IEICE TRANSACTIONS

on Information and Systems

VOL. E106-D NO. 1
JANUARY 2023

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. INF. & SYST., VOL.E106–D, NO.1 JANUARY 2023

PAPER

Auxiliary Loss for BERT-Based Paragraph Segmentation

Binggang ZHUO[†], ***Nonmember***, **Masaki MURATA**[†a)], ***and*** **Qing MA**[††], ***Members***

SUMMARY Paragraph segmentation is a text segmentation task. Iikura et al. achieved excellent results on paragraph segmentation by introducing focal loss to Bidirectional Encoder Representations from Transformers. In this study, we investigated paragraph segmentation on Daily News and Novel datasets. Based on the approach proposed by Iikura et al., we used auxiliary loss to train the model to improve paragraph segmentation performance. Consequently, the average F1-score obtained by the approach of Iikura et al. was 0.6704 on the Daily News dataset, whereas that of our approach was 0.6801. Our approach thus improved the performance by approximately 1%. The performance improvement was also confirmed on the Novel dataset. Furthermore, the results of two-tailed paired *t*-tests indicated that there was a statistical significance between the performance of the two approaches.

key words: *paragraph segmentation, natural language processing, text segmentation, BERT, auxiliary loss*

1. Introduction

Paragraph segmentation can be viewed as a binary classification problem that involves determining whether two consecutive sentences belong to the same paragraph. Automatic paragraph segmentation can not only aid in article writing, but can also make it possible to better understand the content of a poorly formatted text.

Paragraph segmentation is a type of text segmentation task. In the field of text segmentation, paragraph segmentation is less studied than topic segmentation. Due to the similarity between the two tasks, research results obtained in topic segmentation should be considered when studying paragraph segmentation. However, paragraph segmentation is considered more difficult than topic segmentation because it largely depends on personal judgment.

In recent studies on automatic paragraph segmentation, Iikura et al. [1] achieved excellent performance by introducing focal loss to Bidirectional Encoder Representations from Transformers (BERT). Their approach was superior to previous approaches mainly because BERT is one of the best pre-trained models based on the concept of dynamic word representations. In addition to the general usage of BERT, Iikura et al. replaced binary cross entropy (BCE) loss with focal loss to achieve better performance. A detailed discussion of related studies is provided in Sect. 2.

Focal loss [2] is a loss function that can alleviate the problem of class imbalance by penalizing a model's overconfident predictions. BERT [3] is fundamentally a transformer [4] language model with a variable number of encoder layers and self-attention heads. BERT has been demonstrated to achieve high performance in various natural language processing (NLP) tasks in recent years. Due to the nature of neural networks, it is difficult to determine why BERT achieves high performance. However, Clark et al. [5] noted that substantial syntactic information is captured in BERT's attention.

In this study, we investigated paragraph segmentation based on the latest research results reported by Iikura et al. [1]. We observed two problems in the study by Iikura et al. First, although the focal loss they relied on is effective for novel datasets with high class imbalance, it is unknown whether it is equally effective for other datasets. For this reason, we conducted experiments on the 2019 Daily News dataset (hereafter referred to as Daily News), where the class imbalance is more moderate. Second, as an improvement, we found that the performance of the model can be further improved by introducing an auxiliary loss when handling larger window sizes.

We studied paragraph segmentation on two datasets. The first was the Daily News dataset, while the second was the novel dataset used by Iikura et al. (hereafter referred to as Novel), consisting of novels by Natsume Soseki. The Novel dataset exhibited higher class imbalance than the Daily News dataset. To improve the performance of paragraph segmentation, we employed an auxiliary loss to train the model. The loss corresponding to the position where paragraph segmentation must be decided is called the *main loss*, while the losses corresponding to the connection points of the surrounding sentences are called *auxiliary losses*. We describe the auxiliary loss in detail in Sect. 3.3. The advantage of using auxiliary loss is that it encourages the model to focus on a wider range of contextual information, which is beneficial for paragraph segmentation.

We experimentally verified that the auxiliary loss can lead to performance improvement. Furthermore, we confirmed through comparative studies that models with different architectures have different decision-making behaviors.

Manuscript received May 24, 2022.
Manuscript revised September 7, 2022.
Manuscript publicized October 20, 2022.

[†]The authors are with Tottori University, Tottori-shi, 680–8552 Japan.
[††]The author is with Ryukoku University, Otsu-shi, 520–2194 Japan.
a) E-mail: murata@tottori-u.ac.jp

DOI: 10.1587/transinf.2022EDP7083

2. Related Work

2.1 Topic Segmentation

Both paragraph segmentation and topic segmentation are text segmentation tasks. However, topic segmentation has received more attention than paragraph segmentation [6]. Since the two tasks are similar, topic segmentation techniques can be easily applied to paragraph segmentation.

