Identifying Indonesian Sentences Containing Idiomatic Expression Using the BERT Model

AAIN Eka Karyawati[1,*] and NM Yuli Cahyani[2]

[1, 2] Udayana University, South Kuta, Jimbaran, Indonesia
*eka.karyawati@unud.ac.id

Abstract. Idiomatic expressions are expressions that consist of a series of two or more words that have a meaning that cannot be predicted from the meaning of the individual words that compose them. Idiomatic expressions exist in almost all languages but are difficult to extract because there is no algorithm that can precisely decipher the structure of idiomatic expressions, so most rule-based machine translation systems generally translate idiomatic expressions by translating the constituent words word by word, but the translation results are not produce the true meaning of the idiom expression. In this research, the BERT model is used to identify sentences in Indonesian sentences that contain idiomatic expressions. The dataset used is a collection of basic Indonesian sentences that contain idiomatic expressions and basic Indonesian sentences that do not contain idiomatic expressions. This data amounts to 2000 sentences which have been labeled as non-idiomatic sentences and idiomatic sentences manually based on the Indonesian idiom dictionary book, with the number of sentences on each label being 1000 sentences. From the research conducted, the classification process using BiDirectional Encoder Representations from Transformers (BERT) obtained an *Accuracy* of 0.97, *Precision* 0.96, *Recall* 0.98 and *F1-Score* 0.97, respectively, with *Learning Rate* 2e-5 and *Epoch* 5.

Keywords: Idiomatic Expression, BERT Classifier Model, Idiomatic Sentence Identification.

1 Introduction

Idioms are phrases consisting of a series of two or more words whose expressed meaning cannot be derived from their parts [1]. Idioms exist in almost all languages and are difficult to extract because there is no algorithm that can decipher the structure of an idiom precisely. Identification of idiomatic expressions is a challenging and widely applicable problem. Several researchers have conducted research on the identification of idiomatic expressions. Most studies use machine learning models.

K. Dharmawan and N. A. Sanjaya ER (eds.), *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, Advances in Computer Science Research 110,
https://doi.org/10.2991/978-94-6463-413-6_16

The use of machine learning with non-contextual word embedding models, such as word2vec, is less good at handling the identification of idiomatic expressions [2]. Simple superposition of word embeddings is unable to express the semantics of idiomatic phrases precisely. Therefore, contextual embedding models (e.g. BERT models) are needed to properly understand the meaning of multi-word expressions in idiomatic expressions.

[3] used a pre-trained BERT model which was compared with three basic models namely the linear embedding layer model, ELMo model, and RoBERTa. The pre-trained BERT model is also frozen for all base and DISC models. The results show that for a random split of the MAGPIE dataset, it is noteworthy that all models achieve at least 86% F1 score. For MAGPIE type separations, DISC is clearly the best performing model with an absolute gain of at least 1.4% in F1. For the SemEval5B dataset type split, BERT outperforms DISC by 14.6% in terms of SA with an improvement of at least 11.1%.

Similar to [3], [2] used a large-scale cross-language trained language model, multilingual BERT and XLM-RoBERTa with Softmax classifier on top of a pre-trained LM model to train a binary classification model. The final result uses a fusion model on thirteen models. The zero-shot model is ranked fourth with an *F1-Score* of 77.15%, and the one-shot model is ranked first with an *F1-Score* of 93.85%.

Other researchers, [4] implemented BERT-based-multilingual-cased (mBERT). They performed hyper-parameter tuning for 30 epochs with Cross-Entropy loss criteria, adopted an early stopping strategy with a patience value of 5, an *Adam* optimizer, and *Learning Rate* 1e-5. Experiments show that the system is able to generalize beyond the idioms seen during training, achieving *F1-scores* of up to 85.4%.

In our research, we adopt a pre-trained BERT model to identify sentences containing idiomatic expressions. Different from others, we apply idiomatic expression identification to Indonesian. The focus of the research is to select the best identification model for Indonesian sentences containing idiomatic expressions with a limited dataset by utilizing the pre-trained BERT model and performing hyper-parameter tuning.

