International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 5, October 2023, pp. 5641~5652
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i5.pp5641-5652

Multi-label text classification of Indonesian customer reviews using bidirectional encoder representations from transformers language model

Nuzulul Khairu Nissa, Evi Yulianti
Faculty of Computer Science, University of Indonesia, Depok, Indonesia

Article Info

Article history:

Received Dec 8, 2022
Revised Feb 2, 2023
Accepted Feb 10, 2023

Keywords:

Convolutional neural network
Customer review
IndoBERT
Multi-label text classification
Word2Vec

ABSTRACT

Customer review is a critical resource to support the decision-making process in various industries. To understand how customers perceived each aspect of the product, we can first identify all aspects discussed in the customer reviews by performing multi-label text classification. In this work, we want to know the effectiveness of our two proposed strategies using bidirectional encoder representations from transformers (BERT) language model that was pre-trained on the Indonesian language, referred to as IndoBERT, to perform multi-label text classification. First, IndoBERT is used as feature representation to be combined with convolutional neural network-extreme gradient boosting (CNN-XGBoost). Second, IndoBERT is used both as the feature representation as well as the classifier to directly solve the classification task. Additional analysis is performed to compare our results with those using multilingual BERT model. According to our experimental results, our first model using IndoBERT as feature representation shows significant performance over some baselines. Our second model using IndoBERT as both feature representation and classifier can significantly enhance the effectiveness of our first model. In summary, our proposed models can improve the effectiveness of the baseline using Word2Vec-CNN-XGBoost by 19.19% and 6.17%, in terms of accuracy and F-1 score, respectively.

Corresponding Author:

Evi Yulianti
Faculty of Computer Science, University of Indonesia
Depok, West Java, Indonesia
Email: evi.y@cs.ui.ac.id

1. INTRODUCTION

Customer review is a critical resource to discover useful information about user experiences on a particular product (or service). Such information is important for a company to help them making a good decision about their products. A review text may contain user's opinion about several aspects of a product, where each aspect may accept different sentiments from the user. Here is an example of Indonesian customer review of hotel experiences that contains different sentiments for different aspects of hotel: *"kamarnya nyaman dan bersih, tetapi TV nya terlalu tinggi jadi kamu tidak bisa nonton" ("The room is comfortable and clean, but the TV is too high, so you can't watch it")*. In that review, the aspect of "cleanliness" has a positive sentiment, but the aspect of "TV" as one of the hotel's facilities has a negative sentiment.

Aspect category detection or aspect classification is one of the subtasks from aspect-based sentiment analysis (ABSA) [1]. For the aspect classification task, the aspects contained in the text review are identified and the polarity of each aspect is then determined by sentiment classification. The results of this system are

beneficial for a company to understand which aspects of their product that are perceived as positive, and which of those that are perceived as negative by customers. The company can then make some decisions to improve the aspect of product that are still perceived as negative by costumers.

Aspect classification is formulated as multi-label text classification problem [2]. A general text classification problem will associate relevant text with pre-existing labels [3]. According to the number of labels that must be identified, the text classification problem is divided into two categories: multi-class classification and multi-label classification. For multi-label classification, a text can be assigned more than one label, while for multi-class classification, a text will be assigned one label only [4].

Several previous studies have used machine learning methods and problem transformation approaches to solve multi-label text classification task [2], [3], [5], [6]. Problem transformation is an approach that can convert the multi-label case into single-label learning tasks [5]. The two approach of the problem transformation that are frequently used include binary relevance (BR) strategy and classifier chain (CC) strategy. Using the BR approach, a multi-label classification problem with n labels is converted into n binary classification problems [7]. As the BR strategy believes that each label in the dataset is independent of every other label, it completely ignores the correlation between labels. To analyze the correlation between the labels, the CC method will link the labels in a "chain" way [8].

Deep neural network-based multi-label text classification techniques have recently gained popularity due to the rapid growth of deep learning, such as: long short-term memory (LSTM) [9], bidirectional long short-term memory (BiLSTM) [10], convolutional neural network (CNN) [2], [11]–[13] and recurrent convolutional neural network (RCNN) [12]. Apart from the development of various types of models to solve the multi-label text classification, the feature representation that can be used are also evolving. There are several methods that can extract the features from text, including: bag-of-words (BoW), term frequency inverse document frequency (TF-IDF) and context independent text embeddings (Word2Vec, fast-Text and GloVe).

Azhar *et al.* [2] have studied the multi-label text classification for Indonesian customer hotel reviews dataset using Word2Vec as text embedding, CNN as the feature extraction method and various kind of machine learning models (i.e., support vector machines (SVM), long short term memory (LSTM) and extreme gradient boosting (XGBoost)) as the top-level classifiers. They used of BR and CC strategies for the problem transformation approaches. Their experiments show that combining CNN and XGBoost classifier with the CC strategy, which can consider dependencies between the labels, produces better results than using the BR strategy.

Nowadays, the contextualized text embedding method from the pre-trained language model bidirectional encoder representation from transformers (BERT) has been developed and we can also use it to solve the multi-label text classification task. BERT is a state-of-the-art of the contextualized pre-trained language model with a deeper understanding of the language as an effect of bidirectional learning [14]. Besides as feature representation, BERT can also be fine-tuned to effectively solve a range of downstream tasks, such as text classification [11]–[13], [15], [16], question answering [17], named entity recognition [18] and sentiment analysis [19]. BERT also has several types of models, namely: multilingual BERT (mBERT) which is a single pre-trained language model on the concatenation of monolingual Wikipedia corpora from 104 languages. Monolingual BERT, on the other hand, is a BERT model that has only been pre-trained in one language. A recent BERT pre-trained model called IndoBERT was particularly trained utilizing a huge number of the Indonesian language corpus [20], [21]. Some recent work have started to exploit IndoBERT to enhance the effectiveness of their methods on various tasks [16], [22].

