DEIM Forum 2023 1a-1-2

Label-dependent Paragraph Selection for Efficient Neural Long Document Classification

Tan-Ning HUANG† and Makoto P. KATO†

† University of Tsukuba
1-2 Kasuga, Tsukuba, Ibaraki 305–8550, Japan
E-mail: †s2126087@s.tsukuba.ac.jp, ††mpkato@acm.org

Abstract BERT is incapable of effectively processing long documents due to its quadratically-growth self-attention mechanism. A common solution can be truncating documents under BERT's maximum input length, but it will be considered to give negative effects on classification performance. To address this problem, we propose several approaches let BERT capable to handle long documents. We first extract paragraphs by matching phrases identical or associated with the target label, therefore truncating a 512-word paragraph representing the original full-length document. In extend of our truncating operation, we introduce a machine-learning based paragraph scoring mechanism. By predicting the classification confidence probability of several paragraph-equivalents, our methods are able to rate the effectiveness between each paragraph-equivalents by a regression analysis model. Aggregated from a Japanese novel self-publishing website as our task dataset, we deploy several classification tasks to compare these approaches with baselines. Experimental results suggested our approaches show a notable improvement in performance if under a 512-word limitation. However, non-neural network models with full-length document input have the highest performance up to this point.

Key words Long document classification, Application of natural language processing, Language model

1 Introduction

The fast-growing information-based society not only gives us more opportunities to fulfill our needs but also overfilled us with irrelevant noises. In 2022, Japan has published over 68,000 books across multiple domains [1]. On the other hand, a Japanese will only read around 15 16 books annually on average [2], which is 0.022% by published books. Therefore, it is important for readers to find the proper book that they are willing to flip through. Nonetheless, modern public access catalog systems commonly need users who already know what they are looking for and are barely available to suggest topic-related items. These kinds of systems are generally distinct from humans' search behavior, thus we need to heavily rely on document ranking methods for recommending unknown titles. Document classification by natural language processing has become a familiar approach to accompany information retrieval systems. It has been proven by several domains such as legal [3] or biology domain [4]. These kinds of documents are often found within large corpus with complex contexts, making them challenging from conventional statistical models.

Bidirectional Encoder Representations from Transformers, BERT, is a state-of-the-art natural language processing model which successfully applied to various tasks, including document classification. By its self-attention mechanism, BERT is able to effectively apprehend the context and meaning of words. This is critical to document-level classification since it requires the classifier to understand the context between phrases. [5] Another benefit to utilizing BERT on document classification tasks can be praised as its availability to effectively handle long-range dependencies, where BERT can understand the relationships between phrases that are sparse in the document. This is important because the purpose of a document depends on the relationships between phrases throughout the entire document. [6] However, by BERT's transformer architecture and self-attention mechanism, it suffers in a $O(n^2)$ complexity in memory and time, which quadratically grows with the length of the input sequence. [7] In general, the maximum input length of BERT models is about several hundred tokens, such that the base version of BERT can only process its input sequences up to 512 tokens. [6] This makes BERT models unable to effectively handle long documents.

To address this issue, we propose our approach for adapting BERT to operate on long documents, this involves truncating long documents by a 512-word paragraph to meet the maximum input limitation of BERT. However, if we truncate

the documents without considering their context will lead to a decline in classification performance. [8] As a result, it is necessary for us to confirm the truncated paragraphs are able to preserve the authentic context of the document. Therefore we identified three requirements for our proposed method: (1) Truncating the document to a maximum of 512 words; (2) Preserving the original context of the document; (3) An efficient truncating operation. It is important to note that we do not trim the documents at the token level in order to avoid using the vectorizer during the truncating operation.

