IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 3, June, 2023, pp. 733-743
Journal homepage: http://iieta.org/journals/ria

Fine-Tuned IndoBERT Based Model and Data Augmentation for Indonesian Language Paraphrase Identification

Benedictus Visto Kartika, Martin Jason Alfredo, Gede Putra Kusuma*

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: inegara@binus.edu

https://doi.org/10.18280/ria.370322

Received: 18 February 2023
Accepted: 9 March 2023

Keywords:
IndoBERT model, Data Augmentation, entailment method, deep neural network, Paraphrase Identification

ABSTRACT

Natural Language Processing tasks in the Indonesian language have recently flourished thanks to the research of IndoBERT and its benchmark. Despite being the fourth most used language over the internet, the Indonesian language NLP task still has some gaps, one of them being the Paraphrase Identification task. In order to solve this gap, we proposed a fine-tuned IndoBERT based model for Paraphrase Identification. Several methods have been researched in this paper from setting the baseline, Data Augmentation, fine-tune the classifier, and task reformulation. Besides the model, this paper also provides the Paraphrase Identification dataset in Indonesian language. The baseline IndoBERT model performs well, it proves that IndoBERT is one of the fittest methods to use. We then researched further and proposed a Modified Easy Data Augmentation that augments very well in this task and potentially on other NLP tasks. We compared traditional machine learning classifiers with deep neural network classifiers, fine-tuned them, and selected the best classifier for this task. Furthermore, we tried entailment task reformulation. The Modified EDA shows a successful augmentation that increases both accuracy and F1 score for all the models. A slightly complex upgrade for the classifier also increased the performance while maintaining a reasonable training time.

1. INTRODUCTION

Indonesian language is the fourth most frequently used language having over 200 million native speakers yet is still underrepresented in NLP (Natural Language Processing) [1] having only around 200 papers related to NLP in 2020 compared to English language which has over 5000 papers [2]. The root causes of this are the lack of annotated datasets, sparsity of language resources, and lack of resource standardization [3]. NLP is one of the most important things to help machines understand and resolve ambiguity in human language. As of now, there are a lot of tasks that NLP can be used on such as speech recognition which can be used to recognize spoken language into text which can be useful in application such as voice assistants and dictation software, sentiment analysis which can be used to analyze customer feedback and social media post to determine the quality of a product or service, machine translation which can be used to translate text from one language to another, question answering which can be used to build chatbots that can understand and answer question in natural language, and so much more.

IndoLEM as the benchmark of Indonesian NLP tasks and IndoBERT as pretrained transformers Indonesian NLP model were introduced as an attempt to redress the situation for Indonesian [3]. Seven NLP tasks and eight sub-datasets are included in the comprehensive dataset known as IndoLEM. Wilie et al. [1] proposed a new benchmark for Indonesian NLP called IndoNLU. that consists of 12 tasks in four main scopes of sentence labeling and classification. It also comes with the new pre-trained IndoBERT which is trained with the Indo4B Dataset at over 4B words with around 250M sentences. It shows that IndoNLU IndoBERT is trained on more data corpus than IndoLEM IndoBERT and shows overall better performance, thus in this research, the author uses IndoNLU IndoBERT. While IndoLEM and IndoNLU has contributed a lot to Indonesian NLP in creating a comprehensive dataset, there are still a lot of gaps in the development of Indonesian NLP compared to English with one of the examples being paraphrase detection.

Paraphrase Identification (PI) is important since it helps with a variety of NLP activities, including text summarization, document clustering, query response, inference of natural language, knowledge retrieval, plagiarism detection, and text simplification [4]. Duplicate Question Identification (DQI), one of the most well-known applications of PI, can increase the processing speed and precision of large-scale community question-answering and automatic QA systems, as well as avoid the creation of duplicate questions with the purpose of identifying whether the paired questions are semantically equivalent. Over 400,000 question pairs make up the Quora Question Pairs (QQP) dataset from GLUE Benchmark [5], each of which is labeled with a binary value showing whether the two questions are paraphrases of each other. BERT [6] and other machine learning frameworks have already shown their performance on QQP. We create the PI task dataset by translating the QQP dataset to the Indonesian language using Google Translate API. There are some preprocessing techniques that help in NLP tasks, one of them Easy Data Augmentation (EDA) [7]. From several studies that have been

reviewed related to the implementation of EDA, the effect on the classification performance of several different datasets mostly resulted in an improvement compared to not applying any augmentation at all. The problem is, some of the pre-trained models can't keep the meaning of a sentence label while EDA is being implemented. Natasya's Modified EDA [8] attempts to solve this problem. This problem can be solved by comparing the context of the word before implementing EDA which focused on Part of Speech (POS) Tagging integration and word similarity. The author proposes a fine-tuned Modified EDA with an enhancement inspired by Natasya's Modified EDA.

In recent years, researchers experimented on training NLP model in a few-shots. One of the best models is Entailment as Few-shot Learner (EFL) [9]. EFL has a key idea to help small LM's into better few-shot learners and the key idea of the approach is to reformulate NLP tasks into entailment ones. In EFL Paper, it shows that when the EFL model is implemented on the whole dataset, it increases both accuracy and F1 score, thus this inspires us to research on entailment reformulation task.

In this paper, we experiment various methods to identify paraphrases for the Indonesian language based on the IndoBERT Base model [1]. The dataset used will be QQP GLUE which has been converted into Indonesian language. To improve its accuracy from the baseline IndoBERT original model, we first created a Modified EDA which then applied to the dataset. Second, we research on a more complex classifier such as ANN, LSTM [10], BiLSTM, GRU [11], BiGRU, SpinalNet [12], XGB, SVM, and RF on top of the IndoBERT token. Last, we experiment on the use of the entailment method [9]. These augmented and complex classifier experiments both show increment on the accuracy and F1 score. But on the other hand, the entailment method doesn't have a significant impact on both the accuracy and F1 score in either the original or augmented dataset.

In this paper, our contributions are:

1-Setting up baseline for Paraphrase Identification task in Indonesian language based on IndoBERT.

2-Compares classifier from traditional machine learning up to deep neural network (DNN) classifier on top of IndoBERT tokenizer.

3-Data Augmentation based on Easy Data Augmentation (EDA) that has been modified to fit the best for augmenting Duplicate Question Identification (DQI). It increases the F1 score and accuracy for almost all the tested scenarios. Most of the Modified EDA function augmentation can also be reused for generic NLP tasks.

4-Experiment on entailment task reformulation which gives slightly increased accuracy from the baseline.

5-Prepare Paraphrase Identification dataset based on QQP GLUE in English which is then translated to Indonesian language.

We present a fine-tuned classifier IndoBERT based model for Paraphrase Duplicate Question Identification with the augmentations of Modified EDA. We first introduce the background and related works in Section II. We then introduce the framework and architecture in Section III, which includes Modified EDA details, IndoBERT Tokenizer, and comparison of DNN classifiers. Section IV shows the experiment's detail, hyperparameter tuning, and results. Section V concludes this paper and shows further research that can be done.

2. RELATED WORKS

Two research studies produced two different IndoBERT. The first one is IndoBERT with IndoLEM [3] as the benchmark. IndoLEM IndoBERT was trained at over 220M words which result comes from three main sources which is Indonesian Wikipedia, news articles from Kompas, Tempo, Liputan6, and Indonesian Web Corpus. The IndoLEM benchmark has 3 scopes of tasks which is Morpho-syntax and Sequence Labeling Tasks, Semantic Tasks, and Discourse Coherence Tasks. On morpho-syntax and sequence labeling tasks it consists of part-of-speech tagging, named entity recognition, and dependency parsing. On semantic tasks it consists of sentiment analysis and summarization. On discourse coherence tasks, it consists of next tweet prediction and tweet ordering. IndoBERT mostly won as the best model compared to the other baseline model on this IndoLEM benchmark tasks.