Khasanah and Krisnadhi [11], used IndoBERT embedding with a single channel CNN as the classifier to perform the extreme multi-label text classification task. The results demonstrate that their suggested approach (single CNN with IndoBERT) performs better than the single CNN with embedding of fastText and the single CNN with embedding of Word2Vec. Another study from Neruda and Winarko [13] utilized the social media data to identify traffic events using the combination of IndoBERT as the text embedding and CNN as classifier. In comparison to non-contextualized text embedding, BERT's contextualized embedding helps in understanding the context and gives better results.

Depart from the results of Azhar *et al.* [2], Khasanah and Krisnadhi [11], and Neruda and Winarko [13], we propose two strategies to perform multi-label text classification to predict or categorize the aspects from Indonesian hotel reviews dataset. In our first strategy, following the method with the best results in the study from Khasanah and Krisnadhi [11], we propose to use IndoBERT as text embedding that is combined with CNN feature extraction method and XGBoost classifier, replacing Word2Vec embedding that was used in previous studies from Azhar *et al.* [2]. Then, in our second strategy, we use IndoBERT in end-to-end fashion, as feature representation as well as classifier, to directly solve the multi-label classification task. As far as we know, no previous studies have investigated the use of IndoBERT as a multi-label classifier for aspect classification in the Indonesian hotel reviews dataset, since the pre-trained model IndoBERT was still relatively new.

Our research questions in this paper are as follows: i) How is the effectiveness of our first model using IndoBERT as feature representation that is combined with CNN and XGBoost classifier for multi-label text

classification?; ii) How is the effectiveness of our second model using IndoBERT end-to-end model for multi-label text classification?; and iii) How much different is the effectiveness of our models using monolingual language model IndoBERT compared to those using multilingual language model mBERT for multi-label text classification?

2. RESEARCH METHOD

The aspect classification task is formulated as multi-label text classification. This section explains dataset, multilabel classification task, system components (BERT, CNN, and XGBoost), research method, evaluation method and experiment details. The general architecture for our two suggested strategies will also be illustrated in this section.

2.1. Dataset

We use Airy Rooms hotel reviews dataset in Indonesian language from Azhar *et al.* [2]. We divided the dataset for this study into train, valid, and test sets according to the standardization of IndoNLU documentation [20]. The dataset consists of 2,854 reviews, that was divided as follows: 2,283 for the train dataset, 286 for the test dataset and 285 for the validation dataset. Each review consists of text and a set of assigned labels. Table 1 presents the labels distribution of Airy Rooms hotel reviews dataset.

Based on the label distribution in Table 1, there are ten labels in this dataset: ac (*ac*), hot water (*air panas*), smell (*bau*), general (*general*), cleanliness (*kebersihan*), linen (*linen*), service (*service*), sunrise meal (*sunrise meal*), tv (*tv*) and Wi-Fi (*Wi-Fi*). We can also identify if the most frequently reviewed aspect was cleanliness (*kebersihan*), and the least reviewed aspect was the sunrise meal (*sunrise meal*).

Table 1. Labels distribution each aspect

Labels (Eng)	Labels (Indo)	Train	Valid	Test
AC	*AC*	469	59	53
Hot water	*Air panas*	361	55	41
Smell	*Bau*	372	38	42
General	*General*	260	32	43
Cleanliness	*Kebersihan*	933	128	128
Linen	*Linen*	670	95	89
Service	*Service*	634	74	79
Sunrise_meal	*Sunrise meal*	175	24	22
TV	*TV*	208	31	29
Wi-Fi	*Wi-Fi*	355	41	36

2.2. Multi-label classification task

Classification task is generally described as: "Given a train set made up of pairs (x_j, y_j), discover a function $f(x)$ that maps each attribute vector x_j, to its associated label y_j, with $j = 1, 2, 3, \ldots, n$, where the amount of training examples is n" [23]. In multi-label classification, each input sample may correspond to more than one labels. More specifically, for each input sample, there exist a set of labels "M" to which the input sample belongs [24]. Figure 1 presents an illustration of the multi-label text classification formulation for aspect classification.

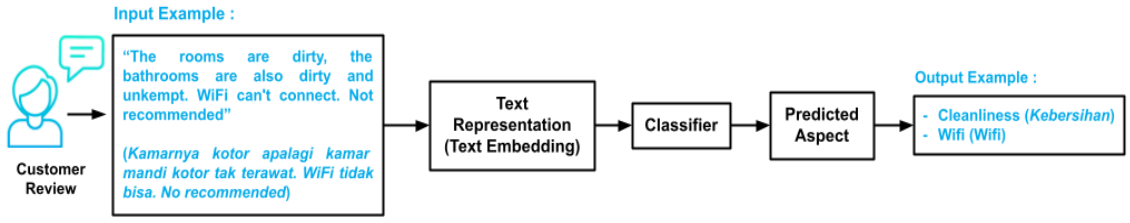

Figure 1. The general research process of multi-label text classification

The aim of this study is to predict or categorize the aspect category of the customer review in the Indonesian dataset of hotel customer reviews. Given a customer review, the text embedding for the review is initially generated. The embedding results are then inputted into the classifier that will produce the predictions of the aspect labels of the customer review. For example, given a hotel review displayed in Figure 1, two aspects are classified from the review: "cleanliness" (*kebersihan*) and "Wi-Fi" (*Wi-Fi*). More detailed explanation

about our methods, including the text representation and the classification methods, are explained in section 2.4.

2.3. System components

2.3.1. BERT

The current state-of-the-art for many natural language processing (NLP) applications is BERT, which stands for bidirectional encoder representations from transformers [14]. BERT is expected to learn a word's context right-to-left to predict the previous word or in left-to-right to predict the next word in a sequence [16]. There are various kinds of BERT models which have been developed, such as: m-BERT [14], distil-BERT [25], XLMRoBERTa [26], IndoBERT (base, lite, and large) from Wilie *et al.* [20], IndoBERT (base) and IndoBERTweet from Koto *et al.* [21], [27]. For more detailed explanation on each of these methods, please refer to the original paper. Table 2 summarizes the hyperparameter settings that were used in previous work to build all pre-trained language model BERT versions that are utilized in this work. Subscripts W and K in the table denotes the respective model was trained by Wilie *et al.* [20] and Koto *et al.* [21], respectively.