Our approach is based on the hypothesis that, if a paragraph containing lexicons identical or similar to the classification target, should keep the same context as the original document. Therefore, we can reduce the document's contents by the section having similar lexicons. For implementation, we propose two basic truncating operations: **First-Match(FM)** and **Nearest-K (NK)**. **FM** traverses the document from the beginning and searches for phrases identical to the target label. A 512-word paragraph is then selected after the identified phrase. However, **FM** is unable to apply to all documents due to the searching phrases needing to be equivalent to the target lexicon. To address this limitation, we then propose the **Nearest-K (NK)** truncation operation. Based on Word2Vec, **NK** forming a similar-vocabulary list and identifying the phrases that appear in the document. The size of the list is represented by variable K. The **NK** method will then truncates the paragraph that has the most occurrences of these matched similar phrases.
In addition to our hypothesis-based approaches, we merge our truncating operations with machine-learning approaches. We introduced **Confidence-Learning (CL)** method, which is a bottom-up instance-based approach. The **CL** method involves tagging multiple truncated paragraph-equivalents with the confidence-score ranked by a fine-tuned language model. Therefore, by training a regression model, we can predict each paragraph's possible effectiveness for the classification task.

Our long documents dataset was collected from Kakuyomu (Hatena, co ltd., kakuyomu.com), and consists of 69,627 documents with an average length of 127,000 words across 40 labels. We conduct our experiment using DocBERT [9], a BERT variant for document-level classification that also shares a maximum input length of 512 tokens. To verify our proposed truncating operations, we include two baseline operations: **First-512 (F512)** and **FULL** document. The **F512** method involves simply truncating paragraphs from the beginning at 512 words, while the **FULL** method contains the entire document without any truncation. Our experimental result shows that, when operating under the 512-word limitation, our proposed methods significantly improve classification accuracy. However, at present, the **FULL** document still can perform the highest classification accuracy.

In this paper, we present and compare various document truncating methods for Japanese fiction literature documents. Our specific contributions include:

（1） In order to avoid BERT's maximum 512-token input limitation, we proposed several truncating methods to handle long documents.

（2） To improve the accuracy of document classification tasks, we apply a machine-learning strategy to distinguish effective paragraphs.

In Section 2, we will discuss related work and compare the similarities and differences between our approaches. Our approaches will be presented in Section 3. Section 4 will discuss our experiment details and findings.

2 Related works

Since its publication in 2018, Bidirectional Encoder Representations from Transformers (BERT) has become a state-of-the-art model and has been applied to multiple NLP tasks, including document classification. Based on its multi-layer Transformer encoder, BERT enables a pre-training mechanism that allows the language model effectively learn from a large-scale corpus before fine-tuning for a more specific task. [10] During this process, the number of input tokens for BERT is typically limited to a fixed number, usually 512 tokens. [6] Since BERT's attention mechanism does not only focus on fixed positions or special tokens but broadly references the whole sentences in lower layers. [11] This makes BERT challenging to train on longer documents due to the suffering from a $O(n^2)$ time complexity. Several approaches working on this attention mechanism of BERT have been conceived to allow longer documents exceeding 512 tokens, therefore being able to process by BERT-like models. Longformer [7] and Big bird [12] introduced the sliding window attention mechanism, where instead of full-length global attention, Longformer and Big bird allow local-focused attention pattern that is available for linear complexity. Both Longformer and Big bird can process input sequences with up to 4,096 tokens, which is sufficient for most classification tasks, but they still hold a fixed-length input limitation which is a drawback for processing overly-long documents.

A common approach to handling these kinds of documents without modifying the language model is to divide them into smaller segments.: Kong et al. [13] and Khandve et al. [14] proposed approaches that divided documents into smaller segments that could be processed by BERT individually. With statistics or neural networks, they managed to coordinate classification results based on output aggregated from segments. Another similar approach presented by He et

al. [15] combined segmentation with a recurrent neural network. Their model only focuses on significant phrases rather than the whole document, allowing for more efficient processing of the input. The encoding produced by the model's local attention mechanism is then processed by a recurrent neural network decoder in a sequential manner. Although segmentation is effective at handling long documents, its disadvantages can be recognized in that the model still needs to process the entire document, resulting in a high time cost in total when handling extremely long documents.