Topic segmenters proposed in various studies can be divided into two major categories: endogenous and exogenous approaches [7]. Endogenous approaches rely on text surface information, whereas exogenous approaches rely on deep semantic information (usually introduced by external models). Machine learning approaches that extract surface features from text are regarded as endogenous approaches.

Endogenous approaches can be traced back to the work of Halliday and Hasan [8], who proposed that text fragments from the same topic have similar vocabulary. Most classic endogenous approaches, such as TextTiling [9] and C99 [10], are based on this concept. TextTiling calculates the cosine similarity between two text blocks to determine the segmentation boundaries, whereas C99 clusters text on the cosine similarity matrix between sentences. Other endogenous approaches include LCseg [11], F06 [12], and TopicTiling [13]. LCseg, which is based on lexical chains and machine learning techniques, outperformed all other tested endogenous approaches [7]. Both F06 and TopicTiling are based on TextTiling. When calculating sentence similarity, F06 uses the dice metric instead of the cosine metric. TopicTiling computes sentence similarity using topic vector representations generated by the Latent Dirichlet Allocation model.

A typical early practice for exogenous approaches was to introduce dense low-dimensional static word representations, such as Latent Semantic Analysis [14], word2vec [15], and GloVe [16]. According to Naili et al. [7], exogenous topic segmenters based on dense low-dimensional static word representations significantly outperformed their endogenous predecessors. Models such as word2vec can obtain the deep semantic relationships between words by training on large domain-independent datasets; this is considered the reason for the performance improvement.

Most research that produced excellent results in the field of topic segmentation a few years ago involved exogenous approaches based on word2vec or GloVe, such as [17]–[20].

Several approaches to learning dynamic contextual word representations have emerged in recent years to address the problem of polysemy, which is intractable with static word representations. These dynamic word representations, extracted from pre-trained models such as those presented in [3], [21]–[23], greatly outperformed their static predecessors in various NLP tasks [24]. In the field of topic segmentation, the introduction of dynamic word representations has also produced results superior to those of previous approaches [25]–[27].

2.2 Paragraph Segmentation

Like topic segmentation approaches, paragraph segmentation approaches can be divided into two categories: endogenous and exogenous. In comparison with topic segmentation, paragraph segmentation has not been extensively studied; to the best of our knowledge, the majority of studies have been on endogenous approaches [28]–[31]. Bolshakov et al. [28] used an approach similar to TextTiling, but with paragraph segmentation based on different text cohesion measure. Genzel et al. [29] used a sparse-voted perceptron model with lexical and syntactic features, which can be considered an early type of neural network model. Machine learning techniques were used in both [30] and [31]; their approaches were based on language model and linguistic features, respectively. The approach of Sporleder et al. [30] was reported to outperform Filippova et al. [31].

This study focuses on exogenous approaches based on the conclusions of Naili et al. [7]. Among exogenous paragraph segmentation studies, the study by Iikura et al. [1] was the first to introduce dynamic word representations into the field of paragraph segmentation. The authors focused on novel datasets and introduced focal loss into BERT to alleviate the impact of class imbalance, achieving excellent results [1].

Our proposed approach is an exogenous approach based on dynamic word representations. To further improve the performance of paragraph segmentation on Daily News dataset, auxiliary losses were introduced based on the approach of Iikura et al. [1].

3. Architectures

The four main architectures for comparison in our experiment are as follows. The first two are baselines, while the last two are our proposed architectures.

- BERT + BCE loss (abbreviated as Vanilla, the baseline approach)
- BERT + focal loss (abbreviated as FL, approach of Iikura et al.)
- BERT + BCE loss + auxiliary loss (abbreviated as AUX, our approach)
- BERT + focal loss + auxiliary loss (abbreviated as FL+AUX, our approach)

The improvement proposed by Iikura et al. was to simply replace BCE loss with focal loss. Our proposed auxiliary loss can be considered a method for combining losses and can be used on any loss. To enable auxiliary loss, it is necessary to modify BERT's general pooling strategy. A pooling strategy is a method for obtaining the embedding representing the input. Common pooling strategies include using the embedding corresponding to the [CLS] token or using the average of the embeddings corresponding to all tokens. We