Multilingual BERT (M-BERT) is a single pre-trained language model on the concatenation of monolingual Wikipedia corpora from 104 languages, including Indonesian language [14]. As a result of the m-BERT model being pre-trained on high number of languages, it expands the applicability of this model, and researcher can use it to solve the task in various languages. The DistilBERT is a distilled variation of the BERT; it is smaller and operates faster. It is also capable of retaining 97% of BERT's ability to understand a language [25]. One of the multilingual model that has been trained on 100 different languages is called XLM-RoBERTa [26].

Just two years ago, a huge number of Indonesian corpus were used to pre-train the BERT model, and the resulting model called IndoBERT [20], [21] and were publicly available for research purpose. A lot of attention has been paid to the exploration of IndoBERT for several text processing task. The large-scale Indonesian dataset used to train IndoBERT by Willie *et al.* [20] was compiled from texts found on websites, news, blogs, and social media. This dataset consist of around 4 billion words, with around 250M sentences [20]. 220 million words from the Indonesian web corpus, news articles, and Wikipedia were used to train IndoBERT by Koto *et al.* [21]. Koto *et al.* [27] also released the IndoBERTweet, a BERT language model that has been pre-trained with 409M word tokens from Indonesian Twitter dataset.

In this study, we build our model for aspect classification using the IndoBERT pre-trained model developed by Wilie *et al.* [20] and Koto *et al.* [21], [27]. For our multi-label text classification task, we utilized IndoBERT using two strategies: i) using IndoBERT as feature representation only and ii) using IndoBERT as end-to-end model (i.e., it serves as feature representation as well as classifier). In addition, we also do further analysis on the comparability of our models' results with those using multilingual BERT, such as m-BERT, distil-BERT, and XLM-RoBERTa.

Table 2. Hyperparameter configurations for BERT

Model	Type	Embedding size	Hidden layers	Attention heads	Vocab size	Parameters
IndoBERT^{W}	base	768	12	12	30522	124.5 M
IndoBERT^{W}	lite	128	12	12	30000	11.7 M
IndoBERT^{W}	large	1024	24	16	30522	335.2 M
IndoBERT^{K}	base	768	12	12	31923	125 M
IndoBERTweet^{K}	base	768	12	12	31984	125 M
m-BERT	base	768	12	12	30000	110 M
distil-BERT	base	768	6	12	30522	66 M
XLM-RoBERTa	base	768	12	12	30522	125 M

2.3.2. CNN

It has been shown that CNN is one of the models that performs great for the multi-label text classification as investigated in [2], [11]–[13]. The type of convolution used in text-processing tasks is called one-dimensional convolution. It involves mapping the input text into a set of embedding vectors that correspond to the text's word order [28]. In order to extract indicative information, over the sequence of word embedding vectors the convolutional layer moves a sliding window of size k, while performing a linear transformation along with a non-linear activation function. The pooling layer selects only the information that is suitable for prediction for each window [28]. In this study, we used a single channel convolutional layer following [11], with rectified linear unit (ReLU) activation function. The model of CNN single is detailed in Figure 2.

The convolution window's length is determined by the kernel size. For this study, the kernel will slide along the input embedding and examine two words at a time because we used a kernel size of 2 (this essentially corresponding to the bigram features) [11].

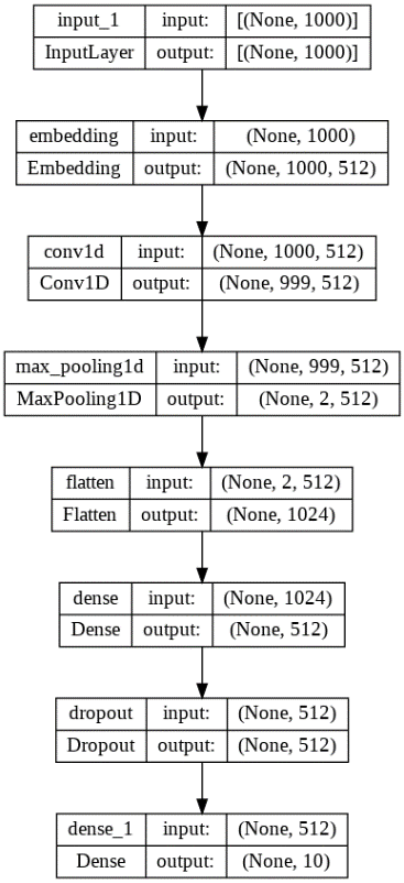

Figure 2. The architecture model of CNN single

2.3.3. XGBoost

An expanded variant of the gradient boosting ensemble method is called XGBoost, or extreme gradient boosting [29]. To create an efficient learning machine for the ensemble decision three approach, the gradient boosting technique successively combines the results of weak classifiers. The components of XGBoost are an optimization objective function, a parameter adjustment, and a learning model. It is feasible to carry out objective function optimization and reduce model complexity by optimizing the penalty function and minimizing the loss function [30].

2.4. Research method

The research method section will illustrate the overall architecture of our proposed strategies. We also explain in more detail our two proposed strategies for identifying all aspects contained in each customer review by performing multi-label text classification. The framework of our first and second strategies are illustrated respectively in Figures 3 and 4.

2.4.1. IndoBERT-CNN-XGBoost model

For the first strategy, we used IndoBERT as embedding, CNN as feature extraction method, and XGBoost as the classifier. Figure 3 depicts the research flow of this model. First, the dataset has to be transformed into the BERT input format. We preprocessed the input data using the BERT Tokenizer. Because the task that we used in this study is multi-label classification, every sentence must have a [SEP] token added at the end and a [CLS] token added at the beginning. To fit the maximum sequence length of 128 tokens, each sentence must be truncated or padded. Besides that, we used the 'attention_mask' which consists of $'0'$ (denotes not token) and $'1'$ (denotes token). From this step we got the 'input_ids' and 'attention_mask' for each sentence.