BERT is pre-trained on general text corpora, such as BookCorpus [16] and Wikipedia. This makes BERT ideal for cross-domain language processing tasks but lacking task-specific or domain-related knowledge. [17] This makes BERT inefficient on some specific tasks which rely on domain-related knowledge. Legal documents, differ from the corpora BERT was originally pre-trained on, are often exceed the maximum input sequence length limitation. Limsopatham et al. [8] utilized different approaches, including simple truncation, segmentation, and the use of Logformer and Big bird, to apply BERT-based model to the legal domain. They found that Longformer and Big bird can achieve better performance on long legal document classification tasks, while simply truncating or discarding part of the document without regarding the document's context can result in unsatisfactory performance. In general, utilizing language models that are specifically designed to handle longer documents can be a practical solution for handling long document classification tasks. However, this does not fully resolve the problem of maximum input length, as the input is still limited to a particular extent.

3 Methodology

3.1 Reseatch Question

As we mentioned before, BERT can not effectively handle long documents that exceed its maximum input sequence length of 512 tokens. Consequently, we need to truncate long documents under BERT's input limitation, at a maximum of 512 tokens, to apply BERT in the long-document classification task. Regardless, truncating the document into a shorter version without any consideration will result in poorer performance due to the loss of contextual information. [8]

Therefore, our research question can be expressed as:

RQ:

How can contextual information of the long document be preserved while truncating into a shorter paragraph?

As our intent, we need to find a solution to obtain a paragraph that not only fulfills the 512 tokens input limitation but also keeps the contextual information unharmed. In general, the goal of our research is to find a truncating operation that maximizes classification accuracy, therefore, is effective for document classification tasks.

3.2 Hypothesis

To address the research question, we proposed our experiment methods based on two hypotheses:

H1:

If the truncated paragraph contains the same phrase as to target label, the contextual information remains intact.

H2:

If we accept phrases whose lexical is similar to the target label, it is also as effective as the phrase we use in H1.

Therefore we proposed two truncation operations, **First-Match (FM)** and **Nearest-K (NK)** to test our hypotheses. We also tried to combine our truncating operation with a machine-learning strategy to make advancements in our proposed methods:

H3:

If we label truncated paragraphs with confidence-scores generated by a fine-tuned BERT model, we can determine the expression of which paragraph will have higher classification performance by a machine-learning approach.

Here we define the confidence-score as **"The possibility of correctly classifying the document evaluate by a fine-tuned BERT model."** Therefore, the confidence-score should be a serial number from 0.0 to 1.0, for us to conduct a regression model to train on the relation between confidence-scores and truncated paragraphs. We implemented this approach as the **Confidence-Learning (CL)** method.

3.3 Definition

Based on our hypothesis, we can define our research question as, by giving any long document d and target label l, we are able to find a truncating operation $\text{Trunc}(d, l)$ that produces a shortened paragraph $p_{(d,l)}$ of the original document, which is a sub-string of d, the truncating operation is different by each proposed operation, we will caption them in later subsections.

$$p_{(d,l)} = \textbf{Trunc}(d, l) = d_{[\textbf{start}:\textbf{end}]}$$

Where $d_{[\textbf{start}:\textbf{end}]}$ is a sub-string of d beginning from the positional index **start** while closing at index **end**.

As our research question suggests, each truncated paragraph p should obtain the contextual information of the original document. The contextual information is defined as **"If**

the target label can be recognized in the original document." Thus, the contextual information should be a boolean variable indicating the condition that if the paragraph, or document, is labeled by target label. This suggests that the truncated paragraph $p_{(d,l)}$ should contain the same classification result as how the original document is labeled, implies that the classification task $\mathbf{Cls}(p,l)$ is a binary classification task:

$$\mathbf{Cls}(p,l) = \begin{cases} 1 & \textbf{if } f(p,l) > 0.5 \\ 0 & \textbf{otherwise} \end{cases}$$

$$f(p,l) = \mathbf{sigmoid}(W \cdot \mathbf{BERT}_l(p)_{[\mathbf{CLS}]})$$

3.4 Truncating operations

As the previous section suggested, blindly truncating documents by head or tail is been proven to have poorer effectiveness on the classification tasks due to the loss of contextual information. [8] Hence, our truncating operations should meet the following criteria:

（1） The truncating operation can be applied to all documents.