The second one is IndoBERT with IndoNLU [1] as the benchmark. IndoNLU IndoBERT was trained with the Indo4B Dataset at over 4B words with around 250M sentences. IndoNLU consists of 12 tasks in four main scopes: Single-Sentence Classification Tasks, Sentence-Pair Classification Task, and Single-Sentence Sequence Labeling Tasks, and Sentence-Pair Sequence Labeling Task. On single-sentence classification tasks it consists of EmoT an emotion classification task, SmSA a sentiment analysis task, CASA an aspect-based sentiment analysis task, and HoASA an aspect-based sentiment analysis task. On sentence-pair classification task it consists of WReTE the Wiki Revision Edits Textual Entailment dataset. On single-sentence sequence labeling tasks it consists of POSP a part-of-speech tagging task, BaPOS a POS tagging task, TermA a span-extraction task, KEPS a key phrase extraction task, NERGit a named entity recognition task, and NERP a named entity recognition task. On sentence-pair sequence labeling task it consists of FacQA a question answering task. This IndoBERT also mostly outperforms other baseline models.

These two IndoBERT models are built both based on BERT-Base (uncased) [6] where it has 12 attention transformers layers and heads with 768 dimensions of embedding. Both follow the same pre-training methods which is Masked Language Modelling (MLM) and Next Sentence Prediction (NSP) as BERT. The differences between these two IndoLEM IndoBERT [3] and IndoNLU IndoBERT [1] are the datasets in where they are pre-trained and the benchmark where they are fine-tuned which are explained above. These IndoBERT models is the state-of-the-art for Indonesian NLP tasks. As IndoNLU IndoBERT is trained on a larger dataset than IndoLEM IndoBERT, tested on a wider benchmark, and shown better performance, thus the author is inspired to modify and fine-tune the IndoNLU IndoBERT in this research. These two benchmarks are missing what one of the GLUE tasks have which is Paraphrase Identification, thus the author decided to do IndoBERT on Paraphrase Identification task.

Quora Question Pairs (QQP) is one of the GLUE [5] Benchmark tasks in a scope of paraphrase identification task. In QQP, given a pair of questions and a duplicate label to label whether the pair of questions is duplicate or not. Many contextual language models such as ELMo [13], BERT [6], and its variant of the pre-trained model uses GLUE test as its benchmark, thus many pre-trained language models have been tested and tuned for Paraphrase Identification task. The pre-trained language models mostly see the Paraphrase Identification

task as a classification task which uses [CLS] token to classify whether the two questions are duplicate or not.

Some variants of the pre-trained language models such as Charformer [14] a fast character transformer via gradient-based sub word tokenization achieved F1 score of 88.5 and accuracy score of 91.4; RealFormer [15] a residual attention transformer achieved F1 score of 88.28 and accuracy score of 91.34; and FNet [16] a Fourier transform mixing tokens transformer achieved an F1 score of 85. Most of these variants are trained specifically on one task. While the original BERT-large [3] itself achieved an F1 score of 72.1 on the test dataset and F1 score of 88 on dev set [16]. There are some competitive state-of-the-art results without pre-trained language models such as data2vec [17] that achieved the accuracy score of 92.4.

Many research studies also did Paraphrase Identification in another language. One of the examples is the extraction of lexical, syntactic, and semantic features combined with MaxEnt [18] on Arabic news tweets. Charformer [14] as well can be used on multilingual NLP tasks. For multilingual, the Paraphrase Identification task dataset used is PAWS-X [19].

EFL [9] model is a combination of a task reformulation into entailment and RoBERTa [20] pre-trained language models. EFL reformulates the Paraphrase Identification classification into entailment classification by adding static textual entailment such as 'it was great' or 'it was bad' for good or bad classification. In general, the structure of one sentence task is 'S1 [SEP] It was great 'then the classifier token will classify whether the S1 is an entailment to the textual entailment 'it was great' or not. For two sentence task, EFL research study uses 'S1 [SEP] S2' or 'S1 [SEP] S2' where S1 is an augmentation of S1. EFL achieved the F1 score of 89.2.

Another entailment like method is Prompt-based Finetuning method [21]. In Prompt-based Finetuning, it is more of a masked language model task. In general, the structure of one sentence task is S1 'It was [MASK]' where [MASK] token will only consider the defined words label, for example [MASK] is either 'great' or 'bad'. For two sentence tasks it will be S1 [MASK], S2; In QQP Paraphrase Identification task, the [MASK] token will be 'entailment' or 'not entailment'. From these two methods, the author uses the combined entailment method [9] and prompt-based finetuning [21]. So, in a two-sentence Paraphrase Identification task, it will be [CLS] S1 'is paraphrase to' S2.

Modified EDA [8] inspired by the original EDA [7] with the purpose to produce positive augmentation which means that the augmented sentence has slightly different wording but still has the exact same meaning compared to the original. Modified EDA is used on sentiment analysis tasks with aspect classification. Like the original EDA, modified EDA also consists of four main augmentations. Modified synonym replacement (SR) which replace random k words with its synonym with the same POS tag, modified random insertion (RI) which selects random k words and then insert their synonyms with the same POS tag and insert it adjacent to the original words, modified random swap (RS) only allows swapping words if the pair of words is not an aspect word or adjective, and modified random deletion (RD) which does not delete the word if it is an aspect word or adjective. Both modified RS and RD also consider the POS tag and word similarity. Modified EDA also uses back translation after the modified SR, RI, RS, and RD is executed. In results, modified EDA helps the model for either balancing or increasing the amount of data which in return increases the accuracy and F1 score from the original dataset.

In the BERT [6] paper itself, it is stated that one of the best reasons for using BERT is because it doesn't need a complex classifier. As the BERT transformer layer is already complex and has been trained with billions of words, the original BERT classifier is just one neural network as output layer. But this doesn't stop other researchers from experimenting with another traditional machine learning (ML) classifier up to complex Neural Network (NN) classifiers by feeding them the BERT token. Faisal and Mahendra [22] researched on comparing and find the best classifier for their specific Indonesian Tweet task by comparing traditional ML such as Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and XGBoost (XGB) up to more complex NN such as original BERT classifier, Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Bidirectional Long-Short Term Memories (Bi-LSTM) on top of the BERT token. As the results, on average, the DNN classifier won against the traditional ML while Bi-LSTM being the best overall winning around 2% on almost all metrics. XGB, RF, and LR show better Recall value compared to all classifiers. Yu et al. [23] research on text classification based on the BERT-BiGRU Model. It shows that BERT-BiGRU achieves the highest precision, recall, and F1 with the value of 95% defeating all other competitors such as BERT-CNN, BERT-RNN, ELMo-BiGRU, word2vec-BiGRU. This shows the potential of other complex NN classifiers even though it will take a longer train time.

3. PROPOSED METHODS

3.1 Modified EDA

Modified EDA [8] inspired by the original EDA [7] is an augmentation technique with the goal of making dataset richer and more balanced and indirectly boost performance on text classification tasks. Modified EDA consists of four main operations which are the modification of synonym replacement, random insertion, random swap, and random0020deletion. The problem with the original EDA is that for specific duplicate question identification task, it is quite likely the EDA can't keep the same meaning of a sentence; thus, a modified EDA is made to solve this problem.