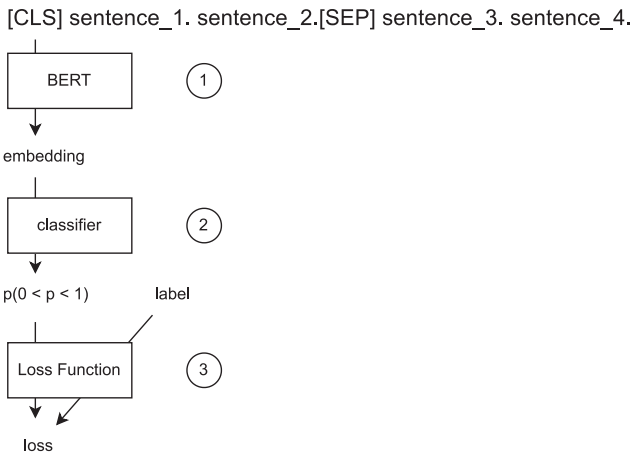

Fig. 1 Classifying text using BERT

describe the different components of these architectures in Sects. 3.1–3.3.

3.1 BERT

BERT is a transformer-based machine learning technique for NLP pre-training developed by Google. It has recently achieved excellent results in various NLP tasks. The approaches used in our experiments are all based on BERT.

A common practice for text classification tasks using BERT is to input the embedding corresponding to the [CLS] token to the multilayer perceptron classifier. Figure 1 illustrates the corresponding steps. It is worth noting that the language of the dataset used in this study was Japanese. BERT was also pre-trained on Japanese datasets. All cases expressed in English in this paper have been translated from Japanese.

3.1.1 Special Tokens of BERT

BERT's tokenizer splits the input text into many tokens. Before inputting these tokens into BERT, it is common to use special tokens to mark special positions in the input text. [CLS] and [SEP] are special tokens in BERT: [CLS] marks the beginning of the input text, while [SEP] marks the point connecting two different parts of the input. Vanilla and FL approaches use only one [SEP] token to represent the point where paragraph segmentation must be performed. In contrast, AUX and FL+AUX use multiple [SEP] tokens to represent the connection points between every two sentences. Therefore, before being input into BERT, multiple [SEP] tokens are added to the token list.

3.2 Base Loss Functions

Next, we introduce the two losses used in our experiment, namely BCE loss and FL loss. It should be noted that the auxiliary loss proposed in this study is strictly a combination of losses; therefore, it can be used based on any type of loss. We evaluate the performance of different combinations in Sect. 4.

3.2.1 BCE Loss

Paragraph segmentation is a binary classification problem whose aim is to determine whether two consecutive sentences belong to the same paragraph. When handling with binary classification problems, the most commonly used loss function is BCE loss. The formula of BCE loss is presented in Eq. (1):

$$BCE(p, y) = -ylog(p) - (1 - y)log(1 - p), \tag{1}$$

where y represents the correct label, 0 represents a non-segmentation point, 1 represents segmentation, and p represents the probability of paragraph segmentation predicted by the model.

The larger the difference between p and y, the larger the loss. When p and y are equal, the loss is 0. Thus, the BCE loss value is between 0 and infinity.

By introducing p_t, as displayed in Eq. (2), we can simplify Eq. (1) to Eq. (3).

$$p_t = \begin{cases} p & if \quad y = 1 \\ 1 - p & if \quad y = 0 \end{cases} \tag{2}$$

$$BCE(p_t) = -log(p_t). \tag{3}$$

In practice, when handling classification problems with class imbalance, the BCE loss often introduces a weight called α_t between 0 and 1. However, we use the BCE loss without α_t to reduce the time for parameter exploration.

3.2.2 Focal Loss

Focal loss was originally used in computer vision. A focal loss function can alleviate the problem of class imbalance by penalizing a model's overconfident predictions. The focal loss is presented in Eq. (4):

$$FL(p_t) = (-\alpha_t)(1 - p_t)^{\gamma} log(p_t). \tag{4}$$

The focal loss is equivalent to BCE loss when γ is equal to 0. p_t is the same as in Eq. (2). We set α_t to 1 for the same reason as explained in Sect. 3.2.1.

As mentioned above, the focal loss penalizes overconfident model output. Specifically, the closer the output is to the label, the more confident the model is; then, the obtained focal loss will be smaller than the BCE loss. γ is an essential hyperparameter of focal loss. To determine the optimal value of γ, we explore four values in [0.5, 1.0, 2.0, 5.0] by grid search.