（2） The truncating operation should keep the contextual information of the original document unchanged.

（3） The truncating operation should be efficient and able to avoid high computation costs.

In the next subsection, we will discuss our baseline truncating operation, followed by our proposed methods. At last, we will introduce a machine-learning aggregated approach.

3.4.1 Baseline

In our baseline truncating operation, we consider two methods: **Front512 (F512)** and **FULL**. Represent the mindless truncating operation and the ideal scenario if not impossible. By the **F512** truncating operation, we simply truncate any phrases by the first 512 words that appeared at the beginning of the document. Therefore, we define our F512 operation as:

$$\mathbf{Trunc}_{f512}(d,l) = d_{[1:j]}$$

Here, the positional index **start** and **end** are fixed numbers for representing the 1st and 512th words of the document. The value of j is varied by each document but will always locate at the final character of the 512th word. **F512** is believed to be the easiest and least complex operation with $O(1)$, however, it is considered to have the most tremendous information loss during the operation.

On other hand, in **FULL** scenario, we use the full-length document without any truncating operation.

$$\mathbf{Trunc}_{full}(d,l) = d_{[1:|d|]}$$

In practice, we do not need to make any modifications to apply **FULL** document for classification, consequently, the information loss of this method should be minimal. However, the extra-long document length will cause extensive computation costs, where the input length of **FULL** document is unbearable for most of the neural network-based language models.

3.4.2 First-Match (FM)

Based on our first hypothesis, the **FM** method traverses the document from beginning to end searching for the first phrase that is identical to the target label. We then truncated a 512-word paragraph after the matching phrase. If no phrase is matched in the document, we select **F512** paragraph instead. First, we introduce W_l, which is a set of phrases representing target label l, which is manually aggregated. For example, the W_l for the target label "超能力 (id:42)" consists of phrases including "超能力", "能力", "異能力", "異能", etc. Therefore if any of these phrases in this list is matched, we consider it as matching the target label. Accordingly, we are able to define truncating operation **FM** as follows:

$$\mathbf{Trunc}_{fm}(d,l) = d_{[\mathbf{fm}:\mathbf{fm}+j]}$$

Where **fm** is the positional index generated by the matching process:

$$\mathbf{fm} = \mathbf{min}\{i | d_{[m:m+n]} \in W_l\}$$

This method had moderate complexity, while the number of application documents can vary by the target label since the phrase in the document must be identical to the target label list as a requirement.

> Example for a Paragraph
> Truncated by FM with Label ID 96（戦争）:
>
> 戦争は終わらずにずっと続いていたのだった。ようやく目的地に着いた。風にはためく軍旗が誇らしげに掲げられ、扉の両脇は長槍を持った軍服姿のケット・シーが固めている。旗の意匠は氷の結晶に渦巻く吹雪と交差する槍。リスティンキーラ軍司令部だ。俺が帰還することは通達済みなため、特にトラブルもなく入口を通過。目的地である三階を目指して階段を登っ…
>
> デウス・エクス・マキナを殺せ, 2020, ほりえる [18]

There are several variations of the **FM** operation: **Last-Match (LM)** used the last matched vocabulary for truncation. **FM-Front-and-Back (FM-FNB)** truncated the paragraph before and after the matching phrase, respectively.

