Chapter 2
Continual Learning of Deep Learning for Indonesian Sentiment Analysis

Carlo Johan Nikanor, Hendri Murfi, Muhammad Adani Osmardifa, and Gianinna Ardaneswari

Abstract High-level social media usage makes this social media frequently used as one of the sources for sentiment analysis. Sentiment analysis is a field of study that analyzes people's opinions or evaluations of entities such as products and services. The Bidirectional Encoder Representation from Transformers (BERT) model is a deep learning architecture that achieves state-of-the-art performance for many natural language processing problems, including sentiment analysis. Several further developments have implemented continual learning on the deep learning model. By applying continual learning, the deep learning model continuously learns based on new data while retaining previously learned knowledge. In this paper, we analyze the performance of the BERT model for continual learning in some domains of Indonesian sentiment analysis. Then it will be compared with two standard deep learning models: fine-tuned embedding with CNN and fine-tuned embedding with LSTM. Our simulation shows the BERT model gives the best accuracy for the transfer of knowledge. However, the fine-tuned embedding with LSTM model is better for retain of knowledge. Moreover, our simulation shows that the order of the source domains affects the performance of BERT for both transfer of knowledge and retain of knowledge.

Keywords Sentiment analysis · Deep learning · BERT · Continual learning · Transfer of knowledge · Retain of knowledge

2.1 Introduction

Mobile applications have become an alternative solution for various needs during the pandemic. Bank Indonesia recorded an increase in transactions through e-commerce applications, namely, to 547 million transactions with a nominal value of IDR

C. J. Nikanor · H. Murfi (✉) · M. A. Osmardifa · G. Ardaneswari
Department of Mathematics, Universitas Indonesia, Depok, Indonesia
e-mail: hendri@ui.ac.id

C.-C. Wang, A. Nallanathan (eds.), *6th International Conference on Signal Processing and Information Communications*, Signals and Communication Technology, https://doi.org/10.1007/978-3-031-43781-6_2

88 trillion per the first quarter of 2021.[1] Shopee, Tokopedia, and Lazada are the top three e-commerce applications with the highest number of visitors per month.[2] This high level of use makes mobile applications collect many user opinions on their negative and positive features. Thus, we require machine learning for sentiment analysis on the opinion text data to provide information related to the advantages and disadvantages of the application or service of the mobile applications as a whole [1, 2].

From a machine learning point of view, sentiment analysis can be grouped as supervised learning because of sentiment labels [3, 4]. Deep learning is the primary machine learning method for unstructured data, such as text data. Deep learning extends the standard of machine learning by additional layers to extract a relevant representation of data [5]. The deep learning models widely used in sentiment analysis are convolutional neural networks (CNN) and long short-term memory (LSTM). Their efficiency and development have been mentioned in [6–8], including for Indonesian sentiment analysis [9]. The Bidirectional Encoder Representation from Transformers (BERT) model is another deep learning architecture that achieves state-of-the-art performance for many natural language processing problems [10]. BERT also improves the performance of standard deep learning for Indonesian sentiment analysis [11].

Continual learning, also known as lifelong learning or incremental learning, is the ability of a model to continuously learn based on new data while retaining previously learned knowledge [12]. The recommender systems on applications like Netflix and Amazon are well-known examples of continual learning. These applications instantly collect new labeled data as people interact with the applications. Continual learning algorithms have also succeeded in computer vision and clinical applications [13–15]. In practice, the main issue regarding continual learning is catastrophic forgetting, i.e., training a model with new information interferes with previously learned knowledge. This phenomenon typically leads to an abrupt performance decrease or, in the worst case, to the old knowledge being entirely overwritten by the new one.

In this paper, we analyze the performance of the BERT as a pretrained model of text data representation for continual deep learning in some domains of Indonesian sentiment analysis. Then it will be compared with two standard text data representations in deep learning: fine-tuned embedding with CNN and fine-tuned embedding with LSTM. Our simulation shows the BERT model gives the best accuracy for the transfer of knowledge. However, the fine-tuned embedding with LSTM model is better for retain of knowledge. Moreover, our simulation shows that the order of the source domains affects the performance of BERT for both transfer of knowledge and retain of knowledge.

