Text Classification Using XLNet with Infomap Automatic Labeling Process

Triana Dewi Salma
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
tdsalma@gmail.com

Gusti Ayu Putri Saptawati
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
putri@informatika.org

Yanti Rusmawati
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
yanti@informatika.org

2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA) | 978-1-6654-1743-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAICTA53211.2021.9640255

***Abstract*—Text data is growing rapidly and used in various fields such as chatbots and question answering systems, which are currently popular, where the system identifies the question category and the possibility of an answer to help provide answers to the questions entered. Having good quality text data, especially in text classification, significantly affects the performance of the model. Manual labeling by humans, generally used in labeling training data in supervised learning, is expensive, prone to mistakes, and has a low quantity. Automatic labeling that providing high quality and high quantity of training data is necessary to improve text classification performance. This study attempts to conduct community detection with the Infomap algorithm for automatic labeling in text classification using XLNet. The accuracy of the model is compared to the baseline, which using data with manual labeling. While the accuracy has not outperformed the overall baseline yet, but the result shows that automatic labeling can improve data labeling quickly with high quantity.**

Keywords—natural language processing, text classification, chatbot, question answering, community detection.

I. INTRODUCTION

In this big data era, the quantity of information that we can find in everyday life is huge with very fast transmission, and the structure of the available data varies greatly [1]. One of the data that is growing rapidly and widely used is text data used in various fields, including Natural Language Processing (NLP) tasks, such as chatbots or question answering systems, which are currently popular. The system will identify the question category and the possibility of an answer to help provide answers to the questions entered [2]. In the NLP process, especially text classification, having past data as training data with good quality will affect the performance of the model. Generally, training data in supervised learning must be created manually by humans who are experts in their fields, resulting in high costs and limited high-quality training data [3]. Manual data labeling by humans is prone to mislabeling the data and it is difficult to trace back whether the data has been labeled correctly or not, which ultimately has an impact on the quality of the model trained in classification [4].

Problem solving in labeling a document is done by unsupervised learning, such as using LDA for topic grouping [5]. A similar study was conducted using the LDA-*K*means hybrid [6]. The use of *K*-means is considered not optimal because the quality of the classification results will greatly affect the *k* value defined by the researcher and requires a lot of experimentation to find the correct *k* value for certain data.

Network science, especially community detection, can also deal with this problem, but there are still very few studies that apply it in text classification [4]. One of the studies on community detection was conducted using Louvain to analyze text in the implementation of text co-clustering on textual data [7]. Community detection also helps in data clustering without depending on user parameters by utilizing modularity as in [8]. Both studies provide examples of the application of community detection in grouping text but have not provided a more detailed explanation of the application and performance effects on classification. The use of community detection specifically in text classification to label training data has been able to improve the performance of text classification using machine learning, namely 3.75% in SVM and 2.68% in Random Forest [4].

In recent years, deep learning has become the center of attention for various fields such as image processing, NLP, and computer vision [9]. Deep learning can take advantage of large datasets to achieve a higher level of accuracy than previous classification techniques [10]. One of the newest deep learning models, XLNet, has received a state-of-the art predicate for 18 NLP tasks [11].

Research [4] also applies the Louvain algorithm as community detection, while on the one hand, Louvain is the second best algorithm after Infomap [12]. In addition, research [4] uses bigram tokenization and explains that the opportunity to improve performance can be done by using a higher *n*-gram because it has a strong role in improving syntactic performance and can capture broader contextual information [13].

In this study, we employ deep learning, especially XLNet to the model for text classification. The Infomap algorithm is used in the training data labeling automatically so that it can improve the quality of the training data. Furthermore, we explore the effect of using Bigram and Trigram to their performance in the experimental model.

II. RELATED WORK

Infomap was first introduced in 2008 to understand the multipartite organization of large-scale biological and social systems [14]. Research on Infomap has been conducted by [15], [16], [12], [17], and [18]. The Infomap algorithm is able to group graphs with good quality [16]. In a study conducted by [18], they were able to classify texts based on their topic using the Infomap algorithm.