Modified EDA implements part-of-speech tagging using Conditional Random Field (CRF) Tagger NLTK so when the operation is being used, it will not change the meaning of the sentence itself. To put it simply, it augments the data while keeping the sentence integrity. With this in mind, we could also do the opposite of what Modified EDA does which is changing the meaning of the sentence to have a completely different meaning from the original sentence and thus break the sentence integrity.

QQP has a dataset imbalance of roughly 1:2 and thus we want to make it balanced by augmenting data with the ratio of 2:1 with Our Modified EDA so it will result in a 3:3 dataset ratio. This can be done by augmenting data to be positive by using Modified Synonym Replacement (MSR), Modified Random Insertion (MRI), Modified Swap (MS), and Back Translate (BT) while augmenting data to be negative with Antonym Replacement (AR), Reverse Modified Synonym Replacement (RMSR), Random Deletion (RD), and Random Swap (RS) (Figure 1).

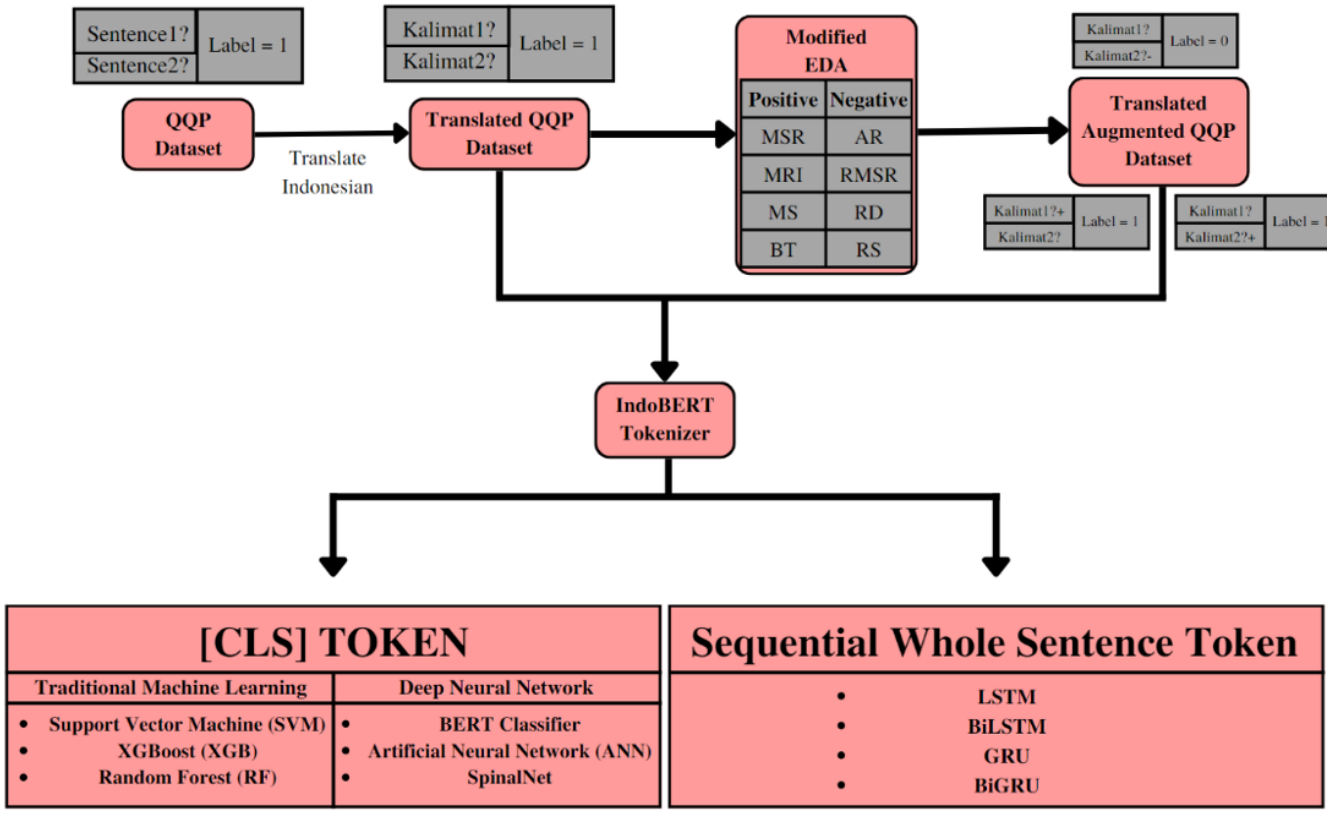

Figure 1. IndoBERT-based model for Paraphrase Identification task on Indonesian QQP DQI dataset

Augmenting data positively with MSR, MRI, and BT can be done the way it is while with MS, we swap the first or two words to the back of the sentence as this will not change the data meaning that much because of the nature of question sentence. The four functions MSR, MRI, MS, and BT have a ratio of 3:2:2:3 to choose between which function to use to augment the data positively. MSR and BT have a better probability of applying because it maintains the best sentence integrity while also changing the sentence and still having the same meaning. While with MRI and MS, sometimes it can't keep the sentence integrity as MRI can insert a random word not connected to the sentence itself while MS can make a peculiar sentence arrangement making the sentence itself hard to understand even for humans.

Augmenting data negatively needs to be done the way that it makes the sentence lose its real meaning or have a completely different meaning while keeping the sentence integrity so the model can learn from the comparison between 2 question pairs sentences even if there is only a small difference of one or two words between each sentence. Keeping this in mind, AR can be used the same way as SR by changing a word with its antonym. RMSR can be applied as normal, but most of the time it will result in not changing the sentence meaning at all because some synonyms have a different POS Tagging while still having the same word on itself. After applying RMSR augmentation, the sentence can then be augmented more by using RS function with 100% probability of it applying and RD with 33% probability to make sure the sentence is successfully augmented negatively. RD and RS have the problem of its augmented sentence having the same meaning because the word that is deleted or swapped might not be significant to its sentence and thus, we use the same approach of augmenting the sentence more. RD augmentation can be augmented more using RS with 50% probability of it applying and RS augmentation can be augmented more using RD with 50% probability of it applying. The four functions AR, RMSR, RD, and RS have a ratio of 3:1:4:2 to choose between which function to use to augment the data positively. RD has the highest probability between all the functions because it has the best negative augmented sentence keeping the sentence integrity while making the sentence have a completely different meaning while RMSR has the problem that it fails to satisfy the requirement to augment a data negatively without the added method of another augment function.

Table 1. Example of positive modified EDA operation in English (but in this research, we applied it in Indonesian)

Operation	Augmented Sentence
Modified synonym replacement	How to command this emotion of anger?
Modified random insertion	How to control this emotion feeling of anger wrath?
Modified swap	Control this feeling of anger how to?
Back translate	How to control this feeling of anger?

Table 2. Example of negative modified EDA operation in English (but in this research, we applied it in Indonesian)

Operation	Augmented Sentence
Antonym replacement	How to surrender this feeling of comfort?
Reverse modified synonym replacement	How to control this sensitive feeling of anger irritate?
Random deletion	To this feeling of anger?
Random swap	Control to how of feeling this anger?

Example of positive augmentation operation is shown Table 1 and negative augmentation operation in Table 2 with k=2 (p=0.3 for Random Deletion). and the sentence of "How to

control this feeling of anger?"

3.1.1 Positive augmentation

Modified synonym replacement (MSR). The main function of modified synonym replacement is to replace random words within the sentence with its synonym with the same POS tag.

Modified random insertion (MRI). The main function of modified random insertion is to insert a word anywhere within the sentence. The modified part is the word and the location. The word is the synonym of the selected random word, and the location will be behind or in front of the selected random word.