Suppose that we used the IndoBERT embedding from Wilie *et al.* [20], which has a vocabulary size of 30522 and a dimension of 768. We got the 30522×768 size of embedding matrix. Then, each input's 'input_ids' and 'attention_mask' are embedded by the embedding layer. An embedding vector e_i, where $e_i \in R^d$ is used to represent each word and d stands for the word embedding dimension. To represent the sentence as a whole, the word vector representations of each word is then concatenated. We can define the sentence

representation as $S = e_{1:m} = e_1 \oplus e_2 \oplus ... \oplus e_m$, where m is defined as the input text's maximum length [11]. Input for a single sentence is thus represented as $m \times d$ matrix [15].

After we have obtained a vector representation for each review, then we used it as a feature to train the CNN model to get the refined version of the text feature. To solve the multi-label text classification, we replace the CNN-trained model's output layer with XGBoost as the top-level classifier. The objective here is we want to extract the text matrix's refined version produced by CNN and utilize it as a feature to train the XGBoost classifier. In this work, we also experimented with a few other classifiers, such as random forest and naïve Bayes algorithms.

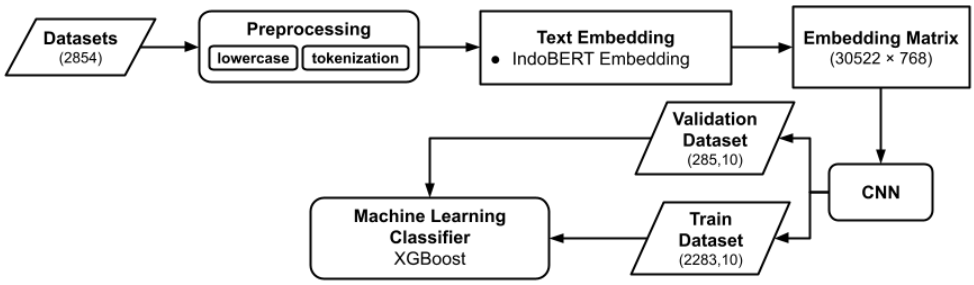

Figure 3. The research process of our first strategy model

We use problem transformation method classifier chain (CC) to perform multi-label classification using machine learning classifier [8], [15]. From our preliminary experiments, the results of CC strategy is better than BR strategy, which is consistent to previous work [5], [31]. The CC strategy is effective because it can overcome the problem of label dependency [6]. Therefore, in our experiment in this work, we use CC strategy to conduct problem transformation of our data for multi-label classification.

2.4.2. IndoBERT end-to-end model

For the second strategy, we used IndoBERT to build end-to-end model for multi-label classification. Here, IndoBERT is used as text embedding as well as classifier. Two versions of IndoBERT were used: IndoBERT that was pretrained by Wilie *et al.* [20], and IndoBERT and IndoBERTweet that was pretrained by Koto *et al.* [21], [27]. In our experiment, we also compared our results with those using multilingual BERT, such as m-BERT [14], distil-BERT [25] and XLM-RoBERTa [26] that were trained in previous work using multilingual corpus (the different configurations of each model have been explained earlier in section 2.3.1). The research process of the second strategy is illustrated in Figure 4.

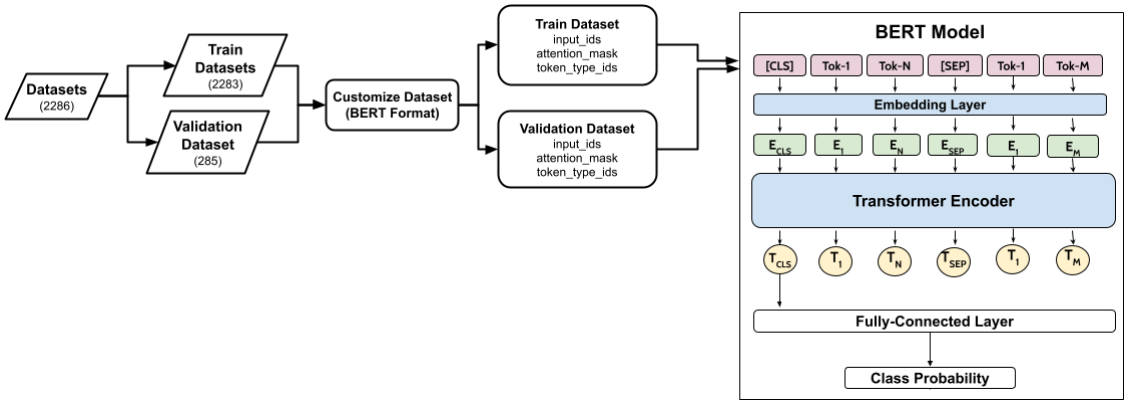

Figure 4. The research process for our second strategy model

As explained in the section 2.4.1 above, we need to adjust the dataset into BERT input format. After obtaining the appropriate data format, we must add a classifier layer on top of the model and allowing the BERT model to do the multi-label classification. To define the final value of the multi-label classification

process, the [CLS] output from the final hidden layer, which is represented as a vector with dimensions of 768, will be entered through a fully connected layer and then calculated using the sigmoid activation function. The outputs from sigmoid activation function give a value between 0 and 1, which represents the probabilities of each of the 10 predicted aspect labels. In this study, the predicted label output is decided using a threshold value of 0.5 [32].

2.5. Evaluation method

In this study, we use micro F1-score, hamming loss and accuracy for the model evaluation. Micro F1-score calculated using the value of false negative (FN), true positive (TP) and false positive (FP). The micro F1-score which obtained from the average number of calculations from each aspect is defined as (1) [33].

$$Micro - F1 = \frac{\sum_{j=1}^{L} 2 \times TP_j}{\sum_{j=1}^{L} (2 \times FP_j + FN_j + TP_j)} \quad (1)$$

where the total number of aspects is L, the aspect index is j, FP value on an aspect with index j is FP_j, FN value on an aspect with index j is FN_j and TP value on an aspect with index j is TP_j.