2 Research Method

2.1 Idiomatic Expressions

The idiom expressions that will be identified in this research are idioms that have the categories of noun phrases, verb phrases and adjective phrases which are composed of two words. Examples of idiomatic expressions can be seen in Table 1.

Table 1. Example of Indonesian Idiomatic Expressions.

Idiomatic Expression	Meaning	Category

Indonesian	English (Direct Translation)		
buaya darat	land crocodile	playboy	Noun Phrase
kuda hitam	dark horse	unexpected winner	Noun Phrase
jago merah	red hero	fire	Noun Phrase
angkat kaki	lift the feet	leave	Verb Phrase
gulung tikar	roll up the mat	bankrupt	Verb Phase
banting tulang	slam the bone	work hard	Verb Phrase
gelap mata	darken the eyes	very angry	Adjective Phrase
rendah hati	lower heart	humble	Adjective Phrase
naik daun	get on leaf	popular	Adjective Phrase

2.2 Dataset

The dataset is a collection of basic patterned sentences in Indonesian that contain idiomatic expressions and basic patterned sentences in Indonesian that do not contain idiomatic expressions. This data amounts to 2000 sentences which have been manually labeled as idiomatic and non-idiotic sentences with the number of sentences in each label being 1000 sentences. This labeling is based on the Indonesian idiom dictionary book [5]. Examples of sentences that use idiomatic expressions in Table 1 can be seen in Table 2.

Table 2. Example of Dataset.

No	Sentence	Label
1	Yoga seorang *buaya darat* yang suka mempermainkan wanita. (Yoga is a playboy who like playing with woman.)	Idiomatic
2	Liverpool menjadi *kuda hitam* di liga champion 2018. (Liverpool became *unexpected winners* in the 2018 Champions League.)	Idiomatic
3	Si *jago merah* membakar habis ruko di pasar tersebut. (The *fire* completely burned down the shophouses in the market.)	Idiomatic
4	Mereka dipaksa *angkat kaki* dari rumah kontrakannya. (They were forced to l*eave* their rented house.)	Idiomatic
5	Perusahaan besar itu pada akhirnya *gulung tikar* juga. (The big company finally went *bankrupt* too.)	Idiomatic
6	Andi *banting tulang* untuk menghidupi keluarganya. (Andi *works hard* to support his family.)	Idiomatic
7	Ia mengamuk di kantor karena *gelap mata.* (He threw a tantrum in his office because *so angry*.)	Idiomatic
8	Pak tua merupakan sosok yang sangat *rendah hati*. (The old man is a very *humble* figure.)	Idiomatic
9	Chelsea Islan merupakan salah satu aktris yang tengah naik daun. (Chelsea Islan is a currently popular actress.)	Idiomatic

2.3 Text Preprocessing

In this research, dataset is preprocessed first so that the data is ready and can be processed for the next stage. Text preprocessing is used to present data in the form of text in an appropriate format. The steps taken for text preprocessing in this research were punctuation removal, tokenization, and case conversion.

2.4 The BERT Model

The BERT model layers are created based on the BERT architecture shown in Fig. 1. The architecture based on the BERT base [6] architecture with output layers [7].

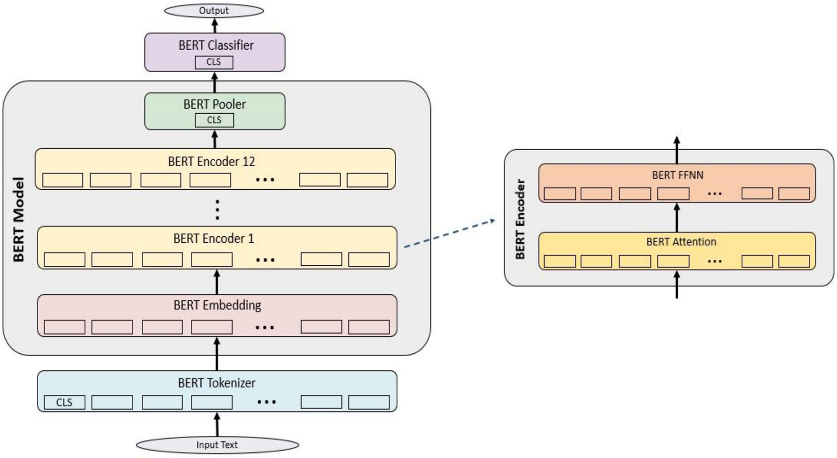

Fig. 1. The Bert Architecture [7].