XLNet is a pre-training method that uses permutation language modeling objectives to combine the advantages of Autoregressive (AR) and Autoencoder (AE) methods and was introduced in 2019 [19]. The application of XLNet has been carried out in studies [18], [20], [21], and [22]. XLNet has become a state-of-the art task range from NLP and excels at 18 NLP tasks [11].

III. METHOD

The system architecture used in this study refers to research [4] which is adjusted so that it becomes two stages, namely the automatic labeling stage of training data using Infomap and the text classification stage using the XLNet model.

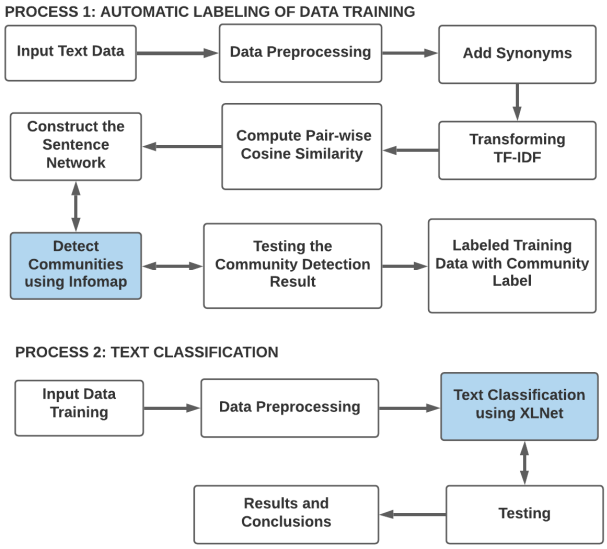

Fig.1. Research flow

A. Data

The data used in this study was taken from TREC-QA, one of the open domain question-answer datasets that are often used in research [20]. This dataset is taken from research conducted by [23], consisting of two columns, questions and class labels. There are 6 class labels, namely ABBR (Abbreviation), DESC (Description), ENTY (Entity), HUM (Human), LOC (Location), NUM (Numeric value). There are 5,452 training data collected from three sources and labeled manually, while the test data is 500 questions.

B. Automatic Labeling of Data Training

This stage aims to generate labeled training data automatically by utilizing community detection using the Infomap algorithm.

1) Preprocessing: Preprocessing steps include expanding Contractions or restoring contractions to the original word order, deleting URLs, removing punctuation marks, deleting stopwords, and stemming.

2) Addition of Synonyms: The addition of synonyms aims to increase the variety of words and is expected to improve the understanding of the resulting classification model for different words but describe the same meaning. Adding synonyms makes use of the NLTK-wordnet in Python to add synonyms in different contexts.

3) TF-IDF Transformation: Each sentence is transformed into a vector representation by performing TF-IDF calculations. This stage considers the word or term frequency (TF) in the document as well as how unique or infrequent (IDF) a word is throughout the corpus, so TF-IDF gives higher scores for unique words and devalues common words [24]. In his research, Zhang stated that the basic form of TF-IDF could be denoted by the weight value of TF-IDF *wt*, *d* for the word *t* in document d in the equation

$$w_{t,d}=tf_{t,d} \times \log_{10}\left(\frac{N}{df_t}\right) \quad (1)$$

Where *tft*, *d* is the frequency of the word *t* in document *d*, *N* is the number of documents in the collection, and *dft* is the number of documents where the word t appears. The TF-IDF transformation consisted of two types of Ngrams, namely bigram and trigram. These two types of Ngrams will be compared in this study to determine their effect on the automatic labeling process.