The mismatches between the actual and the predicted aspect labels are measured using hamming loss (HL), which is determined as (2) [15].

$$HL = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} 1_{(y_{i,k} \neq \hat{y}_{i,k})} \quad (2)$$

where $y_{i,k}$ is the actual aspect label and $\hat{y}_{i,k}$ denotes its predicted aspect label. K is the total number of aspects and N is the number of sample size.

Accuracy is the probability of a label that has the same value between actual data and predictive data. The (3) is the formula for accuracy in multi-label classification [6].

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \frac{|\hat{y}_i \cap y_i|}{|\hat{y}_i \cup y_i|} \quad (3)$$

where the total number of data is N, the actual aspect label set is y_i and prediction aspect label is $\hat{y}_i$.

2.6. Experiment details

In our experiment, we use several baseline methods such as: random forest, naïve Bayes, XGBoost, CNN, CNN-RandomForest, CNN-Naivebayes, and CNN-XGBoost. For each of these models, we also experimented with the variation of text embeddings: Word2Vec and IndoBERT. The CNN-XGBoost model with Word2Vec embedding was actually the best performing method in Azhar *et al.* [2], while the CNN model with IndoBERT embedding was the best performing method in Khasanah and Krisnadhi [11]. For our second model, we experimented with some versions of IndoBERT, such as: IndoBERT (lite), IndoBERT (base), and IndoBERT (large) from Willie *et al.* [20]; and IndoBERT and IndoBERTweet from Koto *et al.* [21], [27]. Further, a comparison with some multilingual BERT, such as m-BERT, distilBERT, and XLM-RoBERTa, was also performed. For a summary of the configurations of each BERT model, we have detailed them in Table 2.

The model architecture was created using Python 3.7 and the model was trained using a NVidia Tesla T4 with single core. For the hyperparameter of the CNN, we follow the single CNN architecture and settings from Khasanah and Krisnadhi [11], such as using Adam optimizer, batch size of 64, the dimension of the input 512, dropout rate of 0.5, max epoch of 70, learning rate of 0.001 and kernel size of 2. For the machine learning top classifier, we used random forest, XGBoost and naïve Bayes with the CC strategy. For the hyperparameter settings of BERT model, we followed Devlin *et al.* [14] recommendations, such as using dropout probability rate of 0.1, learning rate of 2e-5 and use Adam optimizer. We set the train and valid batch size of 32, and maximum input length of 128. For the baseline methods using Word2Vec embedding, the training parameters are a window size of 5 and vector size of 512.

3. RESULTS AND DISCUSSION

3.1. The results of our first model: IndoBERT-CNN-XGBoost

Based on Table 3, the results show that deep learning model CNN is more effective than machine learning models, i.e., random forest, naive Bayes and XGBoost, for aspect classification task. Further combining deep learning model CNN with machine learning methods can increase the effectiveness of the model. It appears from the results of CNN-random forest, CNN-naive Bayes, and CNN-XGBoost models that outperform the results of CNN model. This finding is consistent with the one reported in [2]. It indicates that

the use of CNN as feature extraction method results in more refined features that enable the machine learning classifier to classify the aspects from a review text more accurately.

Among all models, CNN-XGBoost models consistently gain the best results in terms of micro F1-score, hamming loss, and accuracy in each type of text embedding. Our model using IndoBERT as text embedding for CNN-XGBoost model is shown to significantly outperform the CNN-XGBoost model of Azhar *et al.* [2] that uses Word2Vec embedding, by achieving the micro F1-Score of 0.8992, hamming loss of 0.0404 and accuracy of 0.7228. This finding shows the effectiveness of our first model in using IndoBERT as text representation in CNN-XGBoost model. These results are also consistent with the IndoBERT in IndoNLU benchmark results obtained by Wilie *et al.* [20], in which one of their findings is the contextualized pretrained models significantly outperformed the static word-embedding-based models, the advantage of contextualized word embeddings over static word embeddings is illustrated by this [20]. Note that the scores of Word2Vec-CNN-XGBoost models presented in the original paper of Azhar *et al.* [2] are slightly different from those displayed in our table because they used a different data split from ours (here, we utilize the distribution of the Airy Room dataset split provided by IndoNLU [20]). However, we have ensured to follow similar hyperparameter settings used by Azhar *et al.* [2] to generate the results of all Word2Vec-CNN-machine_learning variations in our table.

Table 3. The comparison results of using text embedding from Word2Vec and IndoBERT

Embedding	Model	Micro F1-score	Hamming loss	Accuracy
Word2Vec	Random forest	0.4689	0.1589	0.2070
	Naïve Bayes	0.4374	0.3782	0.0561
	XGBoost	0.5390	0.1494	0.2456
	CNN	0.6258	0.1087	0.3322
	CNN-random forest	0.8675	0.0512	0.6316
	CNN-naïve Bayes	0.7273	0.1147	0.3193
	CNN-XGBoost (Azhar *et al.* [2])	0.8743	0.0491	0.6386
$IndoBERT^{W}$	Random forest	0.3938	0.1642	0.1614
	Naïve Bayes	0.5020	0.3063	0.0737
	XGBoost	0.5987	0.1298	0.2877
	CNN (Khasanah and Krisnadhi [11])	0.6244	0.1122	0.3508
	CNN-random forest	0.8877	0.0442	0.6982
	CNN-naïve Bayes	0.7797	0.0807	0.5263
	CNN-XGBoost (ours)	0.8992	0.0404	0.7228

3.2. The results of our second model: end-to-end IndoBERT

We used five types of monolingual models, IndoBERT, to do the multi-label text classification. The models are IndoBERT (base, lite, and large) from Wilie *et al.* [20], IndoBERT (base) [21] and IndoBERTweet from Koto *et al.* [27]. Table 4 presents the findings of these models.