The BERT Embedding represents the input sentence in vector form which then becomes input to the BERT Encoder [8]. In this layer there are three embeddings, namely [6] which are token, segment, and positional embedding. Token embedding is a vector representation of each token (word) in a sentence. Segment embedding is a representation to display the first sentence or second sentence. Positional embedding is a representation of the order or position of each token in a sentence. Positional embedding accepts input in the form of position_ids, namely the position of each token in the sentence.

BERT attention is self-attention which is an attention mechanism that connects different positions of a sequence to compute a representation of the same sequence. The attention mechanism allows the output to focus attention on the input while producing the output whereas the self-attention model allows the inputs to interact with each other.

BERT pooler takes the first token output from the Bert Model, namely the [CLS] token, this token is a special classification token that is used as an aggregate series representation for classification tasks. After getting the output token [CLS] it will then go through the Linear layer. Next, to obtain the combined output, the Tanh activation function is applied.

The BERT Classifier layer is build based on architecture in Fig. 2. This BERT Classifier layer produces classification results using the Softmax function. The resulting output has a value of 0 for non-idiomatic sentences and 1 for idiomatic sentences.

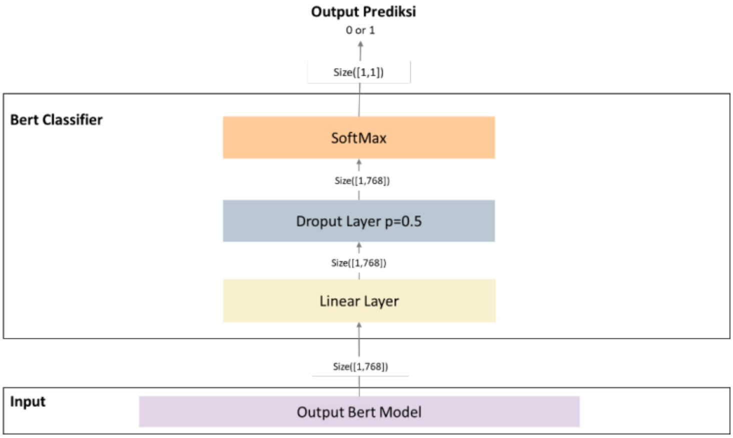

Fig. 2. The BERT Classifier.

2.5 Tuning Hyper-Parameter

Tuning hyper-parameter were carried out to select the best BERT model using the 5-Fold Cross-Validation method. The evaluation metrics used in this research are *Accuracy*, *Precision*, *Recall*, and *F1-Score*. The BERT classification testing process is carried out based on the hyper-parameter configuration as can be seen in Table 3.

Table 3. Configurations of Hyper-parameter.

Hyper-parameter	Values
Dropout	5
Learning Rate	2e-4, 3e-4, 2e-5, 3e-5, 2e-6, 3e-6
Batch Size	16
Epoch	3, 5

3 Results and Discussion

3.1 Tuning Hyper-Parameter Results

In this research, the BERT classification model training uses a dataset of Indonesian sentences which is divided into a ratio of 80% for training and validation data and then 20% for testing data. The BERT model validation results for each fold can be seen in Table 4 (*Epoch* = 3) and Table 5 (*Epoch* = 5).

As seen in Table 4, the validation results of the BERT Model with *Epoch* = 3 show that the worst result is at a learning rate of 2e-4. However, the result values increases significantly when the learning rate decreases to 2e-5. After that, the performance decreases again when the learning rate drops to 2e-6. Hyper-parameter tuning shows good results when the *Learning Rate* value is small enough less than 2e-4. The best results were achieved at *Learning Rate* = 2e-5, where the respective evaluation results were 0.97125, 0.96538, 0.97511 and 0.97006 in terms of *Accuracy*, *Precision*, *Recall* and *F1-Score*.

Table 4. The BERT model validation results Epoch = 3.