3.4.3 Nearest-K(NK)

To address the problem that **FM** may have led to a deficiency of applicable documents, we proposed our second approach with the **Nearest-K** operation. In short, we compared the similarity between the target label and each vocabulary appearing in the document, then truncated a 512-word

paragraph having the most related phrases. To be more specific, based on each target label and its corresponding sub-collection dataset, we broke up the vocabulary in all documents. Behind that, we used word2vec to calculate the relation score between the target label and each phrase to create a list of similar vocabularies. We then selected the top-K similar vocabularies by the size of K and compared each phrase within a document to find the paragraph where the top-K vocabularies appeared the most. In this experiment, we only built a similar vocabulary list using nouns since they tend to show the most significant differences by each label.

We calculate the similarity between words w that appeared in the document to target label l by the cosine similarity, therefore:

$$\mathbf{sim}(w,l) = \frac{\mathbb{e}_w \cdot \overline{\mathbb{e}_l}}{\|\mathbb{e}_w\| \, \|\overline{\mathbb{e}_l}\|}$$

$$\overline{\mathbb{e}_l} = \frac{1}{|W_l|} \sum_{w \in W_l} \mathbb{e}_w$$

Hence, we are able to acquire the similar word set $S_l^{(k)}$ as the set of k-words, corresponding to W_l, ranked by similarity. By these premises, we defined the **NK** operation as:

$$\mathbf{Trunc}_{nk}(d,l) = \mathbf{argmax}_{p \in P_d} \sum_{w \in S_l^k} tf(p,w)$$

$$p \in P_d = \{d_{[i:i+j]} | 1 \leqq i \leqq |d| - j\}$$

The **NK** approach is a more complex operation, since not only did **NK** is needed to create a similar vocabulary list, but it also requires applying K words to each document for the matching process. However, **NK** had the benefit that almost all of the document could be found with a "related" 512-word paragraph if K size is large enough. In general, **N-100** (K = 100) provided a good balance between time requirements and application documents.

> Example for a Paragraph
> Truncated by N-100 with Label ID 204（勇者）:
> $S_{l_{42}}^{100}$ = { 勇者、魔王、戦士、英雄、賢者、ラスボス…}
>
> 不安な表情を浮かべる少年に、アリコが自己紹介をする。「申し遅れました。私は王国騎士隊の曹長、アリコと申します。あなた様のお名前は？」「……僕はユウトです」——この後、ユウトはアリマン王、フリル大臣と謁見した。伝説の勇者であるか？を確認するためにユウトは勇者の装備を装着したが、勇者の装備を装着して軽々と動くユウト…
>
> 伝説の勇者様 !? ……いえ、補助職の僧侶です,
> 2020, 田山照巳 [19]

3.4.4 Confidence-Learning(CL)

In previous sections, we propose **FM** and **NK**, two basic truncating operations based on the hypothesis that the contextual information may be preserved by having identical or similar phrases to the target label. In this section, we would like to discuss another truncating operation based on a machine-learning approach. **CL**, an instance-based paragraph selection method aggregating truncating operation and machine learning. We first truncate documents by paragraph alternatives, different paragraphs truncated from the same document, as basic. For example, the paragraph alternatives of **FM**, **FM(top-10)** will be the 10 paragraphs in which the matching phrase first appears. **N-100(top-10)** will be the first 10 paragraphs which applied to **N-100**, ranked by score from high to low.

By having multiple paragraphs truncated from the same document, in an ideal situation, every alternative paragraph should have the same context as the original document. However, in practice, the result by the classifier might vary due to the variation between each paragraph does not have the same context. Therefore, if we were able to comprehend the expression between "good paragraphs" and "bad paragraphs", we should be able to distinguish which expression from a paragraph is sufficient to document classification tasks.

To be more specific, we trained our **CL** application using these procedures:

（1） We obtained paragraph alternatives by a truncating operation. (**FM**, **NK**, etc.)

（2） With a fine-tuned BERT classifier, we embedded the confidence-score with each alternative.

（3） Combining the confidence-score with paragraph alternatives, we trained a regression model to predict the confidence-score.