[1] https://www.google.com/amp/s/ekbis.sindonews.com/newsread/472710/39/e-commerce-jadi-andalan-dongkrak-penjualan-di-masa-pandemi-162531223.

[2] https://www.webretailer.com/b/online-marketplaces-southeast-asia/.

The structure of this paper is as follows: in Sect. 2.2, we briefly explain methods. We describe the experiments in Sect. 2.3. Finally, a general conclusion about the results is presented in Sect. 2.4.

2.2 Methods

In this section, the methods used in this research will be explained. They are convolution neural networks (CNN), long short-term memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT).

2.2.1 Convolutional Neural Network

The convolutional neural network (CNN) is a deep learning model widely used in text classification. CNN uses filters to extract essential features from each region for text classification. During the process of word representation, the input will go through the convolution layer and the max-pooling layer (Fig. 2.1).

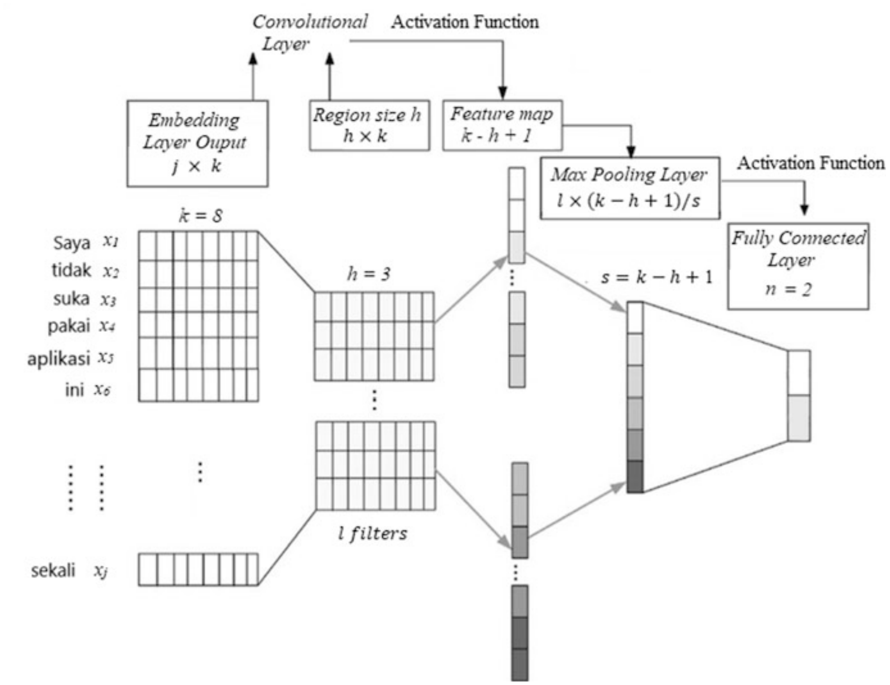


Fig. 2.1 CNN architecture

Convolution Layer In the convolution layer, the input will be processed by l filters W to find the essential features of each region with a specific region size. Suppose that the vector representation of the i-th word is denoted by x_i and the combination of the vectors of the words x_i to x_{i+h-1} is denoted by $X_{[i:i+h-1]}$. Eq. (2.1) calculates the feature vector $\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}]$ for each filter, where j and k represent the rows and columns of the matrix, and f is a nonlinear activation function. The convolution layer's output is then used as the input for the max pooling layer.

$$c_i = f\left(\sum\nolimits_{k=1}^{h}\sum\nolimits_{j=1}^{d} X_{[i:i+h-1]k,j} \cdot W_{k,j}\right) \tag{2.1}$$

Max Pooling Layer The max pooling layer processes the output of the convolution layer by taking the essential features from each feature vector $\boldsymbol{c}$, that is $\widehat{c} = \max\{\boldsymbol{c}\}$. The purpose of this layer is to reduce the dimension of the input, so the CNN will gradually learn to use less information with further iterations.