Learning Rate	**Fold**	**Result**			
		Accuracy	***Precision***	***Recall***	***F1- Score***
2e-4	1	0.45313	0.45313	1.00000	0.62366
	2	0.50313	0.00000	0.00000	0.00000
	3	0.49375	0.00000	0.00000	0.00000
	4	0.47500	0.00000	0.00000	0.00000
	5	0.48438	0.00000	0.00000	0.00000
	AVG	0.48188	0.09063	0.20000	0.12473
3e-4	1	0.45313	0.45313	1.00000	0.62366
	2	0.49688	0.49688	1.00000	0.66388
	3	0.49375	0.00000	0.00000	0.00000
	4	0.52500	0.52500	1.00000	0.68853
	5	0.51563	0.51563	1.00000	0.68041
	AVG	0.49688	0.39813	0.80000	0.53130
2e-5	1	0.90938	0.87662	0.93103	0.90301
	2	0.96875	0.96855	0.96855	0.96855
	3	0.98750	0.98171	0.99383	0.98773
	4	0.99063	1.00000	0.98214	0.99099
	5	1.00000	1.00000	1.00000	1.00000
	AVG	**0.97125**	**0.96538**	**0.97511**	**0.97006**
3e-5	1	0.92188	0.95455	0.86897	0.90975
	2	0.96563	0.97436	0.95598	0.96508
	3	0.99063	0.98182	1.00000	0.99083
	4	0.99063	0.99401	0.98810	0.99105
	5	0.98750	0.99387	0.98182	0.98781

Learning Rate	**Fold**	**Result**			
		Accuracy	***Precision***	***Recall***	***F1- Score***
	AVG	0.97125	0.97972	0.95897	0.96890
2e-6	1	0.88125	0.86897	0.86897	0.86897
	2	0.93438	0.97917	0.88679	0.93069
	3	0.98750	0.97590	1.00000	0.98781
	4	0.99375	1.00000	0.98810	0.99401
	5	0.99375	0.99394	0.99394	0.99394
	AVG	0.95813	0.96360	0.94756	0.95508
3e-6	1	0.86875	0.84106	0.87586	0.85811
	2	0.93750	0.97279	0.89937	0.93464
	3	0.99063	0.98773	0.99383	0.99077
	4	0.99375	1.00000	0.98810	0.99401
	5	0.99375	1.00000	0.98788	0.99390
	AVG	0.95688	0.96032	0.94901	0.95429

Similar to the validation results of the BERT Model with *Epoch* = 3, the results of the BERT Model with *Epoch* = 5 obtained were low *Accuracy*, *Precision*, *Recall* and F*1-Score* values at *Learning Rate*s 2e-4 and 3e-4, where most of the results below 0.5. The result value increases significantly when the *Learning Rate* is less than 2e-4, namely around 0.95-0.98. Hyper-parameter tuning shows good results when the *Learning Rate* value is small enough less than 3e-4. The best results were achieved at *Learning Rate* = 2e-5, where the evaluation results were 0.98000, 0.97913, 0.97947, and 0.97925 respectively in terms of *Accuracy*, *Precision*, *Recall* and *F1-Score*.

Table 5. The BERT model validation results Epoch = 5.

Learning Rate	**Fold**	**Result**			
		Accuracy	***Precision***	***Recall***	***F1- Score***
2e-4	1	0.45313	0.45313	1.00000	0.62366
	2	0.50313	0.00000	0.00000	0.00000
	3	0.49375	0.00000	0.00000	0.00000
	4	0.52500	0.52500	1.00000	0.68853
	5	0.48438	0.00000	0.00000	0.00000
	AVG	0.49188	0.19563	0.40000	0.26244
3e-4	1	0.45313	0.45313	1.00000	0.62366
	2	0.49688	0.49688	1.00000	0.66388
	3	0.50625	0.50625	1.00000	0.67220
	4	0.52500	0.52500	1.00000	0.68853
	5	0.51563	0.51563	1.00000	0.68041