The confidence-score is the possibility of correctly classifying the document by a fine-tuned BERT model. For example, if we have a document that has the contextual information of the target label id 42 (l_{42}), therefore by having two paragraph alternatives truncated by **N-100**: p_0 as the 1st result return from $\mathbf{Trunc}_{nk}$ and $p1$ as the 2nd, while can assume that these two paragraphs should have the same contextual information $\mathbf{Cls}(p_0, l_{42}) = \mathbf{Cls}(p_1, l_{42})$. Regardless, if we apply a fine-tuned BERT classifier, we may find out that $f(p_0, l_{42}) = 0.97$ and $f(p_1, l_{42}) = 0.51$, indicates that BERT has higher confidence to classify p_0 than p_1 as label l_{42}. As result, we can assume that the expression of p_0 is superior to p_1, therefore having a higher confidence score.

Therefore, we define our **CL** operation as:

$$\mathbf{Trunc}_{cl}(d,l) = \underset{p \in Pd}{\mathbf{argmax}}(\mathbf{Regression}_l(p))$$

$$p \in P_d = \{d_{[start:end]} | \mathbf{Trunc}(d,l)\}$$

In our experiment, we generated 10 truncated paragraph alternatives using the **N-100** method, ranked by the number of similar vocabulary words. If a document did not have at least 10 alternatives, we randomly selected a 512-word paragraph until we retained a total of 10. We then used a fine-tuned BERT model trained with **FM-FNB** for the language model by step (2). In step (3), we utilized a Rigid Regression model and Logistic Regression models to train our paragraph alternatives by target confidence-scores.

4 Experiment

In the previous section, we addressed the research problem of adapting BERT for long document classification. Here, we present our approaches to addressing the problem: we proposed two truncating operations for creating shortened paragraphs from the original document, with the aim of improving classification accuracy compared to baseline approaches. In this section, we would first provide an overview of the dataset used in this study (Section 4.1). Therefore We will describe the general experiment setup (Section 4.2), followed by the findings from our proposed approaches (Section 4.3).

4.1 Dataset

The difference between literature fiction documents and others is obvious, literature fiction has more literary context, is usually based on imagination, and is more emotional. [20] It is also important to note that literary fiction is usually much longer than other documents. For example, news articles can range from hundreds to thousands of words [21], online review comments are typically around 300 [22], while longer documents such as legal-domain documents [8] and academic papers [14] are around 10,000 words. On the other hand, literary fiction can vary in length depending on the genre or author but is generally around 20,000 to 40,000 words for a short story and 120,000 to 180,000 words for a long story. [23]

To address the unique characteristics of literature fiction documents, we designed our dataset based on Kakuyomu (Hatena, co ltd., kakuyomu.com), an online novel publishing service that allows users to publish their personal works. There are several benefits to using fiction works gathered from Kakuyomu as our dataset: (1) all works are published openly and cover a wide range of genres and characteristics; (2) most of the works are labeled and classified in detail by the authors; (3) the length of each work is considerably lengthy.

We acquired 69,627 fiction works with a total of 237 labels, each work by length of 127,000 words on average. We then excluded low-priority labels for which the number of suggested works was less than 1,000, resulting in a dataset containing 40 labels with its own sub-collection. To create a balanced sub-collection, we selected all fiction works with the target label, as well as an equal number of randomly chosen works from other sub-collections that were not classified by the target label. We then partitioned each sub-collection into three training sets in an 8:1:1 ratio for training, validation, and testing, respectively.

4.2 Experiment setup

The aim of our experiment is to create a document classification task using a 512-word paragraph dataset as training data. This dataset was based on the original full-length dataset and is organized according to the label. In general, our document classification task was dependent on each label, with the classification target being a binary outcome. As we were training for a binary classification task, the language model's predictions were limited to two classes: True or False. Consequently, the values of the evaluation indices (precision, accuracy, f1 score) were the same. In this paper, we would use accuracy for the evaluation index.

More specifically, our experiment consists of four steps:

（1） Extracting a sub-collection dataset based on the target label.

（2） Truncating each document in the sub-collection using our truncation method.

（3） Training the truncated sub-collection with a language model.