4) Calculation of Cosine similarity: When the data has been represented in a vector using TF-IDF, the equation between the two vectors in each training data is calculated using cosine similarity. Cosine similarity has a value range between 0 and 1 with a value of 0 which means that the two vectors have no similarity, and if the cosine similarity equal to 1, the two sentence vectors have high closeness [4]. The result of cosine similarity calculation is a matrix of closeness between sentence vectors with a cosine similarity value of 1 for each sentence vector compared to itself as shown in Fig. 2 below.

$$\begin{array}{c} \\ d1 \\ d2 \\ \vdots \\ d5452 \end{array}\begin{array}{c} \begin{array}{cccc} d1 & d2 & \cdots & d5452 \end{array} \\ \begin{bmatrix} SIM_c\left(\overrightarrow{t_{d1}},\overrightarrow{t_{d1}}\right) & SIM_c\left(\overrightarrow{t_{d1}},\overrightarrow{t_{d2}}\right) & \cdots & SIM_c\left(\overrightarrow{t_{d1}},\overrightarrow{t_{d5452}}\right) \\ SIM_c\left(\overrightarrow{t_{d2}},\overrightarrow{t_{d1}}\right) & SIM_c\left(\overrightarrow{t_{d2}},\overrightarrow{t_{d2}}\right) & \cdots & SIM_c\left(\overrightarrow{t_{d2}},\overrightarrow{t_{d5452}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ SIM_c\left(\overrightarrow{t_{d5452}},\overrightarrow{t_{d1}}\right) & SIM_c\left(\overrightarrow{t_{d5452}},\overrightarrow{t_{d2}}\right) & \cdots & SIM_c\left(\overrightarrow{t_{d5452}},\overrightarrow{t_{d5452}}\right) \end{bmatrix} \end{array}$$

Fig.2. Closeness matrix

5) Formation of Sentence Network: The sentence network is formed from the nodes of each sentence in the training data and the weights obtained from the calculation of the proximity matrix that has been carried out. This sentence network formation utilizes the networkx library in Python.

6) Infomap Community Detection: Infomap, according to [15], to divide the network, the first step is that each community will be coded according to the community level, and then in each community the node will be coded based on the node level. By integrating these two aspects, the coding of one node is confirmed by community-coding and node-coding. Therefore, the problem of community detection can be replaced by the problem of coding compression, which makes the length of the encoding the shortest. When optimizing the objective function, the algorithm will divide the nodes that are closely connected to each other into the same community, because the objective function is the total length of the random walk path encoding in the network. In this case, the best method for community detection would be to obtain the maximum amount of encoding compression.

7) Testing of Automatic Labeling Results: The first test of the automatic labeling stage of training data is carried out using modularity calculations. One type of evaluation based on modularity proposed by [25], known as Newman Girvan Modularity, has a value ranging from 0 to 1, with a value of 1 being the maximum value. The Newman Girvan modularity formula can be calculated through the following equation.

$$Q(S) = \frac{1}{m}\sum_{c\in S}\left(m_S - \frac{(2m_S + l_S)^2}{4m}\right) \quad (2)$$

Where m is total edges in graph, m_s is number of edges in community, and l_S is the number of edges from node S to nodes outside of S.

The second test is by calculating the average value of the class split and class merge. Class split occurs when a manually labeled class is mapped in many communities due to semantic similarity. In contrast, a merge class is when several or many manually labeled classes are detected in the same community [4]. We attempt to minimize both values because if the class split is high, many communities are formed. However, if the merge class has a high value, the results of community detection are considered not grouped correctly. Based on both values, the average is calculated to be taken into consideration in determining the community detection results.

C. Text Classification

The pre-trained XLNet model used refers to research [19], which has passed pre-train on BooksCorpus, English Wikipedia, Giga5, ClueWeb, and Common Crawl. Classification is done by dividing the training data into 80% as training data and 20% as validation data. Text classification is implemented based on a model that has passed the pre-train process with the XLNet model for sequence classification base cased. This model is then fine-tuned to better understand the training data that is owned and the number of epochs is set, namely 10, learning rate 3e-5, batch size in the range 1-5, max_len 64, and AdamW optimizer. The results of the text classification will be measured by accuracy and the value of the loss function to be compared in each scenario.