Learning Rate	**Fold**	**Result**			
		Accuracy	***Precision***	***Recall***	***F1- Score***
	AVG	0.49938	0.49938	1.00000	0.66574
2e-5	1	0.95625	0.93960	0.96552	0.95238
	2	0.96875	0.97452	0.96226	0.96835
	3	0.98438	0.98758	0.98148	0.98452
	4	0.99375	1.00000	0.98810	0.99401
	5	0.99688	0.99398	1.00000	0.99698
	AVG	**0.98000**	**0.97913**	**0.97947**	**0.97925**
3e-5	1	0.94063	0.95652	0.91035	0.93286
	2	0.95000	0.95541	0.94340	0.94937
	3	0.94688	0.90503	1.00000	0.95015
	4	0.98125	0.97647	0.98810	0.98225
	5	0.99375	0.99394	0.99394	0.99394
	AVG	0.96250	0.95747	0.96716	0.96171
2e-6	1	0.85625	0.80745	0.89655	0.84967
	2	0.93750	0.97279	0.89937	0.93464
	3	0.97500	0.98125	0.96914	0.97516
	4	0.99688	1.00000	0.99405	0.99702
	5	0.99063	0.99390	0.98788	0.99088
	AVG	0.95125	0.95108	0.94940	0.94947
3e-6	1	0.90625	0.90780	0.88276	0.89511
	2	0.97188	0.98077	0.96226	0.97143
	3	0.99688	1.00000	0.99383	0.99690
	4	0.99688	1.00000	0.99405	0.99702
	5	0.99688	1.00000	0.99394	0.99696
	AVG	0.97375	0.97771	0.96537	0.97148

3.2 Testing the Best Model with Unseen Data

The best identification model for Indonesian sentences containing idiomatic expressions obtained from the training and validation stages is the BERT model with *Learning Rate* 2e-5 and *Epoch* 5. Testing the best model with unseen (i.e. new) data aims to check whether the model is too fit. or not. Overfitting is a situation when a model produces high performance during training and validation but its performance decreases significantly when tested with new data. Based on the test results, the BERT model does not experience overfitting as seen from the *Accuracy*, *Precision*, *Recall* and

F1-Score values produced when testing new data, namely 0.97; 0.96; 0.98; 0.97 is not much different from the *Accuracy*, *Precision*, *Recall* and *F1-Score* values produced by the best model at the training and validation stages (See Table 5).

4 Conclusion

Validation and testing of the pre-trained BERT model for Indonesian idiomatic expressions shows that the pre-trained BERT model produces good performance even though it was trained with a limited dataset (namely 1600 sentences). Testing the best model resulted in high performance in terms of *Accuracy*, *Precision*, *Recall* and *F1 Score* (i.e. greater than 95%). The study also shows that hyper-parameter tuning has a large impact on results, especially in training on limited datasets. Changing the hyper-parameter configuration will have a major impact on model performance.

References

1. Baldwin, T. and Kim, S. N.: Multiword expressions, In N. Indurkhya and F. J. Damerau, editors, Handbook of Natural Language Processing. 2nd ed. Chapman and Hall/CRC (2010).
2. Chu, Z., Yang, Z., Cui, Y., Chen, Z., and Liu, M.: HIT at SemEval-2022 Task 2: Pre-trained Language Model for Idioms Detection. In: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pp. 221–227, Association for Computational Linguistics (2022).
3. Zeng, Z. and Bhat, S.: Idiomatic Expression Identification using Semantic Compatibility, Transactions of the Association for Computational Linguistics 9, 1546–1562 (2021).
4. Tedeschi, S., Martelli, F., and Navigli, R.: ID10M: Idiom Identification in 10 Languages. In: Findings of the Association for Computational Linguistics: NAACL 2022, pp. 2715-2726, Association for Computational Linguistics (2022).
5. Chaer,A.: Kamus Idiom Bahasa Indonesia. Ende: Nusa Indah (1993).
6. Devlin, J., Chang, M.W., Lee, K., and Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies Proceedings of the Conference, pp. 4171–4186 (2019).
7. Kachuee, S., and Sharifkhani, M.: Latency Adjustable Transformer Encoder for Language Understanding (2022). https://arxiv.org/abs/2201.03327
8. Rothman, D.: Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP & Python, PyTorch, TensorFlow, BERT, RoBERTa & more. Birmingham: Packt Publishing (2021).