Modified swap (MS). The main function of modified swap is to swap words anywhere within the sentence. The modified part is the word will only be swapped next to each other so the sentence will not lose its integrity and initial meaning. It also restructures the whole sentence where the given context is in the front while the interrogative word is at the back of the sentence. It gives the data richer sentence structure yet keep maintaining the same meaning.

Back translate (BT). The main function of back translate is to translate the sentence into another language and then back to the original language. For our operation, based on preliminary experiments, Chinese language have shown more diversity and variation of the original sentence in the backtranslation result and thus each sentence will be back translated with Chinese language (Taiwanese Mandarin) which could produce different sentence as a result while keeping the same meaning.

3.1.2 Negative augmentation

Antonym replacement (AR). The main function of antonym replacement is to replace random words within the sentence with its antonym.

Reverse modified synonym replacement (RMSR). The main function of reverse modified synonym replacement is to replace random words within the sentence with its synonym with a different POS tag.

Random deletion (RD). The main function of random deletion is to randomly delete each word in the sentence with certain probability. If all the sentences are deleted, then just return a random word.

Random swap (RS). The main function of random swap is to randomly swap random words within the sentence without any limitation of which word is swapped.

3.2 IndoBERT tokenizer

IndoBERT [1] is a BERT based model that is trained using the Indo4B dataset. It is compiled from 12 datasets: two Indonesian colloquial language, eight formal Indonesian language, and the rest is a mixed style of both. Its sources from online news, social media, Wikipedia, online articles, subtitles, and parallel datasets.

IndoBERT follows the BERT model structure which means it consists of the modification of the Transformer structure. While transformer structures use encoder-decoder structure, BERT only uses the encoder and BERT is compiled from multilayer bidirectional encoder. IndoBERT follows how BERT is pre-trained, which means it is trained using two unsupervised approaches: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Just like the original BERT, pre-trained IndoBERT can be used for two main purposes, fine tuning which means that it can be used and fine-tuned for a specific NLP task or extracting embeddings. IndoBERT embeddings consist of word embeddings and positional encoding. It has a special token [CLS] other than other word tokens that can be used as the classifier token. In this research paper, we use the IndoBERT embeddings and feed them into the various classifiers to achieve the best classifier for the Paraphrase Identification task.

As the Paraphrase Identification task is a two-sentence task, before it goes to the tokenizer, it needs to be concatenated between the two sentences and between the IndoBERT special token. The concatenation will be [CLS] S1 [SEP] S2 [SEP] [PAD] … [PAD]. In the exploratory data analysis, it shows that each of the questions is mostly not over 40 words, hence the question that has more than 40 words counted as outliers, and we limit the max length to 40 each sentence. Thus, it results in 83 token length for the concatenated (80 for two sentences plus [CLS] plus two [SEP]). It is important to note that IndoBERT doesn't get trained again as we only use the tokenizer.

For classification tasks, as IndoBERT produces a sequence of hidden states, it needs a way to be reduced to a single vector, thus the original author adds a special token [CLS] that can represent the whole sentence. There are some ways to convert a whole sentence into a single vector that represents the whole sentence, one by max or mean pooling, another one by applying an attention mechanism. In BERT and IndoBERT, they apply attention throughout the whole sentence and aggregate it towards the [CLS] token embedding. Attention mechanism allows the model to selectively focus on different parts of the input sequence at each step of the decoding process. It does this by computing a set of attention scores that measure the relevance of each input position to the current decoding step. These attention scores are then used to compute a weighted sum of the input sequence, which is used as input to the next decoding step. The computation of attention scores involves three components: a query vector, a set of key vectors, and a set of value vectors. The query vector represents the current decoding state, while the key and value vectors represent the input sequence. The attention scores are computed by taking the dot product between the query vector and each key vector. Thus, in the next classifier section, most of the classifiers utilize only the [CLS] token.

3.3 Classifier

One of the most common approaches to fine-tune pre-trained BERT models is to upgrade the original output layer to a more complex or task-specific layer [24]. To an extent of modifying and learning the weights of Transformer blocks, word embeddings, the pooler, and the output layer parameters show some significant impact especially on the small datasets. Most of the trials are done by following the BERT train method which validates in dev set [6].

3.3.1 Traditional machine learning

For the traditional machine learning classifiers shown in Figure 2, authors use the best performing and the most used from the past research over the IndoBERT token. The first one is XGBoost, a distributed gradient boosting library that has been developed to be very effective, adaptable, and portable. It uses the Gradient Boosting framework to implement machine learning algorithms. XGBoost offers a parallel tree boosting method (sometimes referred to as GBDT or GBM) that quickly and accurately solves many data science problems including classifying paraphrases. The second one is SVM

with the goal to classify new data point in the correct category by creating the best line or decision boundary that can slice apart n-dimensional space into classes. A hyperplane is the name given to this optimal decision boundary. SVM selects the extreme vectors and points that aid in the creation of the hyperplane. Support vectors are used to describe these extreme scenarios. The third one is Random Forest, constructed by fitting several decision tree classifiers on various sub-samples of the dataset to build an estimator. It uses averaging to increase predictive accuracy and reduce overfitting. This three-classifier model is taking an input of the [CLS] token from the IndoBERT tokenizer. The hyperparameter and the results of this traditional machine learning are shown in experiments setup and experiments result sections.

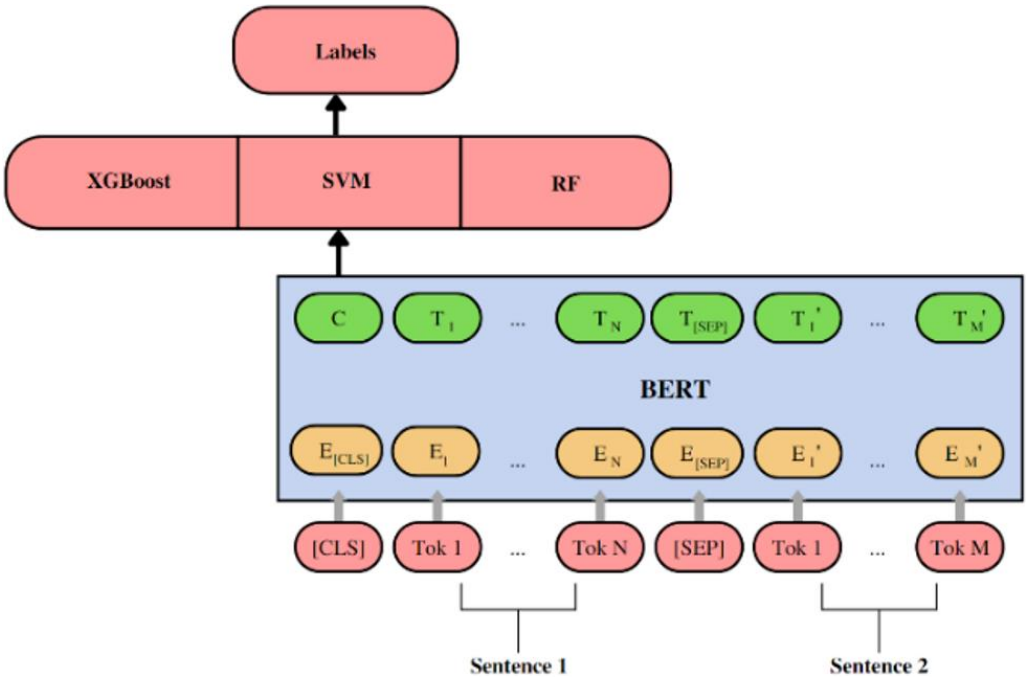

Figure 2. Framework of traditional machine learning classifier on top of the IndoBERT [CLS] token