IV. Experiment and Analysis

The experimental scenario is as follows:

- Baseline text classification model using XLNet with manually labeled training data sourced from TREC-QA.
- Text classification uses XLNet with training data from the detection of the Infomap community and uses Bigram.
- Text classification uses XLNet with training data from the detection of the Infomap community and uses Trigram.

A. Result

In the process of forming a sentence network on automatic labeling, the nodes represent each sentence of the training data. The edge weight is obtained from the calculation of the proximity matrix that has been done. The number of nodes formed is 5452 nodes with a number of edges of 1,232,614 edges. This network of sentences then enters the community detection process, which is carried out using the cdlib library. Community detection will produce training data that has been labeled as a community, the number of communities formed, modularity, and the value of class split and class merge. The series of experiments were carried out on bigram data and trigram data to be compared between the results of automatic labeling of training data.

TABLE I. Result of automatic labeling

Ngram	Number Of Community Formed	Modularity	Class Split	Class Merge
Bigram	1007	0,559	364,833	2,171
Trigram	706	0,5793	309,33	2,626

The results of automatic labeling training data using Bigram and Trigram are used in text classification using the XLNet model as previously explained. The hyperparameters used are max len 64, batch size 2, Adamw optimizer, and learning rate 3e-5. The results of recording experiments carried out can be seen in the following table.

TABLE II. Text classification results with automatic labeling data

Epoch	Bigram			Trigram		
	Time	*Accuracy*	*Loss*	*Time*	*Accuracy*	*Loss*
4	17min 59s	0,0112	6,0271	17min 58s	0,0169	0,0169
6	26min 56s	0,0123	6,0268	26min 33s	0,0169	5,7236
8	56min 12s	0,0112	6,0305	36min 25s	0,0169	5,718
10	44min 23s	0,0123	6,0780	44min 58s	0,0169	5,73

As a comparison, text classification using manually labeled training data gives experimental results with the same hyperparameter as follows

TABLE III. Text classification results with Manual labeling data

Epoch	Manual Label Data		
	Time	*Accuracy*	*Loss*
4	35min 31s	0,4651	1,3933
6	53min 17s	0,24192	1,62577
8	1h 10min 45s	0,8868	0,66171
10	1h 29min 24s	0,224321	1,67461

B. Discussion

The experimental results according to the scenario cannot outperform the baseline, namely text classification with manually labeled training data. Things that affect the results can be analyzed in the following points.

1) Threshold

The experimental results according to the scenario cannot outperform the baseline were caused by the weight of the cosine similarity used in the construction of the edge of the sentence network to form an edge of 1,232,614 for 5452 nodes. This also results in a high class split and class merge values. If the class split is high, it affects the effectiveness of the classification because more and more communities are formed with only a few members in each community, even only one node. On the other hand, if the merge class is high, the community detection results cannot group nodes properly because it maps many classes from manual labels in the same community. Therefore, the value of class split and class merge must be minimized for optimal community detection results.

The experimental threshold is intended to limit the edge that has a weight below the threshold. In this experiment, the threshold test was carried out in a value range of 0.0 to 0.6 with the following results.

TABLE IV. THRESHOLD TEST RESULTS ON BIGRAM

Threshold	Number Of Community Formed	Modularity	Class Split	Class Merge
0.0	1007	0,559	364,833	2,171
0,1	311	0,69251	161,1666	3,09
0,2	194	0,827	101,833	3,118
0,3	350	0,8009	210	3,582
0,4	348	0,624	134,834	2,3074
0,5	97	0,425	35,5	2,134
0,6	22	0,2698	13,666	3,4545

TABLE V. THRESHOLD TEST RESULTS ON TRIGRAM

Threshold	Number Of Community Formed	Modularity	Class Split	Class Merge
0,0	706	0,5793	309,33	2,626
0,1	283	0,7637	144,334	3,03886
0,2	447	0,82031	286	3,8255
0,3	789	0,6225	236	1,787
0,4	83	0,4702	42,664	3,012
0,5	55	0,2837	21	2,1818
0,6	12	0,168	9,666	4,333

Based on the experiment of limiting the edge with the threshold, normalization is then carried out to determine the optimum threshold point that can be used. The class split and class merge graphics after normalization are as shown in Fig. 3 and Fig. 4 below.