2.2.2 Long Short-Term Memory

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that aims to remember long-term information. The LSTM model has reasonable control over what information should be kept and removed at each training stage (time step) (Fig. 2.2). At the t-th time step, LSTM receives two input vectors which are the

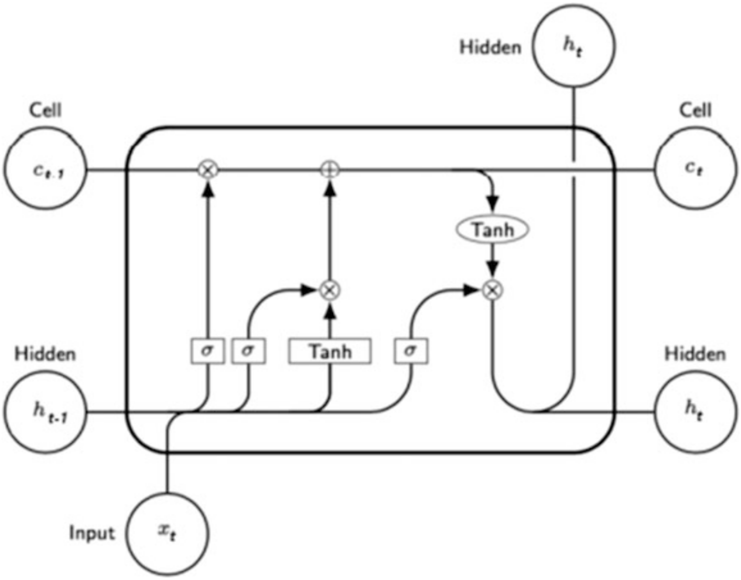


Fig. 2.2 LSTM architecture

vector representation of the t-th word in the sentence ($\boldsymbol{x}_t$) and the output vector of the previous hidden state ($\boldsymbol{h}_{t-1}$). The model will first determine what information should be removed from the cell state $\boldsymbol{C}_{t-1}$. This process is done at the forget gate ($\boldsymbol{f}_t$) shown in Eq. (2.2).

$$\boldsymbol{f}_t = \sigma\left(W_{fx}\boldsymbol{x}_t + W_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{2.2}$$

Next, the model will store selected information in the cell state $\boldsymbol{C}_t$. During this step, the model will also determine the value to be updated through the input gate ($\boldsymbol{i}_t$) as shown in Eq. (2.3) and the construction of a new vector that is the candidate cell state value ($\tilde{\boldsymbol{C}}_t$) in Eq. (2.4).

$$\boldsymbol{i}_t = \sigma(W_{ix}\boldsymbol{x}_t + W_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{2.3}$$

$$\tilde{\boldsymbol{C}}_{\boldsymbol{t}} = \tanh(W_{cx}\boldsymbol{x}_{\boldsymbol{t}} + W_{ch}\boldsymbol{h}_{\boldsymbol{t}-\mathbf{1}} + \boldsymbol{b}_{\boldsymbol{c}}) \tag{2.4}$$

The cell state $\boldsymbol{C}_{t-1}$ is updated to a new cell state ($\boldsymbol{C}_t$) using the outputs of the forget gate, input gate, and candidate vector $\tilde{\boldsymbol{C}}_t$, as shown in Eq. (2.5).

$$\boldsymbol{C}_{\boldsymbol{t}} = \boldsymbol{f}_{\boldsymbol{t}} \cdot \boldsymbol{C}_{\boldsymbol{t}-\mathbf{1}} + \boldsymbol{i}_{\boldsymbol{t}} \cdot \tilde{\boldsymbol{C}}_t \tag{2.5}$$

The last step for the model is to determine the output using the new cell state. This is done at the output gate ($\boldsymbol{o}_t$) shown in Eq. (2.6). The vector $\boldsymbol{o}_t$ will then be used with the cell state $\boldsymbol{C}_t$ to determine the hidden state $\boldsymbol{h}_t$ in Eq. (2.7). The vector $\boldsymbol{h}_t$ will be used in the next time step.

$$\boldsymbol{o}_{\boldsymbol{t}} = \sigma(W_{ox}\boldsymbol{x}_{\boldsymbol{t}} + W_{oh}\boldsymbol{h}_{\boldsymbol{t}-\mathbf{1}} + \boldsymbol{b}_{\boldsymbol{o}}) \tag{2.6}$$

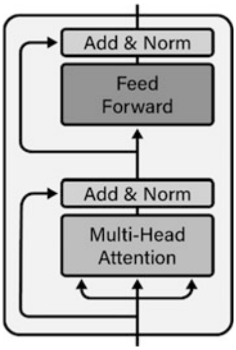
$$\boldsymbol{h}_{\boldsymbol{t}} = \boldsymbol{o}_{\boldsymbol{t}} * \tanh(\boldsymbol{C}_{\boldsymbol{t}}) \tag{2.7}$$

where $W_{\{fx, fh, ix, ih, cx, ch, ox, oh\}}$ is a weight matrix and $\boldsymbol{b}_{\{f, i, c, o\}}$ is a bias vector [16].