3.3 Auxiliary Loss

As mentioned earlier, paragraph segmentation can be viewed as a binary classification problem. Since [SEP] tokens are used to mark the connection points of sentences,

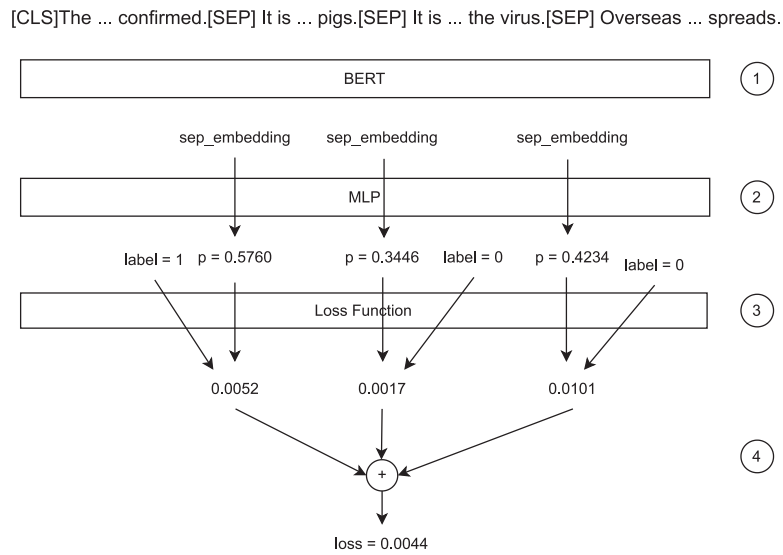

Fig. 2 Steps of BERT + BCE loss + auxiliary loss

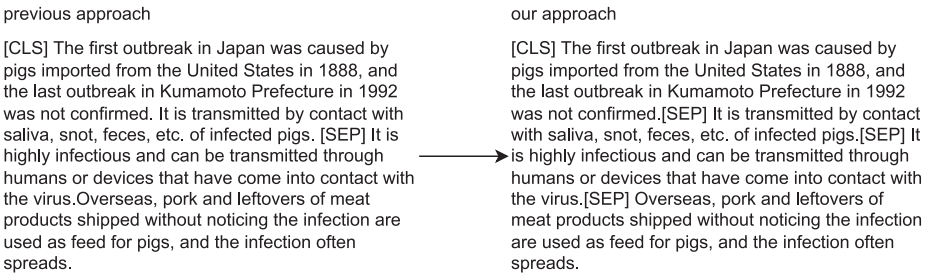

Fig. 3 Obtaining the embedding of sentence connection points

they can also be used to judge whether paragraph segmentation should be performed. For the input of four sentences, we obtain three [SEP] tokens, which we can use to calculate three losses. Except for the main loss in the middle, two auxiliary losses remain. We use the combined loss to tune the model parameters. We refer to this combined loss as the *auxiliary loss*. The advantage of using the auxiliary loss is that it encourages the model to focus on a wider range of contextual information, which is beneficial for paragraph segmentation. This improvement is compatible with the two losses introduced. The steps for obtaining the combined auxiliary loss are as follows:

1. All sentence connection points in the input are marked with [SEP] tokens before being inputted into BERT.
2. All embeddings corresponding to [SEP] positions are selected from the BERT output.
3. These embeddings and the corresponding labels of paragraph segmentation are used to calculate losses separately.
4. The resulting multiple losses are combined into the final loss.
5. The combined loss is used to tune the model parameters.

The above steps are presented in Fig. 2. Steps 1 and 4 are described in Sect. 3.3.1.

3.3.1 Embeddings of Sentence Connection Points

As mentioned in Sect. 3.1.1, BERT's tokenizer splits the input text into many tokens. Before inputting these tokens into BERT, it is common to use special tokens to mark special positions of the input text. Unlike previous approaches, which use [SEP] to mark the sentence connection point in the middle of the window only, we must use multiple [SEP] tokens to mark the connection points of all sentences in the window to obtain the auxiliary losses. The difference in token usage is illustrated in Fig. 3. In actual use, a token is not directly added to the text, but is added to the token list as a token ID. For example, the ID of [CLS] is 2, while the ID of [SEP] is 3.

3.3.2 How to Combine Losses

After the losses corresponding to all sentence connection points are obtained, they must be combined into a final loss. A simple way to achieve this is to sum the losses.