（4） Verifying the accuracy of the training results.

In our experiment, we applied several language models including neural network models such as DocBERT [9], Kim-CNN [24], HAN [25] and Reg-LSTM [26], as well as statistical models like linear regression, support vector machine, and naive Bayes. However, we primarily used DocBERT as the main model for our experiment.

4.3 Experimental results

The results of our approaches were shown in Table 1. The vertical axis represents the truncation method applied to the documents, while the horizontal axis represents the language model used for training. The values in each cell represent the average accuracy across all 40 labels. As previously mentioned, the **FULL** documents are too long to be processed by neural network models, so there are no values for this condition in Table 1.

Our results indicate that the BERT model consistently outperforms the other language models in terms of accuracy. However, there is not a clear difference in performance between neural network models and statistical models, except for BERT and naive Bayes. When considering only para-

Table 1 Results for Baseline, FM and NK

	BERT	Kim-CNN	HAN	Reg-LSTM	LR	SVM	B-NaiveBayes	M-NaiveBayes
F512	0.7162	0.6626	0.6875	0.6806	0.6920	0.6839	0.6828	0.6776
FULL	-	-	-	-	**0.7593**	**0.7683**	0.6406	0.6799
FM	0.7365	0.7064	0.7222	0.7135	0.7168	0.7115	0.7096	0.6991
FM-FNB	**0.7457**	0.7013	0.7151	0.7170	0.7161	0.712	0.7081	0.6979
LM	0.7400	0.7072	0.7209	0.7081	0.7084	0.7055	0.7013	0.6956
LM-FNB	0.7430	0.7029	0.7191	0.7085	0.7109	0.7071	0.7001	0.6995
N-10	0.7404	0.7002	0.7130	0.7024	0.7201	0.7155	0.7128	0.6928
N-20	0.7382	0.7016	0.7170	0.7146	0.7206	0.7201	0.7125	0.6956
N-30	0.7393	0.7006	0.7074	0.7214	0.7204	0.7148	0.7128	0.6964
N-100	**0.7455**	0.7017	0.7085	0.7128	0.7239	0.7205	0.7124	0.7005

graphs with a maximum of 512 words, our proposed methods all have higher performance than the baseline **F512**, although the methods using **FULL** document input still have the highest accuracy, even without the BERT model. When comparing the different truncation methods, there is no significant difference in performance between them However, **FM-FNB** and **N-100** do show slightly better results.

Based on the results of our experiment, we can draw the following conclusions:

H1:

If the truncated paragraph contains the same phrase as to target label, the contextual information remains intact.

Our hypothesis, H1 is confirmed. Since all of the **FM** variants can perform a better classification accuracy compared to baseline **F512** in all language models, suggesting that including phrases equivalent to the target label can give a positive effect on training the language model. However, the result that **FULL** document input is able to outperform our truncating operation, implies that we still experience some loss of contextual information or noise from negative samples.

H2:

If we accept phrases whose lexical is similar to the target label, it is also as effective as the phrase we use in H1.

We can partially confirm hypothesis (H2) by our experiment result. Just as **FM**, **NK** show a similar result when compared to baselines, implying that similar phrases can be equivalent to the target label phrases we use in **FM** operation. However, we cannot verify the distinction through **FM** and **NK**, with only a slight accuracy difference, thus it is not clear whether using similar vocabulary can actually have an impact on classification accuracy since phrases identical to the target label is also included in the similar phrases list.

H3:

If we label truncated paragraphs with confidence-scores generated by a fine-tuned BERT model, we can determine the expression of which paragraph will have higher classification performance by a machine-learning approach.

Table 2 Results for CL

	Ridge	LR	Ideal
N-100(top-10)	0.7465	0.7432	0.9349

The results for **CL** are presented in Table 2. All of the language models used in this experiment were trained with the **N-100** method for 10 paragraph alternatives. The values in each cell represent the accuracy of the **CL** methods in selecting the correct paragraph based on the highest predicted confidence-score. An ideal situation is also included in the table to show whether any of the 10 variants can be correctly classified.