2.2.3 *Bidirectional Encoder Representation from Transformers*

Bidirectional Encoder Representations from Transformers, commonly called BERT, is a trained language representation model developed by Devlin et al. [10]. Unlike the current language representation model, BERT does not use the traditional left-to-right or right-to-left language model. However, BERT is designed to train a bidirectional representation that simultaneously looks at each layer's left and right contexts. The main architecture of BERT is the transformer's encoder layers (Fig. 2.3).

Fig. 2.3 Transformers encoder layer



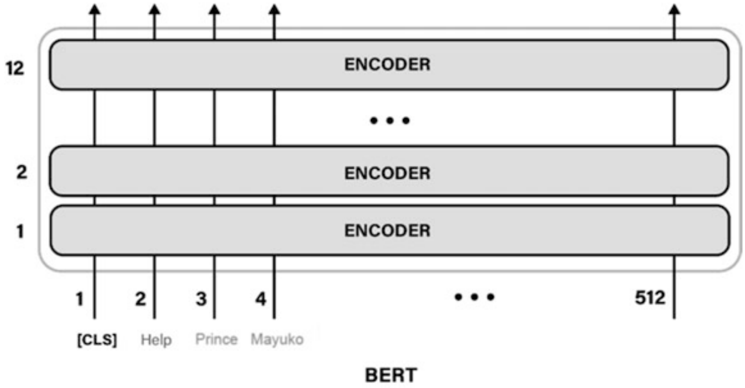


Fig. 2.4 BERT architecture

BERT comprises 12 layers of transformers encoder, each with a hidden size of 768, and the value of ⍰ in the multi-head self-attention layer is 12. The transformer encoder layer consists of two sub-layers in each layer: multi-head attention and position-wise feed-forward network (Fig. 2.4).

Multi-head Attention Multi-head attention is an architecture that simultaneously performs the attention function ⍰ times using different Query, Key, and Value matrices. The goal of multi-head attention is to generate as much as different amounts of attention for each word. As the model processes each word (each position in the input sequence), attention allows it to look at other positions in the input sequence for clues that can help better encode this word.

Position-Wise Feed Forward Network Position-wise feed-forward network is a neural network architecture used to transform the representation of all sequence positions using the same feed-forward network. The feed-forward network architecture consists of two linear transformations with a ReLU (rectified linear unit) activation function between the two linear transformations.

$$\mathrm{FFN} = \max(0, xW1 + b1)W2 + b2 \tag{2.8}$$

With x as the input vector, W_1 as weight matrices from the first layer, W_2 as weight matrices from the second layer, and b as bias.

The BERT model used in this study is IndoBERT-based uncased. IndoBERT-based uncased is the Indonesian version of the BERT model that uses uncased data during pre-training. This model has 12 layers of transformer encoder, 768 hidden sizes, and 12 heads in the attention sub-layer.

2.3 Experiment

In this section, we will describe the process of the experiment. In this study, we will implement continual learning on some domains of Indonesian sentiment analysis using BERT and then compare it to two other models, the fine-tuned embedding with CNN and the fine-tuned embedding with LSTM. The model is trained on personal computer with Intel(R) Core i7, 16GB RAM, an NVIDIA GeForce RTX 3050, and Python 3.7.

2.3.1 Data Sets

There are six data sets used in this study, shown in Table 2.1. *Calon Presiden* contains tweets about the Indonesian Presidential Elections in 2014, while E-commerce contains tweets about e-commerce's existence in Indonesia. Four of the data sets, DANA, Shopback, Grab, and Jenius, have Indonesian reviews about applications from Google Playstore.