However, the loss corresponding to the sentence connection point at the center of the window is more important because all approaches must use it to ultimately determine

Table 1 Dataset information (Daily News)

Dataset	Articles	Sentences	Segmentation points	Non-segmentation points	Non-seg/seg
Train	2000	27436	8604	18832	2.19
Dev	1000	13508	2073	4566	2.20
Test1	500	6644	2106	4538	2.15
Test2	500	6349	2081	4268	2.05
Test3	500	7193	2180	5013	2.30
Test4	500	6364	2033	4331	2.13
Test5	500	6430	2031	4399	2.17
Test6	500	7130	2222	4908	2.21
Test7	500	7026	2142	4884	2.28
Test8	500	6942	2203	4739	2.15
Test9	500	7806	2327	5479	2.35
Test10	500	7410	2312	5098	2.21

Table 2 Datasets information (Novel dataset)

Dataset	Chapters	Segmentation points	Non-segmentation points	Non-seg/seg
Train	340	3625	24553	6.77
Dev	110	630	4327	6.87
Test	188	1874	9225	4.923

paragraph segmentation. Thus, we call the loss in the middle of the window the *main loss* and the remaining losses *auxiliary losses*. Finally, we use the following equation to calculate the final loss:

$$loss = sum(auxiliary\_losses) * auxiliary\_loss\_rate + main\_loss, \quad (5)$$

where *auxiliary_loss_rate*, the weight of the auxiliary loss, is a hyperparameter that must be re-selected under different experimental settings, such as different datasets or window sizes.

When determining the possible range of the hyperparameter *auxiliary_loss_rate*, a reasonable assumption is that the sum of the weights of the auxiliary losses should be smaller than the main loss. In the case of a window size of 4, there are two auxiliary losses and only one main loss; thus, a reasonable *auxiliary_loss_weight* should be below 0.5.

In a preliminary investigation prior to formal experiment, we determined that the performance is optimal when *auxiliary_loss_weight* is 0.1 or 0.2. In the formal experiment, we explored four values in [0.0, 0.1, 0.2, 0.3] through grid search.

4. Experiments

In this section, we compare the performance of the four previously mentioned architectures, two of which are our proposed architectures. The performance metrics, datasets, and parameter settings for each architecture are described as follows.

4.1 Metrics

Following the trend of many studies on text segmentation, we evaluate our approaches using the F1-score. In practice, the sentence at the beginning of an article is not used for training or evaluation because it is trivial to determine that it is at the beginning of a paragraph.

4.2 Datasets

To ensure the general effectiveness of our approach, we conducted experiments on two datasets. One was the Daily News dataset, while the other was the Novel dataset used by Iikura et al. [1]. Detailed information on the two datasets is presented in Tables 1 and 2, respectively. These tables indicate that the class imbalance is greater in the Novel dataset. For the *t*-test, we divided the test dataset into groups. In the Daily News dataset, we prepared 10 sets of articles, while in the Novel dataset, since the test dataset had 188 chapters, we performed a *t*-test based on chapters.

In the original Daily News dataset, articles were arranged in chronological order; therefore, articles in the same category were not grouped together. When processing the original dataset, we did not change the original order of the articles. After removing articles in special categories, such as "special edition," we removed all articles with indices within a certain range (0 to 1999 for the training dataset) to construct the dataset. We created datasets sequentially without skipping indices.

4.3 Parameter Settings

The deep learning frameworks used in this study were PyTorch and HuggingFace. The pre-trained BERT model was the bert-based-japanese-whole-word-masking model of Tohoku University. The BERT-based models consisted of 12 layers, 768 hidden state dimensions, and 12 attention heads. The optimizer was AdamW, and the learning rate was 2e-5.

We use the term *window size* to describe the number of input sentences expected by the model. In this study, we only considered the case with a window size of 4. The reason is that we could obtain auxiliary losses only when the number of input sentences was greater than or equal to 3. To balance the information at both sides of the focusing point,

Table 3 Best parameters for each approach

Parameter	Vanilla	FL	AUX	FL+AUX
	News Dataset			
γ (focal loss)	-	2.0	-	5.0
auxiliary loss rate	-	-	0.2	0.1
epoch	2	3	2	2
	Novel Dataset			
γ (focal loss)	-	5.0	-	5.0
auxiliary loss rate	-	-	0.3	0.1
epoch	2	2	2	2

we chose a window size of 4. Using a larger window size is a future research topic.

Different model architectures had different hyperparameter settings, such as γ and *auxiliary_loss_rate* for focal and auxiliary losses, respectively. For the sake of fairness, we performed a grid search on the hyperparameters that were important to each architecture. The results are presented in Table 3. During grid search, 10 models were trained for each parameter setting, and their average F1-score on the development (dev) dataset was calculated. Consequently, the parameter setting with the best performance was selected. Considering that the optimal epoch for model training may change after changing the parameter settings, we also explored the optimal epoch between 1 and 3. It is worth noting that the epoch parameters were not independent to save experimental time. In other words, we used the same 10 models for the first, second and third epoch.