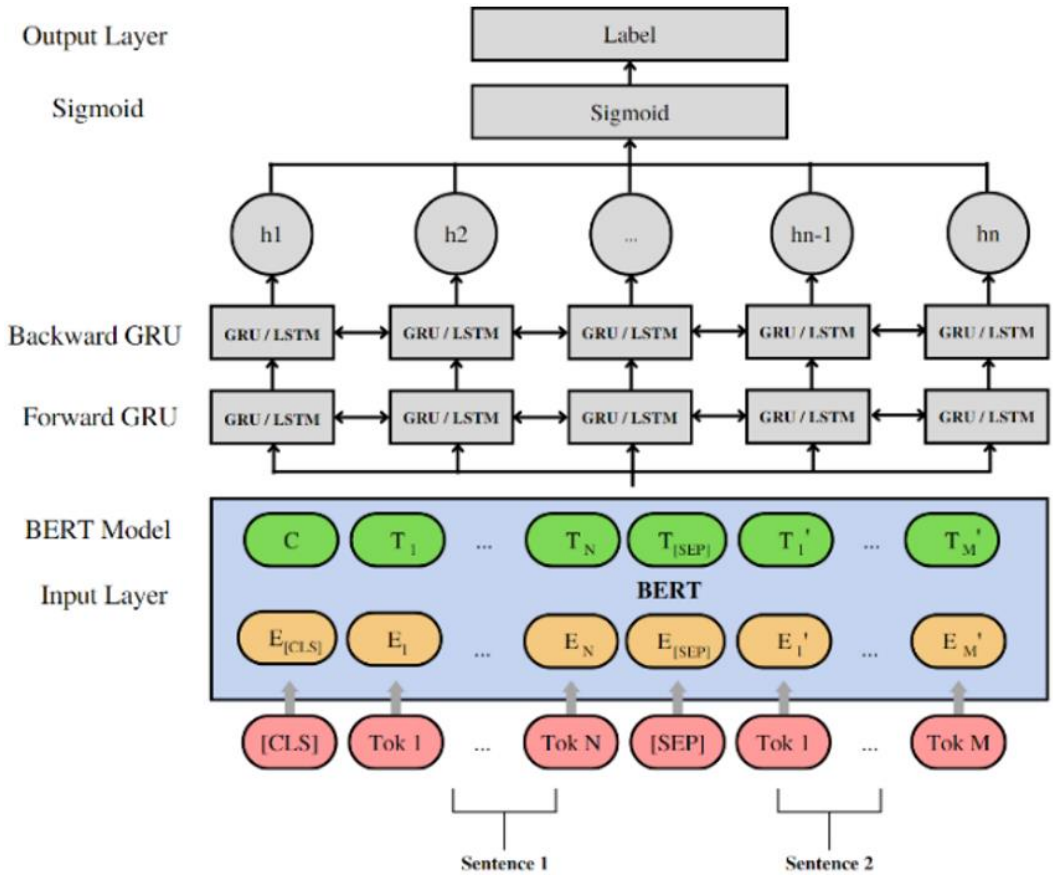

Figure 3. Framework of LSTM/GRU/BiLSTM/BiGRU classifier on top of all IndoBERT token

3.3.2 LSTM and GRU

From Figure 3, GRU [11] and LSTM [10] are both variants of RNN. LSTM comes first and then GRU is the evolution of the LSTM. LSTM is a patch on RNN to fix the serious gradient disappearance issue when processing sequences. The horizontal line that runs through the top of the diagram is called the cell state and it is the main key to LSTMs. The LSTM can modify the cell state by removing or adding information, which is carefully controlled via gates. Another use of the gates is to pass information, though this is optional. They are built out of a sigmoid neural network layer and a pointwise multiplication operation. Indicating how much of each component should be allowed through, the sigmoid layer generates integers between zero and one.

The drawback is that LSTM has multiple parameters and a long training time. Thus, GRU is born as a lighter LSTM in processing yet still able to achieve what LSTM can achieve. There are fewer parameters thus it reduces the training time. Just like LSTM, GRU is also suitable for processing sequential data and memorizing information of previous nodes through the gate to solve the gradient vanishing issue. While LSTM has three gates, GRU has only two gates which are update gate and reset gate as the equation above shows, thus it results in fewer parameters. Same as LSTM, GRU uses Sigmoid function to remap the value between 0 and 1 as the gate control signal.

Both LSTM and GRU can be used bidirectionally to solve a task where the current output state is also related to the subsequent, to the previous. Hence there are BiLSTM and BiGRU that utilize bidirectional to process sequential data. Unlike the traditional ML, LSTM and GRU utilize the whole sequence of tokens from the concatenated questions, not only the [CLS] token as it processes sequential data. The hyperparameter and the results of this traditional machine learning are shown in experiments setup and experiments result sections.

3.3.3 BERT, ANN, and SpinalNet classifier

From Figure 4, the original BERT [6] utilizes one NN output layer consisting of one neuron in this Paraphrase Identification task. It uses sigmoid function to remap the value between 0 and 1 means if the value nears 1 then the pair of questions is considered as paraphrase. From this, it came an idea to build a more complex classifier such as ANN. We built a 2 hidden layer ANN. Much research proves that the more correct layers added results in the cost of longer training time.

Finally, we researched the use of SpinalNet classifier [12] used on top of the IndoBERT tokenizer. SpinalNet has shown its performance of regression and classification especially on computer vision, but not yet frequently used on NLP. SpinalNet is a more complex NN architecture. SpinalNet is inspired by the human spinal cord. It consists of 3 parts: input split, intermediate split, and output split. It consists of 3 steps: Gradual input, Local output and probable global influence, and Weights reconfiguration in training step. The input split is a layer where it takes the dataset input, the intermediate split takes output from the previous intermediate split and output of the input splits. Each layer in the SpinalNet contributes to the reflex (local output) and sends a modulated version of inputs to its brain (global output).