Table 2.1 Data sets details

Data sets	Role	Negative sentiment	Positive sentiment	Total sentiments
DANA	Target domain	406	769	1.175
Calon Presiden	Source domain 1	768	1.117	1.885
E-commerce	Source domain 2	422	530	952
Shopback	Source domain 3	979	857	1.836
Grab	Source domain 4	833	755	1.588
Jenius	Source domain 5	943	876	1.819

2.3.2 Preprocessing

There are several changes applied to the text, such as capital letters being changed to lowercase, the website address is removed, the Twitter username deleted, Hashtag removed, punctuation removed, numbers being deleted, the extra spaces being removed, repeating words being separated by removing the dash, letters that are repeated more than two times are deleted into just two times, words with a single letter are removed, and the "*rt*" is deleted. The sentiment labels on the data sets are processed by one-hot encoding. Sentiment on the text has a value of -1 or 1, where -1 represents negative sentiment and 1 represents positive sentiment. Through this preprocessing, a sentiment is mapped into a two-dimensional vector. In the negative sentiment, -1, the mapping result is a vector with the first and second elements 1 and 0, respectively. On the other hand, for the positive sentiment, 1, the mapping result is a vector with the first and second elements being 0 and 1, respectively. In this study, the proportion of training data to testing data is 8:2.

2.3.3 Model Implementation

The first step in the BERT model is to change every word in the sentence input into a numerical vector representation which is then entered into the encoder layer. First, BERT uses the WordPiece model as a tokenizer to tokenize a sentence, and the addition of two special tokens, [CLS] is added at the beginning, and [SEP] is added at the end of the sentence. Padding and truncating are performed to ensure each sentence in the data has the same length of tokens. In this study, the maximum number of tokens is 256. Each document with less than 256 tokens will be padded with a special token [PAD] until the document length reaches 25 tokens, and the sentence with more than 256 tokens will be truncated only up to 256 tokens. The next step is embedding, which functions to map each token to a numeric vector with a particular dimension. Each token has three embeddings, token embedding, segment embedding, and position embedding. The illustration of embedding is shown in Fig. 2.5.

Finally, as shown in Fig. 2.3, the three vectors are added together after obtaining the numerical representation vectors of the token embedding, segment embedding, and positional embedding to obtain the input for the BERT model.

The simulation performs fine-tuning using BERTForSequenceClassification with batch sizes 16, Adam learning method, the learning rate of $2e^{-5}$, and 15 epochs. In this study, the author used early stopping that monitors validation loss with patience $=3$.

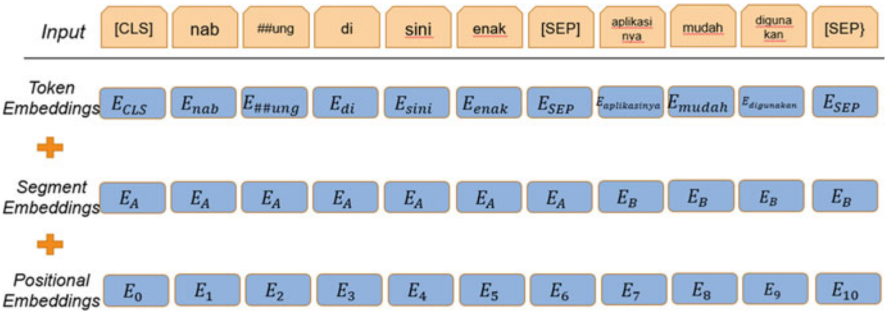

Input
[CLS]
nab
##ung
di
sini
enak
[SEP]
aplikasi nya
mudah
diguna kan
[SEP}
Token Embeddings
E_{CLS} E_{nab} $E_{\#\#ung}$ E_{di} E_{sini} E_{enak} E_{SEP} $E_{aplikasinya}$ E_{mudah} $E_{digunakan}$ E_{SEP}
Segment Embeddings
E_A E_A E_A E_A E_A E_A E_A E_B E_B E_B E_B
Positional Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10}