5. Results

5.1 Performance

The experimental results for the Daily News dataset are presented in Table 4. To ensure that the performance advantage did not occur by chance, we prepared 10 test datasets. We trained 10 models for each of the architectures described in Sect. 3 based on the best parameter setting obtained using grid search. The last column of Table 4, called "All one," represents the F1-score when all outputs are 1. The standard deviation listed in the table is the performance dispersion of the 10 models. The average F1-score indicates that the performance of the models was as follows: FL + AUX > AUX > Vanilla > FL > All one. The average F1-score obtained using the FL approach of Iikura et al. was 0.6704, whereas the score was 0.6801 after introducing auxiliary loss. To ensure that the performance advantage did not occur by chance, we performed two-tailed paired *t*-tests with F1-scores over 10 datasets.

For the Novel dataset, the test dataset (Natsume Soseki's novel "Light and Dark") had 188 chapters. We calculated the F1-scores corresponding to each chapter separately and then calculated their average. The results are provided in Table 5. The standard deviation listed in the table is the performance dispersion of the 10 models. For the significance test, we performed two-tailed paired *t*-tests with F1-scores for 188 chapters.

We can draw the following conclusions from the experimental results on the two datasets. First, FL+AUX achieved the best performance on both datasets, demonstrating the effectiveness of our proposed approach. To ensure that the performance was statistically significant, we performed two-tailed paired *t*-tests, as described in Sect. 5.2. Second, FL did not perform well on the Daily News dataset, where the class imbalance problem was mitigated, whereas AUX performed well on both datasets.

Table 4 Performance on Daily News dataset

	FL+AUX	FL	Vanilla	AUX	All one
Test1	**0.6784**	0.6715	0.6708	0.6759	0.4814
Test2	**0.7004**	0.6937	0.6923	0.6964	0.4937
Test3	**0.6773**	0.6629	0.6671	0.6713	0.4652
Test4	**0.6846**	0.6745	0.6783	0.6772	0.4842
Test5	**0.6860**	0.6743	0.6787	0.6805	0.4801
Test6	0.6658	0.6565	0.6620	**0.6696**	0.4752
Test7	**0.6841**	0.6750	0.6793	0.6811	0.4673
Test8	**0.6831**	0.6737	0.6760	0.6752	0.4818
Test9	**0.6698**	0.6596	0.6594	0.6644	0.4593
Test10	**0.6718**	0.6626	0.6640	0.6699	0.4756
Mean	**0.6801**	0.6704	0.6728	0.6761	0.4764
Std.	0.0055	0.0116	0.0144	0.0056	0.0

Table 5 Performance on Novel dataset

	FL+AUX	FL	Vanilla	AUX	All one
Mean	**0.8326**	0.8267	0.8172	0.8242	0.2928
Std.	0.0079	0.0114	0.0290	0.0044	0.0

Table 6 p values between each two architectures

	Vanilla	FL	AUX	FL+AUX
	Daily News Dataset			
Vanilla	-	-	-	-
FL	**0.0135**	-	-	-
AUX	**0.0047**	**0.0005**	-	-
FL+AUX	**0.000001**	**0.0000002**	**0.0047**	-
	Novel Dataset			
Vanilla	-	-	-	-
FL	**0.0001**	-	-	-
AUX	**0.0282**	0.4944	-	-
FL+AUX	**4.47e-9**	**0.0038**	**0.0149**	-

5.2 Significance Test

To ensure that the results were statistically significant, we performed a two-tailed paired *t*-test on the results. To obtain paired test scores, we prepared 10 test sets on the Daily News dataset. For the Novel dataset, since the test set contained 188 chapters, we calculated the scores based on chapters.

Table 6 presents the resulting p-values. When the p-value was less than or equal to 0.05, which is marked in bold, we considered the difference between the results obtained using the two architectures to be statistically significant.

6. Discussion

6.1 Decision-Making Behaviors

The experimental results indicate that auxiliary loss can improve model performance; however, the basis of the neural network's decision remains difficult to understand. It is reasonable to assume that models with different architectures behave differently. Analyzing the different behaviors can deepen our understanding of different architectures.

We divided the four architectures mentioned in Sect. 3 into the following three architectures:

- Archt1: architectures using auxiliary loss
- Archt2: architectures using focal loss
- Archt3: vanilla architecture (BERT + BCE loss)

AUX, FL and Vanilla belong to Archt1, Archt2, and Archt3, respectively; however, FL+AUX belongs to both Archt1 and Archt2.