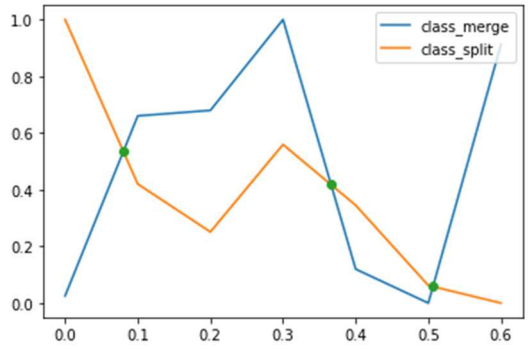

Fig.3. Threshold normalization in Bigram

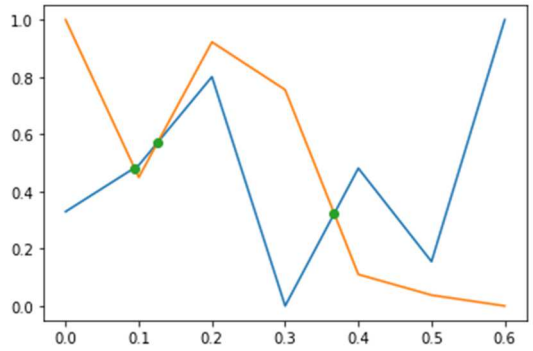

Fig.4. Threshold normalization in Trigram

It can be seen that on the bigram graph, there are three intersection points between class split and class merge. Based on the experimental threshold objective, to minimize class split and class merge value, the third cutoff point is taken at the threshold of 0.506382. While on the trigram graph, there are three intersection points with a minimum value of class split and class merge at point 0.367058.

The experimental threshold results are used in the automatic re-labeling process. The number of edges formed is 7712 on Bigram and 10098 edges on trigram. The result is shown in the following table. The value of class split and class merge can be minimized while maintaining maximum modularity.

TABLE VI. AUTOMATIC LABELING RESULTS WITH AN OPTIMUM THRESHOLD

Ngram	Threshold	Number Of Community Formed	Modularity	Class Split	Class Merge
Bigram	0,506383	69	0,414465	31,166	2,6231
Trigram	0,367058	105	0,540516	57,333	3,219

The training data resulting from automatic labeling using the optimum threshold is used as input in the text classification process using the previous XLNet model. The results of the text classification carried out are as follows.

TABLE VII. THE RESULT OF TEXT CLASSIFICATION WITH THE OPTIMUM THRESHOLD

Epoch	Bigram			Trigram		
	Time	*Accuracy*	*Loss*	*Time*	*Accuracy*	*Loss*
4	4min 49s	0,2766	2,4984	6min 37s	0,2134	3,1927
6	7min 21s	0,2766	2,5238	9min 52s	0,2134	3,1903
8	9min 49s	0,33471	1,9179	13min 7s	0,2134	3,2018
10	12min 3s	0,3554	1,8742	16min 32s	0,2860	2,2220

The superior results of the experiment against the baseline were not all of the epochs that were carried out. In Bigram data, the experiment only excels at epoch 6 and 10. Whereas in the trigram data, the experiment is only superior at epoch 10. When deepening and visualizing the accuracy of each scenario as in Fig. 5 below, it can be seen that in epoch 9 and 10 scenarios with training data labeled manually, the accuracy decreases. On the other hand, in Bigram and Trigram's scenario, the accuracy rate increases at the 10th epoch.