Fig. 2.5 Illustration of input representation for BERT

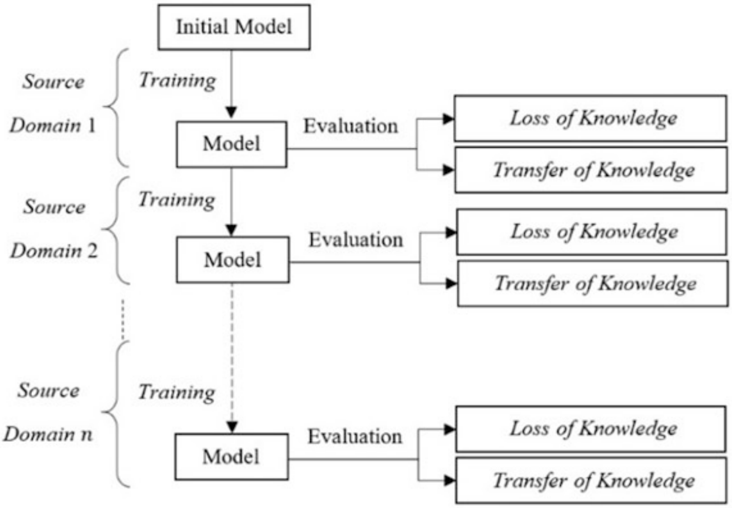


Fig. 2.6 Continual learning implementation process

2.3.4 Continual Learning Implementation

Continual learning is implemented in the model with a flowchart shown in Fig. 2.4 using five data sets. The role of the data sets can be seen in Table 2.1. Based on Fig. 2.4, after the model is built using Source Domain 1, the model continues to learn from Source Domain 2. After learning from Source Domain 2, the model is tested for retain of knowledge (loss of knowledge) by evaluating it to data testing of Source Domain 1. In addition, the model is also tested for the transfer of knowledge by considering it to the Target Domain. The following learning of Source Domains 3, 4, and 5 goes through the same steps (Fig. 2.6).

2.3.5 Transfer of Knowledge

Firstly, we simulate the performance of BERT for continual learning based on the performance of transfer of knowledge, namely, the performance of BERT in the target domain after learning in a series of source domains. Figure 2.7 compares BERT, the fine-tuned embedding with LSTM (LSTM), and the fine-tuned embedding with LSTM (CNN) for transfer of knowledge.

Based on Fig. 2.7, the BERT model increases accuracy to 89.60%. This accuracy increased by 6.6.7% from the initial accuracy of 82.93%, the LSTM model experienced an increase of accuracy to 84.51% or an increase of 19.86% from the initial accuracy of 64.65%, CNN model experienced an increase in accuracy to 85.86%, or an increase of 17.09% from initial accuracy of 68.77%. Based on these results, we can conclude that BERT provides the highest accuracy for the transfer of knowledge.

2.3.6 Retain of Knowledge

Next, we simulate the performance of BERT for continual learning based on retain of knowledge, namely, the performance of BERT in an initial source domain after learning in a series of other source domains. In this simulation, the initial source domain is set to Source Domain 1. Figure 2.8 compares BERT, the fine-tuned embedding with LSTM (LSTM), and the fine-tuned embedding with LSTM (CNN) for retaining of knowledge.

Figure 2.8 shows that the LSTM model retains more knowledge of Source Domain 1 than BERT and CNN. The LSTM model experienced a decrease in accuracy to 83.63% or as much as 2.37% from the initial accuracy of 86.00%, BERT model experienced a reduction in accuracy of up to 80.96% or as much as 8.21% from the initial accuracy of 89.17%, and CNN model experienced a decrease in accuracy to 72.63% or as much as 13.92% from the initial accuracy of 86.55%.