To study the characteristics of different architectures, we first conducted a comparative study on the three architectures except for FL+AUX. Thereafter we evaluated FL+AUX.

6.1.1 Architectures Except for FL+AUX

Cases where only one architecture can answer correctly or incorrectly reflect the characteristics of that architecture. Based on this concept, we divided the cases from the Daily News dataset into six categories listed below and enumerated them in each test dataset:

1. AUX_WIN: The correct answer can only be obtained by AUX.
2. AUX_LOSE: The incorrect answer can only be obtained by AUX.
3. FL_WIN: The correct answer can only be obtained by FL.
4. FL_LOSE: The incorrect answer can only be obtained by FL.
5. VNL_WIN: The correct answer can only be obtained by Vanilla.
6. VNL_LOSE: The incorrect answer can only be obtained by Vanilla.

AUX_WIN signifies that only AUX is correct while FL and Vanilla are incorrect since only AUX, FL, and Vanilla are compared here. The number of cases in each test dataset is provided in Table 7. The probability output of each architecture was averaged across 10 models to mitigate the impact of the randomness of the neural network.

Table 7 indicates that models with auxiliary loss (Archt1) tended to ouput 1 as the number of (label 1, AUX_WIN) and (label 0, AUX_LOSE) were prominent. Since the number of (label 1, AUX_WIN) was greater than that of (label 0, AUX_LOSE), and there were fewer cases with label 1 than cases with label 0, we can conclude that the behavior of outputting 1 was not random. In contrast, models using focal loss (Archt2) tended to answer zero.

6.1.2 AUX+FL

AUX+FL is a hybrid of Archt1 and Archt2. Using the same approach as Sect. 6.1.1, our comparative study demonstrated little difference in decision-making behaviors between AUX+FL and AUX. The results obtained from the comparison of AUX+FL, FL, and Vanilla are also similar to those in Table 7. We conclude that AUX+FL mainly inherited the characteristics from Archt1. This is an expected result because auxiliary loss makes numerous modifications to BERT's input and output, whereas focal loss only changes the loss function.

6.1.3 Cases

Due to the nature of neural networks, it is difficult to determine why an architecture outperforms other architectures; however, for reference we present two cases in this section. Since our interest is mainly in auxiliary loss, we randomly list two cases from the following two categories.

- Case1: The label is 1 and the correct answer can only be obtained by AUX and AUX+FL.
- Case2: The label is 1 and the correct answer can only be obtained by FL and Vanilla.

Case1 is illustrated below, where the label is 1 and the correct answer is only obtained using AUX and AUX+FL. A new line represents a new paragraph. Whether the third sentence is at the beginning of the paragraph determines whether the label is 1 or 0.

> Ichibanya Co., Ltd., which develops "Curry House CoCo Ichibanya," announced on the 15th that it will raise the prices of pork curry and sweet pork curry beginning on March 1. The reason for this is that the prices of ingredients like rice and labor costs are rising.
> At stores in Tokyo's 23 wards, Yokohama, and Kawasaki, the price will be raised by 21 yen from 484 yen to 505 yen. Tokyo, excluding the 23rd ward, and Kanagawa, Saitama, Chiba, and Osaka prefectures, excluding Yokohama and Kawasaki, will remain at 484 yen.

Case2 is presented below, for which the label is 1 and the correct answer can only be obtained by FL and Vanilla.

> Although the boom has passed, IBC donations remain small, with 25,440 yen in FY2003, 280,150 yen in FY2016, 54,100 yen in FY2017, and 10,000 yen in FY2018 until the end of January.
> According to the association, donations are used to support activities to improve patients' quality of life, and as an incentive for research on treatment development, 3 million yen was subsidized to three teams each year for three years from