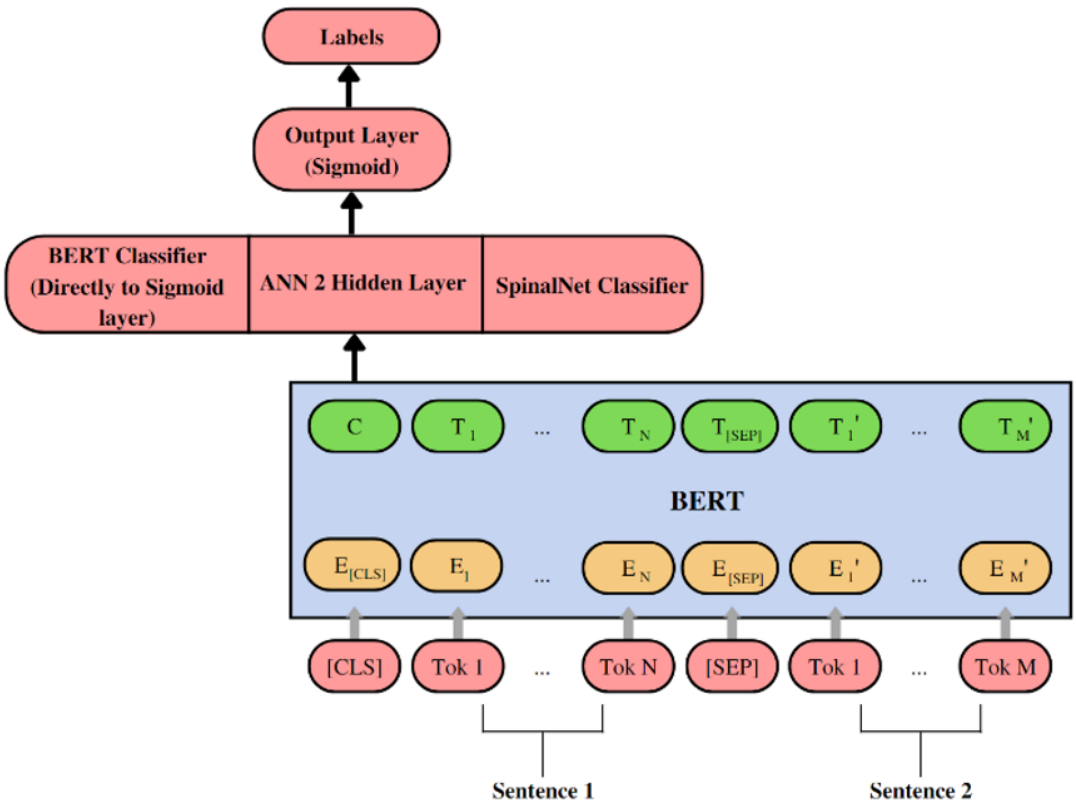

Figure 4. Framework of DNN classifier on top of the [CLS] IndoBERT token

These DNN architecture classifiers utilize only the [CLS] token of the IndoBERT tokenizer. The hyperparameter and the results of this traditional machine learning are shown in experiments setup and experiments result sections.

3.4 Entailment Method

From Figure 5, entailment as Few-shot Learners (EFL) [9] is an approach which can turn small Language Models (LM)

into better few-shot learners. In general, the structure used by EFL for one sentence task is 'S1 [SEP] <Labels prompt>', for example, if the task is to classify if the sentence is a good or bad as sentiment analysis, it would be 'S1 [SEP] it was great', thus this can be learned by knowing whether S1 is an entailment to 'it was great' or not. For two sentence tasks it's 'S1 [SEP] S2' or 'S1 [SEP] S2' where S1 is an augmentation of S1.

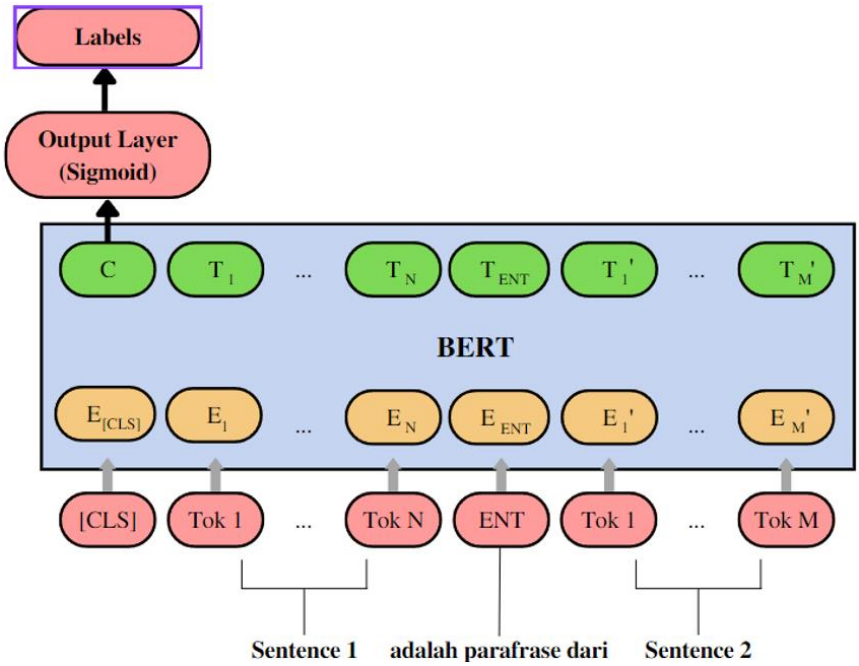

Figure 5. Framework of entailment task reformulation on Paraphrase Identification task

QQP itself is not a small dataset on its own, EFL has proven to increase model accuracy if trained with the full train dataset. There is also another entailment like method which is Prompt-based Fine Tuning method [21] which is more of a masked language model task. In general, the structure used by this method for one sentence task is S1 'It was [MASK]' and for two sentence tasks it is S1 [MASK], S2; In QQP dataset, the [MASK] token will be 'entailment' or 'not entailment'. From these two methods, the author uses the combined entailment method [9] and prompt-based finetuning [21]. So, in a two-sentence Paraphrase Identification task, it will be [CLS] S1 [SEP] 'adalah parafrase dari' [SEP] S2 [SEP] where 'adalah parafrase dari' is 'is paraphrase to' in English. With this, we hope that we can achieve better results by training all the train sets like what EFL has achieved as IndoBERT has also been tested on entailment tasks with WReTE dataset. The parameter tuning and the result are shown in experiments setup and experiments result sections. The reason behind this is that the LM will understand the task much faster because of the help of the entailment phrase which is 'adalah parafrase dari', so the LM will understand to classify PI.

Example of our entailment reformulation:

S1: Terlepas dari Miliaran orang, Mengapa India begitu buruk di Olimpiade? Mengapa india kekurangan talenta?

S2: Mengapa India tampil sangat buruk di Olimpiade?

In English, it means:

S1: Despite Billion people, Why India is so bad at Olympics? Why does India lack talent?

S2: Why does India perform so poorly at the Olympics?

Entailment task reformulation: [CLS] 'Terlepas dari Miliaran orang, Mengapa India begitu buruk di Olimpiade? Mengapa india kekurangan talenta?' [SEP] 'adalah paraphrase dari' [SEP] Mengapa India tampil sangat buruk di Olimpiade? [SEP]. In English, it means: [CLS] 'Despite Billion people, Why India is so bad at Olympics? Why does India lack talent?' [SEP] 'is a paraphrase to' [SEP] Why does India perform so poorly at the Olympics? [SEP]

Note that the entailment method experiment is conducted on top of the 3.1 Modified EDA and 3.2 IndoBERT tokenizer and separately from the various classifier experiments. In this case, we only compare the baseline of the PI task and the entailment reformulation task using the original BERT classifier.

4. EXPERIMENTS

4.1 Overview

To achieve the best IndoBERT-based model for Indonesian language Paraphrase Identification task, we experiment on some techniques such as augmentation, task reformulation, and building a more complex classifier. Modified EDA is used as the Data Augmentation, entailment task reformulation is used for task reformulation, and complex classifier deep neural network such as ANN, LSTM, BiLSTM, GRU, BiGRU, SpinalNet which also then compared to the original BERT classifier and traditional machine learning classifier. The experiment results show the comparison between the augmented data and non-augmented data, between the PI task and the entailment task, and between the classifier on top of augmented or non-augmented data to show the differences and increases of the F1 score and accuracy. We do hyperparameter tuning to see how the model works best on this Paraphrase Identification task.

4.2 Dataset

The dataset originated from the GLUE Benchmark [5] QQP task. It is translated into Indonesian language through Google Translate API. The reason being there is still no legit dataset

for Paraphrase Identification tasks. This encourages us to build a model based on the Paraphrase Identification task as Indonesian forum discussion is growing, thus it needs a duplicate question identification model. We can also indirectly compare how well IndoBERT based model understands translated QQP in Indonesian language compared to English based model such as BERT understanding the original English QQP. We are aware that just Google Translate API will not produce the perfect translation. The train set on the non-augmented data consists of 363,847 pairs of questions. The augmented train set consists of 766,981 pairs of questions.