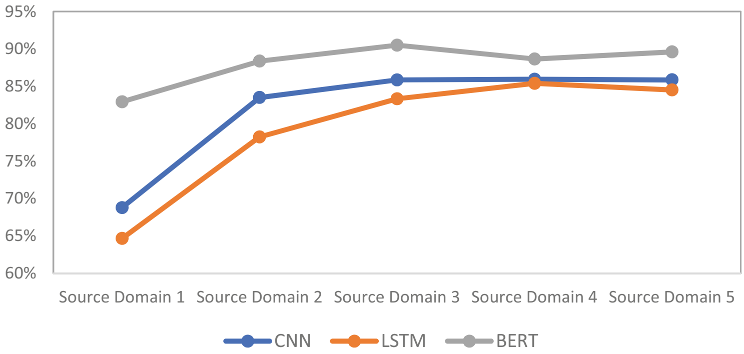


Fig. 2.7 The accuracies of BERT, LSTM, and CNN for transfer of knowledge

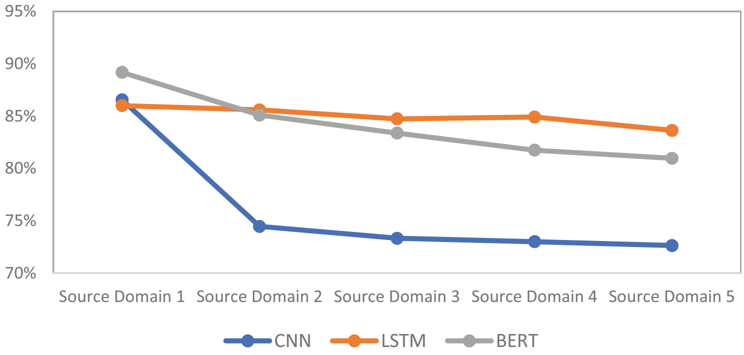


Fig. 2.8 The accuracies of BERT, LSTM, and CNN for retain of knowledge

Table 2.2 The top five highest accuracies of BERT for transfer of knowledge

The sequence of source domain	Accuracy (%)
1-5-4-2-3	91.66
3-1-5-2-4	91.57
5-1-3-2-4	91.49
5-2-3-1-4	91.49
3-5-1-2-4	91.49

Table 2.3 The top five highest accuracies of BERT for retaining knowledge in the source domain of Calon Presiden

The sequence of source domain	Accuracy (%)
1-5-3-2-4	83.29
1-5-2-3-4	81.70
1-2-4-5-3	81.43
1-3-5-4-2	81.42
1-2-3-4-5	80.96

2.3.7 Sequences of Source Domains

Further experiments were conducted on the 120 possible combinations of sequences of the 5 source domains. The experiments aim to see whether or not the order of source domains impacted the accuracy of the BERT model for lifelong learning.

Table 2.2 shows the top five BERT accuracies for transferring knowledge. The highest overall accuracy for transferring knowledge with the BERT model is achieved with the domain sequence of 1-5-4-2-3 and an accuracy of 91.66%. An improvement of 2.06% from an earlier experiment with the sequence of 1-2-3-4-5 that had an accuracy of 89.6%. These simulations show that the order of the source domains affects the performance of BERT for the transfer of knowledge.

We use five scenarios to simulate retain of knowledge, where each source domain becomes the initial source domain. Tables 2.3, 2.4, 2.5, 2.6, and 2.7 show the five highest BERT accuracies for retaining knowledge in each initial source domain. Based on Table 2.3, the highest accuracy of BERT in the source domain Calon

Table 2.4 The top five highest accuracies of BERT for retaining knowledge in the source domain of E-commerce

The sequence of source domain	Accuracy (%)
2-1-4-3-5	90.58
2-3-4-1-5	90.58
2-4-3-1-5	90.58
2-3-1-5-4	90.42
2-1-3-4-5	89.53

Table 2.5 The top five highest accuracies of BERT for retaining knowledge in the source domain of Shopback

The sequence of source domain	Accuracy (%)
3-4-1-2-5	95.65
3-4-1-5-2	95.11
3-1-2-4-5	94.84
3-5-1-4-2	94.84
3-4-5-1-2	94.57

Table 2.6 The top five highest accuracies of BERT for retaining knowledge in the source domain of Grab

The sequence of source domain	Accuracy (%)
4-2-5-1-3	95.28
4-5-1-2-3	93.71
4-1-2-5-3	93.40
4-2-1-5-3	93.08
4-2-5-3-1	92.77

Table 2.7 The top five highest accuracies of BERT for retaining knowledge in the source domain of Jenius

The sequence of source domain	Accuracy (%)
5-3-1-4-2	97.53
5-3-4-1-2	96.43
5-4-1-3-2	96.15
5-1-3-4-2	95.60
5-1-4-3-2	95.60

Presiden was obtained after studying a series of other source domains, with the order of 1-5-3-2-4 being the highest, with an accuracy of 83.29%. In the source domain of E-Commerce, the highest accuracy is 95.65%, provided by the source domain sequence of 3-4-1-2-5 in Table 2.4. For the rest, the source domains of Shopback, Grab, and Jenius, the highest accuracies resulted from the source domain sequences of 3-4-1-2-5, 4-2-5-1-3, and 5-3-1-4-2, respectively.

