract the feature values from these selected sentences and used it in the next step. We split our dataset to be data training and testing with ratio 7:3. We test our model using testing data and evaluate the precision, recall and F1-score. We selected the best model from the combination of filter features and candidate features then we used it for the prediction. Complete combination of filter features and candidate features that used in this stage can be seen as follows:

Table 1. Complete combination of filter and candidate features

Num.	Filter	Feature Combination
(1)	**(2)**	**(3)**
1	F_2	BOW
2		TF-IDF
3		BOW + (F_2 + F_4 + F_5)
4		TF-IDF + (F_2 + F_4 + F_5)
5		BOW + TF-IDF
6		BOW + TF-IDF + (F_2 + F_4 + F_5)
7		BOW + (F_2 + F_4 + F_5) with weight
8		TF-IDF + (F_2 + F_4 + F_5) with weight
9		BOW + TF-IDF + (F_2 + F_4 + F_5) with weight
10	F_4	BOW
11		TF-IDF
12		BOW + (F_2 + F_4 + F_5)
13		TF-IDF + (F_2 + F_4 + F_5)
14		BOW + TF-IDF
15		BOW + TF-IDF + (F_2 + F_4 + F_5)
16		BOW + (F_2 + F_4 + F_5) with weight
17		TF-IDF + (F_2 + F_4 + F_5) with weight
18		BOW + TF-IDF + (F_2 + F_4 + F_5) with weight
19	F_5	BOW

20		TF-IDF
21		BOW + ($F_2 + F_4 + F_5$)
22		TF-IDF + ($F_2 + F_4 + F_5$)
23		BOW + TF-IDF
24		BOW + TF-IDF + ($F_2 + F_4 + F_5$)
25		BOW + ($F_2 + F_4 + F_5$) with weight
26	F_5	TF-IDF + ($F_2 + F_4 + F_5$) with weight
27		BOW + TF-IDF + ($F_2 + F_4 + F_5$) with weight
28	$F_2 + F_4 + F_5$	BOW
29		TF-IDF
30		BOW + ($F_2 + F_4 + F_5$)
31		TF-IDF + ($F_2 + F_4 + F_5$)
32		BOW + TF-IDF
33		BOW + TF-IDF + ($F_2 + F_4 + F_5$)
34		BOW + ($F_2 + F_4 + F_5$) with weight
35		TF-IDF + ($F_2 + F_4 + F_5$) with weight
36		BOW + TF-IDF + ($F_2 + F_4 + F_5$) with weight
37	($F_2 + F_4 + F_5$) with weight	BOW
38		TF-IDF
39		BOW + ($F_2 + F_4 + F_5$)
40		TF-IDF + ($F_2 + F_4 + F_5$)
41		BOW + TF-IDF
42		BOW + TF-IDF + ($F_2 + F_4 + F_5$)
43		BOW + ($F_2 + F_4 + F_5$) with weight
44		TF-IDF + ($F_2 + F_4 + F_5$) with weight
45		BOW + TF-IDF + ($F_2 + F_4 + F_5$) with weight

Modeling

After the feature values have been obtained then the next stage is modeling. Modeling was conducted using SVM one vs rest algorithm with linear kernel. One vs rest is derived algorithm from SVM that aimed to solve multiclass classification problem. This method simplifies multiclass classification into binary classifications (classification with two classes and an object included into one class). In one vs rest (OVR), the classification for K classes will be simplified into K binary classifications problem, in which each classification compares a class with another K-1 classes [19].

The matrix value contains best feature combination was fed into algorithm. The result is classification model then is stored in the form of a file by using a pickle on python. We chose pickle format due to the good ability to serialize and de-serialize a python object structure. Saving the model in the files is intended to speed up the prediction process since there is no need to reformat the model.

Evaluation

Parameters that used for model evaluation are precision, recall and F1-score. We selected model with the best score in overall after some testing. Evaluation used random data from corpus.

Visualization

At this stage, the prediction result that already obtained will be displayed to the user. The visualization method used is mind map. Each sentence will be grouped by its class and will be connected to one node that will be rooted. The root is an image that represents the title of the article. The image is located on the center of mind map. The other nodes that represent the 5W+1H class are connected to the root. Other nodes are the word / phrase / sentence that are predicted to be one of its classes. Image for root was taken by using the service from google i.e. the google custom search.

We also conducted a further process into the result of predictions in some classes. In 'when' class, we extract time component from selected sentences using regular expression. For 'where' and 'who' classes, we use named entity recognition (NER) to extract related component from selected sentences. For 'what' class, we chose to display only sentence with the highest F5 score and leave 'why' and 'how' classes as are.

Named entity recognition is a process to extract information from word(s). Named entity recognition will give word(s) a label according to what class that word(s) belong. We used polyglot from python to implement NER. In polyglot there are 3 classes for word(s), person, place, and organization [16]. Label Person and organization used for leaves in 'who' branch, and label place used for leaves in 'where' branch.