The GLUE Benchmark itself has its own leaderboard that score the test set. The best model for QQP task is Vega v1 which holds an accuracy of 91.1 and F1 score of 76.7. The ground truth of the test set is not open publicly, thus for those who want to get a score on the test set must submit to their website. This creates some boundary and not a few papers do score on the dev set such as EFL [9] paper. Currently the best score on dev set is F1 score of 89.2 held by EFL. In this dev set, we mainly focus on F1 score as the metric because of the imbalance labels: 63.2% on 'not duplicate' and 38.2% on 'duplicate'. These rank 1s have a quite small improvement from rank 2 or 3. It usually only differs around 0.2% to 0.3% on the test set and around 0.7% to 1% on the dev set. From this leaderboard, it shows how difficult the dataset is, thus small improvement is valued. Some of the pair questions are also difficult to distinguish whether it is counted as duplicate or non-duplicate even by humans mostly because of the context given by the labeller.

4.3 Experimental setup

The IndoBERT tokenizer used is 'indobert-base-p2' which is trained on Indo4B (24.43 GB of text). It has 124.5 million parameters and 768 output size. Each pair of questions will be limited to a max sequence of 80 for two questions without the special token. For the deep neural network classifier, we apply early stopping at 5 non increasing steps, dropout: 30%, learning rate: (1e-5, 3e-5), batch size: (64, 128), optimizer: Adam, loss: binary cross entropy loss, objective: Paraphrase Identification and entailment classification, and output layer sigmoid. For the post labelling, when the output is >= 0.5 then it is considered 1, else is 0. Thus, each classifier is run in 4 combinations of learning rate and batch size and thus the represented comparison between the same corresponding hyperparameter is legit and not just a coincidence.

For the original IndoBERT classifier it utilizes the [CLS] token and goes directly to 1 neuron sigmoid output layer. For ANN we build two hidden layers in size of 512 and 128 before it goes to 1 neuron sigmoid output layer. For LSTM, BiLSTM, GRU, and BiGRU, we utilize all the token generated and it goes forward and backward in line with each token, it has 128 neurons in its layer before it goes to 1 neuron sigmoid output layer. For XGBClassifier, it has estimators: 512, learning rate: 0.05, colsample: 0.7, and max depth: 12. For RF Classifier, it has estimators: 512, learning rate: 0.05, max features: 768, and max depth: 12. For SVM, it has kernel: linear, C: 1, and cache size: 2,000. We utilize TPU v3.8 for training the deep neural network classifier. For the entailment task reformulation, the hyperparameter is the same with the PI task. The only difference is on the entailment task reformulation where 'Adalah parafrase dari' (in English, it means: 'is a paraphrase of') which has additional 6 tokens, thus the max length is 89 which consists of 40 for each of the question and 9 for the entailment statement. The entailment is done to both train and dev set.

4.4 Main results and discussion

The IndoBERT baseline reached an accuracy of 89.0% and F1 score of 85.3% for non-augmented data. For augmented data, the IndoBERT baseline reached an accuracy of 89.7% and F1 score of 86.4%. Figure 6 shows that LSTM classifier has the best result within the original dataset and Figure 7 shows that the ANN classifier has the best result within the augmented dataset. In the comparison of the training epochs between the original data and augmented data in Figure 6 and Figure 7, the DNN classifier for original data reaches epoch up to 43 before early stopping, while the augmented data reaches epoch up to 32 before early stopping. This shows that by learning the augmented data, the model learns faster as it has more rich and balanced data.

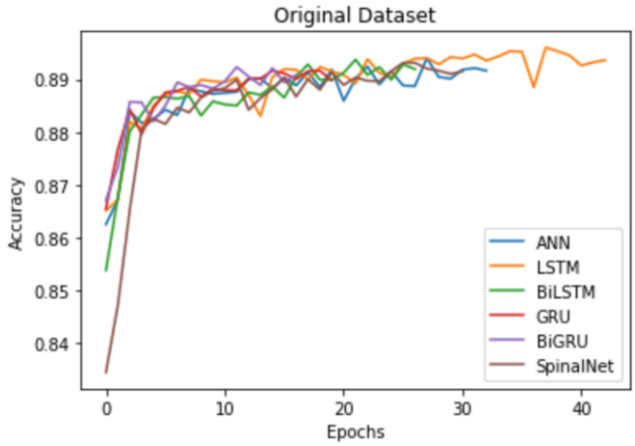

Figure 6. Comparison between how the DNN classifier train within each epochs on original dataset

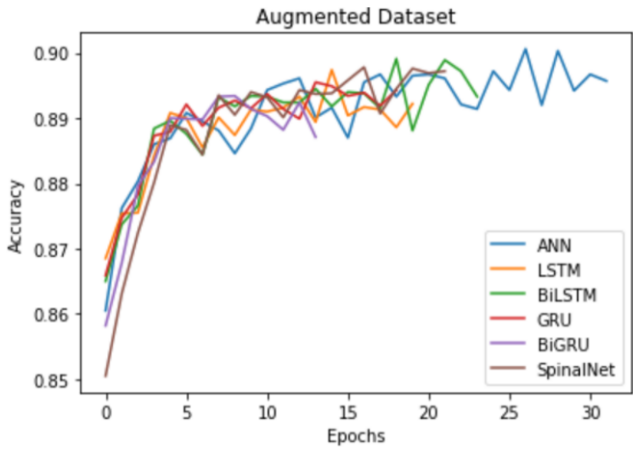

Figure 7. Comparison between how the DNN classifier train within each epochs on augmented dataset

From all the model results, there are several things that can be discussed further. First, the traditional machine learning model that it seems can work on IndoBERT with the highest accuracy of 82.1% and F1 score of 77.6% but this doesn't come close to the deep neural network classifier as can be seen in Table 3. The time needed to train the classifier on the original data is between 1h to 4h with each epoch needs around 200s - 370s on TPU v3.8, while on the augmented data, it needs 1h - 6h, which still makes sense for language models. The

differences between the deep neural network classifier might not seem significant but this might be caused by the complexity of the dataset itself and the reason on which BERT or IndoBERT can perform well in even a simple classifier. For the original dataset, LSTM classifier also held the best performance with an accuracy of 89.6% and F1 score of 85.8%. Which only has a difference of 0.2% and 0.1% from ANN on the original dataset. But this score is still quite far behind the augmented data combined with ANN with the difference of 0.9% of F1 score. It also shows that by upgrading the classifier into the more complex one it gives quite an average accuracy increase of 0.6% of accuracy and 0.5% of F1 score.

Table 3. Experiment results of the performance of traditional machine learning classifier XGB, RF, and SVM on top of IndoBERT-base-p2 for both original dataset and augmented dataset

Classifier	Accuracy (Ori)	Accuracy (Aug)	F1 (Ori)	F1 (Aug)
XGB	83.0	82.0	75.5	77.6
RF	64.3	71.1	60.5	56.6
SVM	76.5	72.1	67.4	68.0

Table 4. Experiment results of the performance of DNN classifier BERT, ANN, SpinalNet, LSTM, BiLSTM, GRU, BiGRU on top of IndoBERT-base-p2 for both original dataset and augmented dataset

Classifier	Accuracy (Ori)	Accuracy (Aug)	F1 (Ori)	F1 (Aug)
BERT	89.0	89.7	85.3	86.4
ANN	89.4	90.1	85.7	86.7
SpinalNet	89.3	89.8	85.6	86.3
LSTM	89.6	89.7	85.8	86.1
BiLSTM	89.3	89.9	85.5	86.6
GRU	89.2	89.6	85.3	86.2
BiGRU	89.2	89.3	85.3	85.6

Table 4 shows that on most of the experiments on comparison between augmented and non-augmented data using DNN classifier, the latter improves on both accuracy and F1 score. On average, Our Modified EDA improves 0.44% of accuracy and 0.77% of F1 score which shows a considerable improvement compared to its competitor model. The IndoBERT original classifier which has only 1 output sigmoid layer is not the best model shown in the comparison between classifier, ANN on augmented data achieves the highest accuracy of 90.1% accuracy and F1 score of 86.7% it shows 1.05% improvement on accuracy and 1.42% on F1 score compared to the baseline with the original data. The ANN classifier itself is compared to the original classifier.