Table 4. The comparison results of using IndoBERT model

Embedding	Model	Micro F1-score	Hamming loss	Accuracy
IndoBERT	$IndoBERT^{W}$-base	0.92700	0.02999	0.76098
	$IndoBERT^{W}$-lite	0.86334	0.05306	0.61357
	$IndoBERT^{W}$-large	0.92828	0.02953	0.76112
	$IndoBERT^{K}$-base	0.91796	0.03345	0.73947
	$IndoBERTweet^{K}$-base	0.92123	0.03207	0.74680

Among all monolingual models, IndoBERT, used in this experiment, the IndoBERT-large model by Wilie *et al.* [20], achieves the best value of micro F1-score 0.92828, hamming loss 0.02953 and accuracy 0.76112. This can be explained because IndoBERT-large has a much bigger number of parameters in its neural network architecture, which is almost three times bigger than the number of parameters for IndoBERT-base, IndoBERTK and IndoBERTweetK Table 2. This makes IndoBERT-large is more accurate in capturing the structure and semantics of the data. This result shows an improvement in accuracy to the baseline Word2Vec-CNN-XGBoost model by up to 19.19%.

Besides that, we also conclude that in general, the performance of our second models using end-to end IndoBERT model are significantly better than the performance results of our first model presented in section 3.1. earlier. This indicates that using IndoBERT as an end-to-end model by fine-tuning the initial pre-trained model with our specific task is more effective than using it as text embedding only. This result confirms the effectiveness of IndoBERT to directly solve various text processing tasks [21].

Beside using the monolingual BERT model, we also conduct a comparative experiment using multilingual BERT models, the results are presented in Table 5. The common multilingual models, m-BERT [14], distil-BERT [25], and XLM-RoBERTa [26], were used.

Based on the findings in Table 5, the m-BERT outperforms the other two multilingual pre-trained language models, with micro F1-score of 0.90901, hamming loss of 0.03705, and accuracy of 0.71768. We might infer from the findings of Tables 4 and 5, that the results of monolingual BERT are still more effective than the multilingual BERT. This is due to the monolingual model is trained using a single language only, so that the model can be more focused and accurate in learning the characteristic of the language on the training data. However, in other languages when the monolingual BERT model is not available, we argue that using the multilingual BERT model is suggested, especially m-BERT since its results are still more effective compared to the results of machine learning or deep learning models CNN using Word2Vec displayed in Table 3.

Table 5. The results of multilingual BERT model

Embedding	Model	Micro F1-score	Hamming loss	Accuracy
BERT	m-BERT-base	0.90901	0.03705	0.71768
	distil-BERT-base	0.87896	0.04817	0.64533
	XLM-RoBERTa-base	0.77074	0.08357	0.47192

3.3. Qualitative analysis

Based on the findings of the model evaluation that was done in the sections before, the IndoBERT-large model gives the best performance results. Therefore, we carried out the qualitative analysis using the evaluation results of that model. The qualitative analysis offers methods for assessing, analyzing, and interpreting the significant patterns in the data.

First, we analyze the results from the IndoBERT-large model using a confusion matrix. The multi-label text classification model was tested using 286 data and the analysis was carried out for each aspect. Figure 5 displays the confusion matrix's results. The confusion matrix, which compares the predicted aspect values with the actual aspect values, is used to determine how effective the classification model is. The value is represented by TN, TP, FN and FP.

Figure 5. Multi-label confusion matrix of IndoBERT-large

Based on the confusion matrix in Figure 5, it can be identified that the TP and TN values are generally bigger than the FP and FN values, and it can be seen from the color difference in the confusion matrix, where the TP and TN values generally have lighter colors when compared to FP and FN. Based on the confusion matrix in Figure 5, we can also obtain the accuracy and micro F1 score values for each aspect, which are shown in Table 6.

Based on Table 6, it can be identified that the 'Wi-Fi' is the most accurately predicted aspect with accuracy of 0.9964 and micro F1-score of 0.9875. It can be happened because when a review discusses the 'Wi-Fi' aspect, customers will tend to explicitly mention the word 'Wi-Fi'. So as a result, the model will better

understand and more accurately predict that aspect. For example: *"hotelnya bagus, makanan cukup, Wi-Fi kurang merata" ("the hotel is good, the food is enough, the Wi-Fi is uneven"*). In that review, the 'Wi-Fi' aspect is mentioned explicitly. Furthermore, the most inaccurate aspect is 'cleanliness', with accuracy of 0.9257 and micro F1-score of 0.9181. Because in some cases, the reviews that discuss the aspect of 'cleanliness', do not explicitly mentioned the word 'cleanliness' or *'kebersihan'* and sometimes the customers associate it with other aspects or elements. For example: *"banyak nyamuk, selebihnya oke2 aja" ("lots of mosquitoes, the rest is quite good"*). In that review, the 'cleanliness' aspect is not explicitly mentioned, but the review associates the aspect of 'cleanliness' with the condition of too many mosquitoes.

From Table 6, we can also see that in some cases, the IndoBERT-large model still could not correctly predict all aspects in a review completely. There are some reviews in which the model can only predict some of the aspects correctly. Based on our further analysis, the prediction results have 226 reviews that are correctly classified and 60 reviews that are not correctly classified in complete. The misclassified results are caused by the system that fails to understand the meaning of the review. Table 7 shows two examples of reviews whose aspects could not be correctly classified in complete. In test data (1), the model can predict correctly two out of three aspects contained in the review. In test data (2), the model cannot predict the only one aspect contained in the review.

In test data (1), we found that the aspect *'kebersihan'* (cleanliness) is not identified in the prediction results, which can be happened because the model misunderstands the data. The model cannot identify that word *'kutu'* (bedbugs) is related to the aspect *'kebersihan'* (cleanliness). Next, in test data (2), the aspect category *'kebersihan'* (cleanliness) also cannot be detected in the prediction results. We analyzed that this is caused by the typo word *"kabersihan"* which should be written *"kebersihan".* This makes the model could not capture the meaning of the review well.