Model Updating

We were aware that an app should be evaluated. Therefore, we facilitated the user of our application for reviewing the result of predictions. After the user reviews the result, they can determine to agree or disagree with the results. If they agree, they can save the results to the database. This scheme enables us to grab new dataset that has been labeled by our application, reviewed and evaluated by user. This data was used to update the model, so the prediction results are more accurate.

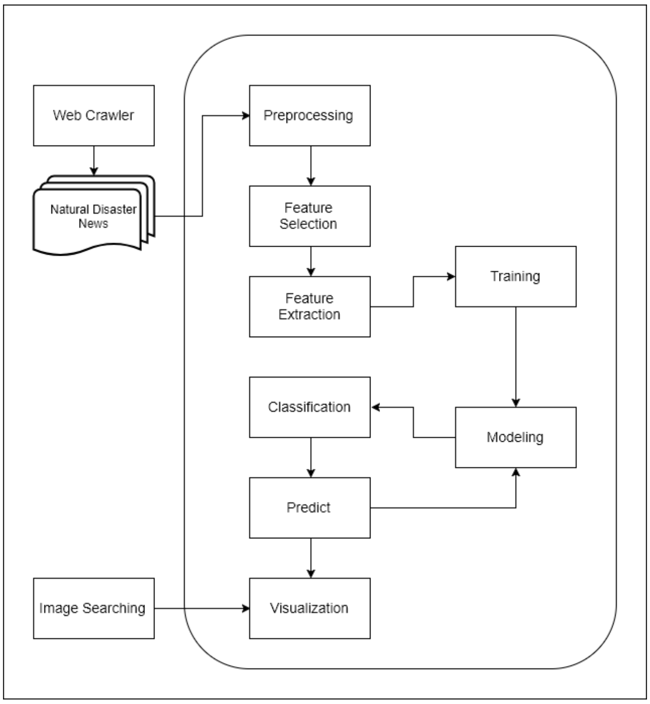

Figure 1: Automatic Mind Map Generator Research Design

Figure 1 outlines our research. Our research focuses on circled section where the steps include preparation step even visualization step. The preparation step starts from crawling for the disaster news from online news channels. Then the disaster news are preprocessed and fed into the algorithms to build classification model. Classification model is used to predict the class of each sentence written in the new disaster news. The class resulted are visualized by using related image crawled from the internet.

V. EXPERIMENTAL STUDY

In our research, we applied python programing language to develop our application. Python programming language provides rich environment for machine learning and text mining. We also used few packages from python in our research. In preprocessing stage, we used NLTK for tokenization and Sastrawi library for stemming. In feature extraction stage, we used Sklearn to extract BOW and TF-IDF and to evaluate the model. In visualization stage, we used Polyglot for named entity recognition and D3js which is JavaScript framework for visualization to create our mind map. From the experiments, we could conclude that F4 is best filter with BOW and TF-IDF as features used in model. The result of precision, recall and F1-score can be seen as follows:

Table 2. The evaluation matrix of model

Filter	Feature	Class	Precision	Recall	F1-score
(1)	(2)	(3)	(4)	(5)	(6)
F_4	BOW and TF-IDF	what	0.86	0.90	0.88
		where	0.74	0.75	0.75
		how	0.33	0.25	0.29
		when	0.78	0.66	0.71
		who	0.80	0.76	0.78
		why	0.45	0.21	0.29
		average	0.77	0.74	0.75

Table 2 gives us information that the classification model can predict well what disaster happened, where disaster happened, and who the victims or people involved and when disaster happened. But, it fails to predict well how and why disaster happened. Our observation on corpus and labeled data set told us that the corpus lacks of sentences which can explain how and why disaster happened.

VI. RESULT AND DISCUSSION

In this research, we have built an application that takes news article as an input and generate mind map visualization as an output. This mind map generator includes complete text mining pipeline in Bahasa Indonesia, facilitates the user to create mind map without reading the passage first, and lets the users evaluate the mind map result. We also created the best model from filtering features and candidate features. From this study, we concluded that F_4 is the best filter to summarize a news article while BOW and TF-IDF features are the best features for classification. The conclusion was considered using the precision, the recall, and the F1-score measurements.