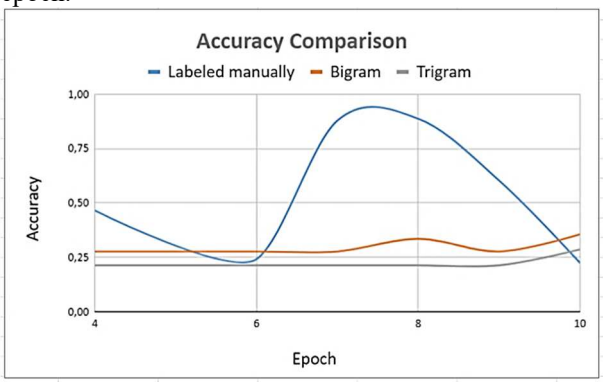

Fig.5. Comparison of classification results with the optimum threshold

2) Nodes without community

Another thing that needs to be considered when using automatic labeling is that there are nodes that do not have a

community, so that the amount of training data that has a community label is reduced. When using the optimum threshold, it is found that training data that has a community is 1446 on bigram and 2010 data on the trigram of the total training data of 5452. This means that 73.4% of the training data is on bigram, and 63.13% of the training data is on the trigram has no community.

3) Class Split and Class Merge

In addition to the effect of the large amount of data that does not have a community, the effect of low accuracy is due to class splits and class merges in every automatic labeling that is carried out. These two things, class split, and class merge, cannot be avoided in this automatic labeling experiment. Even though it has been minimized, some sentences should be grouped into different communities but labeled with the same community. With the class merge value of 2.623188 on bigram data and 3,219 on trigram data, there are still communities that group 6 manual labels in one community. These nodes do not have a close cosine similarity when explored further, but by the Infomap algorithm, they are grouped into one community. Many things can affect this, like semantics, or nodes are connected indirectly by other nodes in the graph network.

4) Bigram and Trigram

Comparison of the effect of bigram and trigram on the experiments conducted shows the general results that the addition of trigrams in training data increases accuracy. However, when viewed after the optimal threshold experiment was carried out, the accuracy of the trigram data is not superior to the accuracy of Bigram data. This is due to the results of determining the different threshold cutoff points in the two data. Bigram data has the optimal cutoff point at 0.506382 and while in trigram data, the intersection point is at 0.367058. The optimal cutoff point for trigram data results in lower accuracy when compared to the accuracy results generated on trigram data with a threshold of 0.5 and 0.6. This comparison can be seen in Table VIII below.

TABLE VIII. COMPARISON OF THE THRESHOLD ON TRIGRAM DATA

Epoch	Threshold 0,5	Threshold 0,6	Threshold 0,367058
	Accuracy	*Accuracy*	*Accuracy*
4	0,3267	0,398	0,213430
6	0,3267	0,393	0,213432
8	0,3336	0,393	0,213432
10	0,3267	0,393	0,28606

5) Use of Keywords

The experimental analysis above resulted in a temporary conclusion that the sentence conditions in the training data used as process input were very important and influential. The sentences in the training data will be processed as input from the initial automatic labeling process to text classification. For example, the addition of synonyms will enrich words with the same context, thereby increasing the effect on the closeness between sentences that have the same context. Likewise, the use of available keywords from the training data will also increase the closeness of sentences that have the same keywords. To ensure that the use of keywords in the training data used is the most appropriate scenario, the researchers reexamined the scenario of using synonyms without keywords. The results of text classification with the same XLNet model at the optimum threshold value with training data using synonyms without keywords in bigram, the accuracy is 0.289 with a loss function of 2.038. Whereas in the trigram data, the accuracy is 0.199270 and the loss function is 3.4359.

6) Use of synonyms

The addition of synonyms aims to have words with the same context to increase the effect of closeness between sentences with the same context. Experiments were carried out with scenarios without synonyms and using keywords to determine the effect of adding synonyms.