Table 6. The result of IndoBERT-large for each aspect

Aspects (Eng)	Aspects (Indo)	TP	TN	FP	FN	Accuracy	Micro F1-score
AC	*AC*	5794	22400	106	200	0.9893	0.9743
Hot water	*Air panas*	5372	22950	128	50	0.9938	0.9837
Smell	*Bau*	3576	23929	224	771	0.9651	0.8779
General	*General*	2631	24679	569	621	0.9582	0.8156
Cleanliness	*Kebersihan*	11860	14523	940	1177	0.9257	0.9181
Linen	*Linen*	8753	17961	747	1039	0.9373	0.9074
Service	*Service*	7097	20217	303	883	0.9584	0.9229
Sunrise meal	*Sunrise meal*	2212	25913	188	187	0.9868	0.9219
TV	*TV*	3089	25230	11	170	0.9936	0.9715
Wi-Fi	*Wi-Fi*	4096	24300	4	100	**0.9964**	**0.9875**

Table 7. The example of misclassified aspect labels

Test data (1)	Label	AC *(AC)*	Hot water *(Air panas)*	Smell *(Bau)*	General *(General)*	Cleanliness *(Kebersihan)*	Linen *(Linen)*	Service *(Service)*	Sunrise meal *(Sunrise meal)*	TV *(TV)*	Wi-Fi (Wi-Fi)
Kasur ada kutu nya, dan badan saya jadi gatal gatal. dan AC sama sekali tdk dingin	Actual	1	0	0	0	1	1	0	0	0	0
(The mattress has bedbugs, and my body itches... And the AC is not cold at all)	Prediction	1	0	0	0	0	1	0	0	0	0
Test data (2)	Label	AC *(AC)*	Hot water *(Air panas)*	Smell *(Bau)*	General *(General)*	Cleanliness *(Kebersihan)*	Linen *(Linen)*	Service *(Service)*	Sunrise meal *(Sunrise meal)*	TV *(TV)*	Wi-Fi (Wi-Fi)
lumayan untuk harga segitu... kabersihan tolong ditingkatkan	Actual	0	0	0	0	1	0	0	0	0	0
(Not bad for that price... cleanliness please improve)	Prediction	0	0	0	0	0	0	0	0	0	0

4. CONCLUSION

In this study, we proposed two strategies using monolingual pre-trained language model BERT on Indonesian language (i.e., IndoBERT) for identifying aspects in the customer review dataset, by performing multi-label text classification. First, we used IndoBERT as text embedding for CNN-XGBoost classifier.

Second, we used the IndoBERT as text embedding as well as the classifier in an end-to-end model. Moreover, as part of an in-depth examination of this study, a multilingual BERT model was also exploited. According to the results of our studies, our proposed strategies significantly outperform some of the state-of-the-art baselines. The use of IndoBERT as embedding for the CNN-XGBoost model give some improvement over some machine learning and deep learning models, with micro F1-Score of 0.8992, hamming loss of 0.0404 and accuracy of 0.7228. IndoBERT as contextualized pre-trained models can give better text representation when compared to the context-independent word-embedding model like Word2Vec. Next, the use of IndoBERT models as embedding as well as classifier to solve multi-label text classification can further significantly enhance our first model which uses IndoBERT for text embedding only. The IndoBERT-large outperformed the other IndoBERT models, according to the results, with micro F1-Score of 0.92828, hamming loss of 0.02953 and accuracy of 0.76112. It has been demonstrated that this approach may improve the accuracy of a Word2Vec-CNN-XGBoost baseline by up to 19.19%. When we compared IndoBERT with the multilingual BERT (m-BERT, distil-BERT and XLM-RoBERTa), we found that the monolingual BERT is slightly more accurate than multilingual BERT. Here, the best-performing monolingual BERT model (i.e., IndoBERT-large) gains 6% higher accuracy compared to the best-performing multilingual BERT model (i.e., m-BERT-base). Some suggestions that can be conducted for future work, include: the use of another alternative architectures such as recurrent neural networks (RNNs) or other transformer-based architectures. Other than that, the imbalanced label dataset can be handled using a good and complex synthetic oversampling technique.

ACKNOWLEDGEMENTS

This research was funded by the Directorate of Research and Development, Universitas Indonesia, under Hibah PUTI Pascasarjana 2022 (Grant No. NKB-103/UN2.RST/HKP.05.00/2022).

REFERENCES

[1] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 27–35, doi: 10.3115/v1/S14-2004.

[2] A. N. Azhar, M. L. Khodra, and A. P. Sutiono, "Multi-label aspect categorization with convolutional neural networks and extreme gradient boosting," in *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, Jul. 2019, pp. 35–40, doi: 10.1109/ICEEI47359.2019.8988898.

[3] E. Deniz, H. Erbay, and M. Coşar, "Multi-label classification of E-commerce customer reviews via machine learning," *Axioms*, vol. 11, no. 9, Aug. 2022, doi: 10.3390/axioms11090436.

[4] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu, "ML-Net: multi-label classification of biomedical texts with deep neural networks," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1279–1285, Nov. 2019, doi: 10.1093/jamia/ocz085.

[5] A. D. Asti, I. Budi, and M. O. Ibrohim, "Multi-label classification for hate speech and abusive language in Indonesian-local languages," in *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2021, pp. 1–6, doi: 10.1109/ICACSIS53237.2021.9631316.

[6] R. Hendrawan, Adiwijaya, and S. Al Faraby, "Multilabel classification of Hate speech and abusive words on Indonesian Twitter social media," in *2020 International Conference on Data Science and Its Applications (ICoDSA)*, Aug. 2020, pp. 1–7, doi: 10.1109/ICoDSA50139.2020.9212962.

[7] E. Montañes, R. Senge, J. Barranquero, J. R. Quevedo, J. José del Coz, and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognition*, vol. 47, no. 3, pp. 1494–1508, 2014, doi: 10.1016/j.patcog.2013.09.029.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, Dec. 2011, doi: 10.1007/s10994-011-5256-5.