As result, we can find that **CL** can correctly select over 74.6% of effective paragraphs which can be correctively classified by the BERT model, however, the result can only slightly overcome **FM-FNB**, therefore, hypothesis H3 is partially confirmed. Since we cannot ensure that the improvement of **CL** was not resulted by irregularity.

5 Conclusion and Future Work

In this work, we addressed the problem that BERT could not effectively process long documents that exceeded 512 tokens due to its self-attention mechanism. Either we would experience prolonged processing times by the lengthy document, or degraded performance if the content is truncated indiscriminately. To address this problem, we proposed three truncating operations to convert lengthy documents into 512-word paragraphs while preserving the contextual information of the original documents. We first introduced the **FM** method, which truncated the paragraph by identifying the first occurrence of the target label within the document. As

a follow-up, we proposed the **NK** method, which utilized the word2vec to identify the similar vocabularies scattered in the document, then truncated the paragraph accordingly. Finally, we proposed the **CL** method based on a machine-learning approach to identify superior paragraphs that are more effective on document classification tasks. Our dataset was sourced from Kakuyomu, a Japanese self-publishing website for novels, known for its collection of extra-long documents that offer rich context. We then conducted experiments to evaluate the effectiveness of our proposed truncating operations in comparison to the baselines. Our results suggested that our methods can effectively extract relevant context from the original document while adhering to the 512-word limitation. However, our approaches could not surpass the performance of the baseline that used the **FULL** document as input. For future work, we planned to further investigate the effectiveness of our approaches with additional datasets and other language models. We also intend to make modifications to our truncation methods in order to improve performance.

Acknowledgments This work was supported by JSPS KAKENHI Grant Numbers 22H03905 and 21H03775.

References

[1] 総務省統計局. 書籍新刊点数と平均価格, 2022.

[2] 文化庁. 平成 30 年度「国語に関する世論調査」の結果について, 2019.

[3] Fusheng Wei, Han Qin, Shi Ye, and Haozhen Zhao. Empirical study of deep learning for text classification in legal document review. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3317–3320. IEEE, 2018.

[4] David Chen, Hans-Michael Müller, and Paul W Sternberg. Automatic document classification of biological literature. *BMC bioinformatics*, 7(1):1–11, 2006.

[5] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021.

[6] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[7] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[8] Nut Limsopatham. Effectively leveraging bert for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216, 2021.

[9] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*, 2019.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[11] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

[12] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[13] Jun Kong, Jin Wang, and Xuejie Zhang. Hierarchical bert with an adaptive fine-tuning strategy for document classification. *Knowledge-Based Systems*, 238:107872, 2022.

[14] Snehal Ishwar Khandve, Vedangi Kishor Wagh, Apurva Dinesh Wani, Isha Mandar Joshi, and Raviraj Bhuminand Joshi. Hierarchical neural network approaches for long document classification. In *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, pages 115–119, 2022.

[15] Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 7:40707–40718, 2019.

[16] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

[17] Shanshan Yu, Jindian Su, and Da Luo. Improving bert-based text classification with auxiliary sentence and domain knowledge. *IEEE Access*, 7:176600–176612, 2019.

[18] ほりえる. デウス・エクス・マキナを殺せ, 2020.

[19] 田山照巳. 伝説の勇者様 !? ……いえ、補助職の僧侶です, 2020.

[20] Spyridon Samothrakis and Maria Fasli. Emotional sentence annotation helps predict fiction genre. *PloS one*, 10(11):e0141922, 2015.

[21] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128, 1999.

[22] Jahna Otterbacher. Gender, writing and ranking in review forums: a case study of the imdb. *Knowledge and information systems*, 35(3):645–664, 2013.

[23] 電撃文庫. 第 30 回電撃大賞 電撃小説大賞 応募要項, 2023.

[24] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[25] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[26] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Rethinking complex neural network architectures for document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4046–4051, 2019.