TABLE IX. ACCURACY RESULTS WITHOUT SYNONYMS

Epoch	Bigram		Trigram	
	Time	*Accuracy*	*Time*	*Accuracy*
4	6min 42s	0,2375	11min 32s	0,2631
6	10min 5s	0,2559	17min 15s	0,2631
8	13min 22s	0,2529	23min 1s	0,2631
10	16min 57s	0,2375	28min 45s	0,2631

The table concluded that the use of synonyms affects the accuracy of the bigram data because it tends to be lower for all epochs. While for trigram data, the accuracy is high at epochs 4, 6, and 8. This is due to the high value of class split in bigrams data with more communities. In trigram data, the communities formed after the community detection process are fewer.

After the experimental scenarios were carried out, it turned out that automatic labeling data could increase the speed of the classification process. Based on experiments with the optimum threshold, both in the scenario using synonyms and without synonyms can increase the average classification speed by 79.13%, although with a decrease of average accuracy of 42, 15%. Especially when compared to humans' manual labeling process, which requires high time and costs, the automatic labeling process is superior in terms of speed.

V. CONCLUSIONS

Automatic labeling with the Infomap algorithm automatically generates training data that is labeled based on the community. In testing, the results of automatic labeling must refer to high modularity as well as pay attention to the level of class split and class merge. The class split affects the number of communities produced. If there are many communities produced, it will affect the accuracy of text classification. Meanwhile, the class merge has an effect on the quality of community detection because it classifies data from multiple classes with manual labels into one community.

Using community detection for automatic labeling allows us to use data that is not community-based. Therefore, the experiment significantly reduced the amount of training data, i.e., 73.4% of the training data on bigram and 63.13% of the training data on the trigram. The accuracy of the experiments has not outperformed the baseline using manually labeled data yet. Although community detection in automatic labeling helps to label quickly with high quantity, several things affect the performance, such as the class split, class

merge, data that does not have a community, and semantic factors. The use of trigram to some extent produces a higher level of accuracy than the use of bigram. However, the results show that the optimum threshold accuracy at trigram is not superior to bigram.

For future works, we intend to address the data handling that does not have a community by considering semantic analysis so that sentences can be labeled as a whole. Moreover, we can minimize class split and class merge to improve the performance of text classification.

ACKNOWLEDGMENT

We sincerely thank to Indonesia Endowment Fund For Education (LPDP), who has funded this research and master's study through the LPDP Regular Scholarship program.

REFERENCES

[1] R. Yu, "Reform in the teaching model of english writing in the big data era," in *2020 IEEE 2nd International Conference on Computer Science and Educational Informatization (CSEI)*., pp. 252-259.

[2] S. Yilmaz and S. Toklu, "A deep learning analysis on question classification task using Word2vec representations," *Neural Computing and Application.*, vol. 32, no. 7, pp. 2909–2928, 2020, doi: 10.1007/s00521-020-04725-w.

[3] F. Prior *et al.*, "Open access image repositories : high-quality data to enable machine learning research," *Clinical Radiol.*, vol. 75, no. 1, pp. 7–12, 2020, doi: 10.1016/j.crad.2019.04.002.

[4] M. Kim and H. Sayama, "The power of communities: A text classification model with automated labeling process using network community detection," in *Proc. of NetSci-X 2020: Sixth International Winter School and Conference on Network Science*. Springer Nature, 2020 pp. 231–243, doi: 10.1007/978-3-030-38965-9_16.

[5] A. Lancichinetti, M. I. Sirer, J. X. Wang, D. Acuna, K. Körding, and L. A. N. Amaral, "High-reproducibility and high-accuracy method for automated topic classification," *Physical Review X*, vol. 5, no. 1, pp. 011007, Jan. 2015, doi: 10.1103/PhysRevX.5.011007.

[6] O. M. Foong and A. N. Ismail, "Document clustering using hybrid LDA-Kmeans," in *Silhavy R. (eds) Applied Informatics and Cybernetics in Intelligent Systems.* CSOC 2020. Advances in Intelligent Systems and Computing, vol 1226. Springer, Cham. https://doi.org/10.1007/978-3-030-51974-2_12.