We performed System Usability Scale (SUS) test to measure our application's performance from user point of view. We used SUS because it is an evaluation method which is cheap, simple and easy to implement. The evaluation conducted to ten users that already have prior knowledge about text mining or data mining. The result of evaluation can be seen as follows:

Table 3. The SUS Score

Num	**Respondent Score**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**	**(11)**
1	4	3	4	5	5	5	4	4	4	5
2	2	2	2	2	2	2	2	2	1	2	2
3	4	5	5	4	4	5	4	5	5	5
4	3	2	2	3	3	2	2	2	2	2
5	4	3	4	4	4	4	5	5	4	4
6	3	2	2	2	2	2	1	3	1	1
7	3	5	4	5	4	4	3	5	4	5
8	2	2	1	2	2	2	2	1	2	2
9	4	4	4	3	4	5	4	4	5	5
10	3	1	2	3	2	2	2	2	3	2

We convert the respondent scores and got average score of 78.5. Score of 78.5 is above the average of 68. Converted to alphabetical score, our application's score is B+. We can conclude that this mind map generator does not confuse the new user to use all delivered menus (application's features) and overall fulfil the users' needs.

For future works, we have some suggestions to improve and extend this research. The first is construction of additional model, so that mind map can be made from news article with themes beside natural disaster. We mean that the themes should be various such as economic, politic, social, etc. The second is about algorithm used to build model. The next work can try to build some classification model from some different algorithms and do benchmarking against them. The last is acceleration of the computing time at the preprocess stage, so it can lower the running time of the program overall. In this research, our application consumes approximately 30 seconds to read the input, process it and visualize the mind map.

REFERENCES

[1] R. J. Tierney and J. W. Cunningham, "Research on teaching reading comprehension," *Handb. Res. Read.*, pp. 609–655, 1980.

[2] K. Zubrinic, D. Kalpic, and M. Milicevic, "The automatic creation of concept maps from documents written using morphologically rich languages," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12709–12718, 2012.

[3] Aristoteles, Y. Herdiyeni, A. Ridha, and J. Adisantoso, "Text Feature Weighting for Summarization of Documents in Bahasa Indonesia Using Genetic Algorithm," *IJCSI Int. J. Comput. Sci.*, vol. 9, no. 3,

pp. 1–6, 2012.

[4] M. Elhoseiny and A. Elgammal, “Text to multi-level MindMaps: A novel method for hierarchical visual abstraction of natural language text,” *Multimed. Tools Appl.*, vol. 75, no. 8, pp. 4217–4244, 2016.

[5] J. Asian, “Effective Techniques for Indonesian Text Retrieval,” RMIT University, 2007.

[6] A. Z. Arifin, I. P. A. K. Mahendra, and H. T. Ciptaningtyas, “Enhanced Confix Stripping Stemmer And Ants Algorithm For Classifying News Document In Indonesian Language,” in *The 4th International Conference on Information & Communication Technology and System*, 2007, pp. 149–158.

[7] F. Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” Universiteit van Amsterdam, 2003.

[8] D. Suhartono, D. Christiandy, and R. Rolando, “Lemmatization Technique in Bahasa: Indonesian Language,” *J. Softw.*, vol. 9, no. 5, pp. 1202–1209, 2014.

[9] S. S. Megala, A. Kavitha, and A. Marimuthu, “Enriching Text Summarization using Fuzzy Logic.,” *Int. J. Comput. …*, vol. 5, no. 1, pp. 863–867, 2014.

[10] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text classification using machine learning techniques,” *WSEAS Trans. Comput.*, vol. 4, no. 8, pp. 966–974, 2005.

[11] S. Bahassine, M. Kissi, and A. Madani, “New stemming for arabic text classification using feature selection and decision trees,” in *Proceedings of the 5th International Conference on Arabic Language Processing*, 2014, pp. 200–205.

[12] L. K., K. S., and R. R., “Information Extraction from Biomedical Literature using Text Mining Framework,” *Int. J. Imaging Sci. Eng.*, vol. 1, no. 1, pp. 1–5, 2007.

[13] A. D. Tahitoe and D. Purwitasari, “Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia dengan Metode Corpus Based Stemming,” 2010.

[14] B. A. A. Nazief and M. Adriani. Confix-stripping: Approach to stemming algorithm for Ba- hasa Indonesia. Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta, 1996

[15] G. G. Chowdhury, “Natural language processing,” *Annual review of information science and technology*, vol. 37, no. 1, pp. 51-89, 2003.

[16] R. Al-Rfou, V. Kulkarni, B. Perozzi and S. Skiena. POLYGLOT-NER: Massive Multilingual Named Entity Recognition, 2014.

[17] R. Kudelić, Robert, M. Konecki, M. Maleković, “Mind map generator software model with text mining algorithm,” *Proceedings of the International Conference on Information Technology Interfaces, ITI.*, pp. 487 - 494, 2011.

[18] N. Chatterjee and S. Mohan, “Extraction-Based Single-Document Summarization Using Random Indexing,” *International Conference on Tools with Artificial Intelligence*, vol. 19, pp. 448-455, 2007.

[19] R. Rifkin and A. Klautau, “Parallel networks that learn to pronounce english text,” *Journal of Machine Learning Research*, pp. 101-141, 2004.