These simulations also show that the order of the source domains affects the performance of BERT in retaining knowledge. Moreover, there is no correlation between the order of sources domains that give the highest accuracies for both transfer of knowledge and retain of knowledge.

2.4 Conclusion

In this paper, we analyze the performance of the BERT model for lifelong learning in Indonesian sentiment analysis. Then it will be compared with two standard deep learning models: fine-tuned embedding with CNN and fine-tuned embedding with LSTM. Our simulation shows the BERT model gives the best accuracy for the transfer of knowledge. Lifelong learning increases the accuracy by 6.67% from the initial source domain to the last source domain and achieves the final accuracy of 89.60%.

The fine-tuned embedding with CNN model is the second with a final accuracy of 85.86%, followed by the fine-tuned embedding with LSTM with 84.51%. However, the fine-tuned embedding with LSTM model is the best model for retain of knowledge. The fine-tuned embedding with LSTM model's final accuracy is 83.63%, while the BERT model only has a final accuracy of 80.96%. Moreover, our simulation shows that the order of the source domains affects the performance of BERT for both transfer of knowledge and retain of knowledge. There is no correlation between the order of sources domains that give the highest accuracies for both transfer of knowledge and retain of knowledge.

References

1. B. Liu, L. Zhang, A survey of sentiment analysis and opinion mining, in *Mining Text Data*, ed. by C. Aggarwal, C. Zhai, (Springer, Boston, 2012), pp. 413–463
2. B. Liu, Sentiment analysis and opinion mining, in *Synthesis Lectures on Human Language Technologies*, (Morgan & Claypool Publishers, San Rafael, California, 2012)
3. M. Mohri, A. Rostamizadeh, A. Talwalkar, *Foundation of Machine Learning* (MIT Press, Cambridge, 2018)
4. Z. Lipton, M. Li, A. Smola, A. Zhang, Dive into deep learning, https://d2l.ai/. Accessed 01 June 2022
5. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016)
6. Y. Kim, Convolutional neural networks for sentence classification, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, (Association for Computational Linguistics (ACL), Doha, 2014), pp. 1746–1751
7. A. Hassan, A. Mahmood, Deep learning for sentence classification. In: IEEE Long Island Systems, Applications and Technology Conference (LISAT), New York (2017)
8. P.M. Sosa, Twitter sentiment analysis using combined LSTM-CNN models. In: Academia.Edu, pp. 1–9 (2017)
9. T. Gowandi, H. Murfi, S. Nurrohmah, Performance analysis of hybrid architectures of deep learning for Indonesian sentiment analysis, in *Soft Computing in Data Science. SCDS 2021. Communications in Computer and Information Science*, ed. by A. Mohamed, B.W. Yap, J.M. Zain, M.W. Berry, vol. 1489, (Springer, Singapore, 2021), pp. 18–27
10. J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proceedings of the 2019 Conference of the North American Chapter of Association for Computational Linguistics: Human Language Technologies*, vol. 1, (Association for Computational Linguistics, Minneapolis, 2019), pp. 4171–4186

11. H. Murfi, T. Gowandi, Syamsuriani, G. Ardaneswari, S. Nurrohmah, BERT-based combination of convolutional and recurrent neural network for Indonesian sentiment analysis. arXiv:2211.05273 [cs.CL] (2022). https://doi.org/10.48550/arXiv.2211.05273
12. G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review. Neural Netw. **113**, 54–71 (2019)
13. C.S. Lee, A.Y. Lee, Clinical applications of continual learning machine learning. Lancet Digit Health **2**(6), e279–e281 (2020). https://doi.org/10.1016/S2589-7500(20)30102-3
14. M. Lenga, H. Schulz, A. Saalbach, Continual learning for domain adaptation in chest x-ray classification. Proc. Third Conf. Medi. Imaging Deep Learn. PMLR **121**, 413–423 (2020)
15. D. Kiyasseh, T. Zhu, D. Clifton, A clinical deep learning framework for continually learning from cardiac signals across diseases, time, modalities, and institutions. Nat. Commun. **12**, 4221 (2021). https://doi.org/10.1038/s41467-021-24483-0
16. T. Ganegedara, *Natural Language Processing with TensorFlow* (Packt Publishing, Mumbai, 2018)