[9] R. Y. Rumagit, "Multilabel classification for toxic comments in Indonesian," *Engineering, MAthematics and Computer Science (EMACS) Journal*, vol. 2, no. 1, pp. 29–34, Jan. 2020, doi: 10.21512/emacsjournal.v2i1.6256.

[10] R. A. Ilma, S. Hadi, and A. Helen, "Twitter's hate speech multi-label classification using bidirectional long short-term memory (BiLSTM) method," in *2021 International Conference on Artificial Intelligence and Big Data Analytics*, Oct. 2021, pp. 93–99, doi: 10.1109/ICAIBDA53487.2021.9689767.

[11] I. N. Khasanah and A. A. Krisnadhi, "Extreme multilabel text classification on Indonesian tax court ruling using single channel CNN and IndoBERT embedding," in *2021 6th International Workshop on Big Data and Information Security (IWBIS)*, Oct. 2021, pp. 59–66, doi: 10.1109/IWBIS53353.2021.9631855.

[12] S. R. Anggraeni, N. A. Ranggianto, I. Ghozali, C. Fatichah, and D. Purwitasari, "Deep learning approaches for multi-label incidents classification from Twitter textual information," *Journal of Information Systems Engineering and Business Intelligence*, vol. 8, no. 1, pp. 31–41, Apr. 2022, doi: 10.20473/jisebi.8.1.31-41.

[13] G. A. Neruda and E. Winarko, "Traffic event detection from Twitter using a combination of CNN and BERT," in *2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2021, pp. 1–7, doi: 10.1109/ICACSIS53237.2021.9631334.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[15] L. Cai, Y. Song, T. Liu, and K. Zhang, "A hybrid BERT model that incorporates label semantics via adjustive attention for multi-label text classification," *IEEE Access*, vol. 8, pp. 152183–152192, 2020, doi: 10.1109/ACCESS.2020.3017382.

[16] L. F. Simanjuntak, R. Mahendra, and E. Yulianti, "We know you are living in Bali: location prediction of Twitter users using BERT language model," *Big Data and Cognitive Computing*, vol. 6, no. 3, Jul. 2022, doi: 10.3390/bdcc6030077.

[17] J. A. Alzubi, R. Jain, A. Singh, P. Parwekar, and M. Gupta, "COBERT: COVID-19 question answering system using BERT," *Arabian Journal for Science and Engineering*, pp. 1–11, Jun. 2021, doi: 10.1007/s13369-021-05810-5.
[18] N. Liu, Q. Hu, H. Xu, X. Xu, and M. Chen, "Med-BERT: A pretraining framework for medical records named entity recognition," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5600–5608, Aug. 2022, doi: 10.1109/TII.2021.3131180.
[19] J. Dong, F. He, Y. Guo, and H. Zhang, "A commodity review sentiment analysis based on BERT-CNN model," in *5th International Conference on Computer and Communication Systems (ICCCS)*, 2020, pp. 143–147, doi: 10.1109/ICCCS49078.2020.9118434.
[20] B. Wilie *et al.*, "IndoNLU: benchmark and resources for evaluating Indonesian natural language understanding," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 843–857.
[21] F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, "IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
[22] E. Yulianti, A. Kurnia, M. Adriani, and Y. S. Duto, "Normalisation of Indonesian-english code-mixed text and its effect on emotion classification," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, 2021, doi: 10.14569/IJACSA.2021.0121177.
[23] G. B. Larrain, N. Rojas-Morales, P. D. O. Jeneral, and N. G. Rogel, "Towards a classifier ensamble to prevent burnout syndrome on University students," in *2022 41st International Conference of the Chilean Computer Science Society (SCCC)*, Nov. 2022, pp. 1–8, doi: 10.1109/SCCC57464.2022.10000313.
[24] A. H. Ombabi, O. Lazzez, W. Ouarda, and A. M. Alimi, "Deep learning framework based on Word2Vec and CNN for users interests classification," in *2017 Sudan Conference on Computer Science and Information Technology (SCCSIT)*, Nov. 2017, pp. 1–7, doi: 10.1109/SCCSIT.2017.8293054.
[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *Computing Research Repository (CoRR)*, 2019.
[26] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.
[27] F. Koto, J. H. Lau, and T. Baldwin, "IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary Initialization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 10660–10668, doi: 10.18653/v1/2021.emnlp-main.833.
[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
[29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[30] E. Shang, X. Liu, H. Wang, Y. Rong, and Y. Liu, "Research on the application of artificial intelligence and distributed parallel computing in archives classification," in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Dec. 2019, pp. 1267–1271, doi: 10.1109/IAEAC47372.2019.8997992.
[31] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014, doi: 10.1109/TKDE.2013.39.
[32] H. Fallah, P. Bellot, E. Bruno, and E. Murisasco, "Adapting transformers for multi-label text classification," in *CIRCLE (Joint Conference of the Information Retrieval Communities in Europe)*, 2022, pp. 1–18.
[33] M. A. Abdurrazzaq, G. A. P. Saptawati, and Y. Rusmawati, "MAGNET architecture optimization on multi-label text classification," in *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, Sep. 2021, pp. 1–6, doi: 10.1109/ICAICTA53211.2021.9640263.

BIOGRAPHIES OF AUTHORS

Nuzulul Khairu Nissa received B.Math. degree from Diponegoro University in 2019. She is currently pursuing a Master of Computer Science in University of Indonesia. Her research interests are related to machine learning and natural language processing. She can be contacted at email: nuzulul.khairu@ui.ac.id.

Evi Yulianti is a lecturer and researcher at Faculty of Computer Science, Universitas Indonesia. She received the B.Comp.Sc. degree from the Universitas Indonesia in 2010, the dual M.Comp.Sc. degree from Universitas Indonesia and Royal Melbourne Institute of Technology University in 2013, and the Ph.D. degree from Royal Melbourne Institute of Technology University in 2018. Her research interests include information retrieval and natural language processing. She can be contacted at email: evi.y@cs.ui.ac.id.