For both original and augmented data and gives an average increase of 0.33% F1 score and 0.39% accuracy resulting in 86.7% F1 score and achieving rank 4th on the dev set leader board compared with the other transformers model that was benchmarked on QQP GLUE. We are aware that this is not completely comparable, but it gives us a picture of how IndoBERT performs in Indonesian language compared to how English BERT transformers perform in English. The results of accuracy and F1 score of the model can be seen in Table 4.

Table 5 shows the comparison between how IndoBERT plus one sigmoid layer classifier on original Paraphrase Identification task compares to entailment task. It shows that if both models are trained on the original dataset, the entailment reformulation task gives slightly increased accuracy from the baseline from 89.0 to 89.2 and increased F1 score from 85.3 to 85.7 on the original data. Further experiment is done by training the model on the augmented dataset. It shows the same accuracy and only an increase of 0.1% on the F1 score. This shows that the entailment reformulation task produces no significant increase in performance compared to the baseline model. But we believe that the entailment task performs better as a few-shot learners as it gives the sentence a context of paraphrase identification through the additional 'adalah parafrase dari' between the questions. It also showed how the entailment improves a lot more in the original dataset rather than on the augmented dataset. His few-shot research is left open for discussion for further research.

Table 5. Experiment results of the performance between PI task and Entailment task on top of IndoBERT-base-p2 for both original dataset and augmented dataset

Task	Accuracy (Ori)	Accuracy (Aug)	F1 (Ori)	F1 (Aug)
PI	89.0	89.7	85.3	86.4
Entailment	89.2	89.7	85.7	86.5

5. CONCLUSIONS AND FUTURE WORKS

We proposed fine-tuned IndoBERT-based model for Paraphrase Identification and Modified EDA as a better, simple, and cheap Data Augmentation for sentences. From the IndoBERT baseline on PI task which achieve comparable results compared to the other models and GLUE leaderboards, it can be concluded that IndoBERT is reliable, generic, and robusts for Indonesian NLP tasks especially PI. IndoBERT also doesn't need an additional complex classifier as it shows where the best model is built with the base of IndoBERT tokenizer and 2 hidden layer ANN classifiers. It reaches 90.1% of accuracy and 86.7% of F1 score. The proposed Modified EDA shows improvement on both accuracy and F1 score in all tested models. It means the Modified EDA is applicable to balance the dataset and IndoBERT-based model shows that it converges faster and perform better on balanced data. On the other hand, the experiment on entailment method instead of classic BERT classification shows slightly increases performance only in some cases. Based on the result, the prepared dataset QQP Indonesia is proper and able to contribute on Indonesian NLP dataset.

We have thought about the possibilities of the future works in Indonesian NLP task such as Research on more complex transformer tokenizers such as FNet, Charformer, RealFormer, and ensemble tokenizers which are trained using the Indonesian Corpus. This modified transformer structure can also be used in another NLP task besides Paraphrase Identification. Might be a good shot to research multilingual models to perform on Indonesian Paraphrase Identification tasks. Finally, Train IndoBERT with the PI datasets to perform well in specific task.

REFERENCES

[1] Wilie, B., Vincentio, K., Winata, G.I., Cahyawijaya, S., Li, X., Lim, Z.Y., Soleman, S., Mahendra, R., Fung, P., Bahar, S., Purwarianti, A. (2020). IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. arXiv preprint arXiv:2009.05387.

[2] Aji, A.F., Winata, G.I., Koto, F., et al. (2022). One country, 700+ languages: Nlp challenges for underrepresented languages and dialects in Indonesia. arXiv preprint arXiv:2203.13357.
[3] Koto, F., Rahimi, A., Lau, J.H., Baldwin, T. (2020). IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. arXiv preprint arXiv:2011.00677.
[4] Kalbhor, P., Patil, G., Agarwal, S., Rajput, A.S., Dhamdhere, P. (2021). Research on paraphrase identification. International Research Journal of Engineering and Technology (IRJET), 8(5): 4050-4055. https://www.irjet.net/archives/V8/i5/IRJET-V8I5743.pdf.
[5] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.
[6] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[7] Wei, J., Zou, K. (2019). Eda: Easy Data Augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.
[8] Girsang, A.S. (2022). Modified EDA and backtranslation augmentation in deep learning models for Indonesian aspect-based sentiment analysis. Emerging Science Journal, 7(1): 256-272. https://doi.org/10.28991/ESJ-2023-07-01-018
[9] Wang, S., Fang, H., Khabsa, M., Mao, H., Ma, H. (2021). Entailment as few-shot learner. arXiv preprint arXiv:2104.14690.
[10] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
[11] Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
[12] Kabir, H.D., Abdar, M., Khosravi, A., Jalali, S.M.J., Atiya, A.F., Nahavandi, S., Srinivasan, D. (2022). SpinalNet: Deep neural network with gradual input. IEEE Transactions on Artificial Intelligence. https://doi.org/10.1109/TAI.2022.3185179
[13] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint. arXiv preprint arXiv:1802.05365.
[14] Tay, Y., Tran, V.Q., Ruder, S., Gupta, J., Chung, H.W., Bahri, D., Qin, Z., Baumgartner, S., Metzler, D. (2021). Charformer: Fast character transformers via gradient-based subword tokenization. arXiv preprint arXiv:2106.12672.
[15] He, R., Ravula, A., Kanagal, B., Ainslie, J. (2020). Realformer: Transformer likes residual attention. arXiv preprint arXiv:2012.11747.
[16] Lee-Thorp, J., Ainslie, J., Eckstein, I., Ontanon, S. (2021). Fnet: Mixing tokens with fourier transforms. arXiv preprint arXiv:2105.03824.
[17] Baevski, A., Hsu, W. N., Xu, Q., Babu, A., Gu, J., Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555.
[18] Mohammad, A.S., Jaradat, Z., Mahmoud, A.A., Jararweh, Y. (2017). Paraphrase Identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features. Information Processing & Management, 53(3): 640-652. https://doi.org/10.1016/j.ipm.2017.01.002
[19] Yang, Y., Zhang, Y., Tar, C., Baldridge, J. (2019). PAWS-X: A cross-lingual adversarial dataset for Paraphrase Identification. arXiv preprint arXiv:1908.11828.
[20] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
[21] Gao, T., Fisch, A., Chen, D. (2020). Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723.
[22] Faisal, D.R., Mahendra, R. (2022). Two-stage classifier for covid-19 misinformation detection using Bert: A study on Indonesian tweets. arXiv preprint arXiv:2206.15359.
[23] Yu, Q., Wang, Z., Jiang, K. (2021). Research on text classification based on Bert-BIGRU model. Journal of Physics: Conference Series, 1746: 012019. http://dx.doi.org/10.1088/1742-6596/1746/1/012019
[24] Zhang, T., Wu, F., Katiyar, A., Weinberger, K.Q., Artzi, Y. (2020). Revisiting few-sample BERT fine-tuning. arXiv preprint arXiv:2006.05987.