[7] L. Celardo and M. G. Everett, "Network text analysis: A two-way classification approach," *International Journal of Information Management*, vol. 51, pp. 102009, Apr. 2020, doi: 10.1016/j.ijinfomgt.2019.09.005.

[8] E. K. Mikhina and V. I. Trifalenkov, "Text clustering as graph community detection," in *Procedia Computer Science*, 2018, vol. 123, pp. 271–277, doi: 10.1016/j.procs.2018.01.042.

[9] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Compuing.*, vol. 22, no. 1, pp. 949–961, Jan. 2019, doi: 10.1007/s10586-017-1117-8.

[10] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. - 2016 IEEE European Symposium on Security and Privacy (EURO S&P)*, pp. 372–387, doi: 10.1109/EuroSP.2016.36.

[11] K. Nagda, A. Mukherjee, M. Shah, P. Mulchandani, and L. Kurup, "Ascent of pre-trained state-of-the-Art language models," in *Advanced Computing Technologies and Applications*, Springer, Singapore, 2020, pp. 269–280.

[12] S. Fortunato and D. Hric, "Community detection in networks : A user guide," *Phys. Rep.*, vol. 659, pp. 1–44, 2016, doi: 10.1016/j.physrep.2016.09.002.

[13] P. Gupta, M. Pagliardini, and M. Jaggi EPFL, "Better word embeddings by disentangling contextual N-gram information," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 933–939.

[14] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," in *Proc. of the national academy of sciences*, vol. 105, no. 4, pp. 1118–1123, 2008, doi: 10.1073/pnas.0706851105.

[15] F. Hu and Y. Liu, "A novel algorithm infomap-SA of detecting communities in complex networks," *J. Commun.*, vol. 10, no. 7, pp. 503–511, 2015, doi: 10.12720/jcm.10.7.503-511.

[16] J. Zeng and H. Yu, "A distributed infomap algorithm for scalable and high-quality community detection," in *ACM International Conference Proceeding Series*, 2018, pp. 1–11, doi: 10.1145/3225058.3225137.

[17] S. Fortunato and A. Lancichinetti, "*Community detection algorithms: a comparative analysis," P Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top*., vol. 80, no. 5, Nov. 2009.

[18] T. Velden, S. Yan, and C. Lagoze, "Mapping the cognitive structure of astrophysics by infomap clustering of the citation network and topic affinity analysis," *Scientometrics*, vol. 111, pp. 1033–1051, 2017, doi: 10.1007/s11192-017-2299-9.

[19] Z. Yang, Z. Dai, Y. Yang, and J. Carbonell, "XLNet : generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, 2019, pp. 5753-5763.

[20] S. Minaee, "Deep learning based text classification: a comprehensive review," *ACM Computing Surveys (CSUR)*, vol. 1, no. 1, pp. 1–42, 2020.

[21] D. Su *et al.*, "Generalizing question answering system with pre-trained language model fine-tuning," in *Proc. of the 2nd Workshop on Machine Reading for Question Answering*, 2019, pp. 203–211, doi: 10.18653/v1/d19-5827.

[22] W. Antoun, F. Baly, R. Achour, A. Hussein, and H. Hajj, "State of the art models for fake news detection tasks," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT),* pp. 519–524, doi: 10.1109/ICIoT48696.2020.9089487.

[23] X. Li and D. Roth, "Learning question classifiers," in *COLING 2002: The 19th International Conference on Computational Linguistics*, pp. 1–7, doi: 10.3115/1072228.1072378.

[24] Y. Zhang, Y. Zhou, and J. T. Yao, "Feature extraction with TF-IDF and game-theoretic shadowed sets," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems,* Springer, Cham, 2020, vol. 1237 CCIS, pp. 722–733, doi: 10.1007/978-3-030-50146-4_53.

[25] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 69, no. 2 2, p. 026113, Feb. 2004, doi: 10.1103/PhysRevE.69.026113.