Table 7 Number of cases in AUX vs. FL vs. Vanilla

Dataset 0	Amount	AUX_WIN	AUX_LOSE	FL_WIN	FL_LOSE	VNL_WIN	VNL_LOSE
label 1	2106	293	20	13	241	23	25
label 0	4538	16	204	102	18	13	43
total	6644	309(4.7%)	224(3.4%)	115(1.7%)	259(3.9%)	36(0.5%)	68(1.0%)
Dataset 1	Amount	AUX_WIN	AUX_LOSE	FL_WIN	FL_LOSE	VNL_WIN	VNL_LOSE
label 1	2081	236	16	15	252	26	34
label 0	4268	33	195	104	13	10	61
total	6349	269(4.2%)	211(3.3%)	119(1.9%)	265(4.2%)	36(0.6%)	95(1.5%)
Dataset 2	Amount	AUX_WIN	AUX_LOSE	FL_WIN	FL_LOSE	VNL_WIN	VNL_LOSE
label 1	2180	339	15	9	271	28	25
label 0	5013	21	253	128	12	17	46
total	7193	360(5.0%)	268(3.7%)	137(1.9%)	283(3.9%)	45(0.6%)	71(1.0%)
...	...	...	...	...	...	...	...
Dataset 9	Amount	AUX_WIN	AUX_LOSE	FL_WIN	FL_LOSE	VNL_WIN	VNL_LOSE
label 1	2312	319	17	3	277	31	30
label 0	5098	25	253	118	7	9	56
total	7410	344(4.6%)	270(3.6%)	121(1.6%)	284(3.8%)	40(0.5%)	86(1.2%)
All datasets (0 to 9)	Amount	AUX_WIN	AUX_LOSE	FL_WIN	FL_LOSE	VNL_WIN	VNL_LOSE
label 1	21637	2981	186	100	2696	311	281
label 0	47657	210	2315	1131	113	134	534
total	69294	3191(4.6%)	2501(3.6%)	1231(1.8%)	2809(4.1%)	445(0.6%)	815(1.2%)

Table 8 Probability output by different methods

	AUX+FL	AUX	FL	Vanilla
Case1	**0.52509**	**0.75856**	0.45897	0.44216
Case2	0.48996	0.42643	**0.51477**	**0.58447**

FY2003.
One of them is a Keio University team that searches for new drug candidates and creates nerve cells from the patient's own iPS cell-induced pluripotent stem cells to reproduce the disease. Shortly, we will start administering candidate drugs to patients.

Table 8 presents the output probability values of various models.

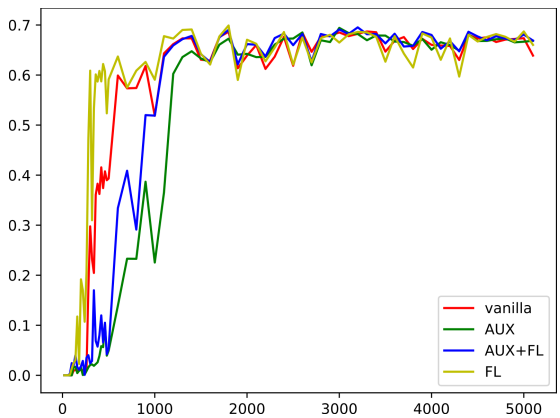

Fig. 4 Learning curve on Daily News dataset

Table 9 F1-scores from learning curve

	Vanilla	FL	AUX	FL+AUX
Mean	0.6616	0.6599	0.6648	**0.6695**
Std.	0.0212	0.0270	0.0159	0.0171

6.2 Learning Curve

Considering that different losses may lead to different convergence patterns of the model, Fig. 4 presents the learning curve on the Daily News dataset. The y-axis represents the average F1-score of the three models on the dev dataset, while the x-axis represents the iteration number. Since the batch size was 16, the first epoch ended at 1,715 iterations, while the second epoch ended at 3,430 iterations.

Figure 4 indicates that all models converged at the end of the first epoch. The performance of each model is difficult to distinguish from the figure, so we calculated the average F1-scores and standard deviation of all points between the second and third epochs; the results are presented in Table 9. This table indicates that the model using auxiliary loss performed better than the other models and was more stable.

6.3 Ablation for Different Pooling Strategies

When using BERT for classification tasks, including paragraph segmentation, a common pooling strategy is to use the representation corresponding to the [CLS] token. To make auxiliary loss possible, one change we made is to use the representation corresponding to [SEP] as the pooling output. To investigate the change in performance caused by different [SEP] configurations and pooling strategies, we conducted a simple ablation experiment. The architectures we wished to compare in this experiment and their performance are presented in Table 10. The special token in bold in the configuration column indicates that it was used as the pooling output.

To reduce the experimental time, the experimental setup was as follows. If possible, the auxiliary loss rate was set to 0 for all architectures in the table, the number of training epochs was set to 2, and five models were trained for each architecture to calculate the average performance. The

Table 10 Ablation setting and scores

Architecture	Configuration	Mean	Std.
AUX_ZERO	[CLS] s1 [SEP] s2 **[SEP]** s3 [SEP] s4	**0.6797**	0.0083
LEFT_SEP	[CLS] s1 [SEP] s2 **[SEP]** s3 s4	0.6773	0.0081
RIGHT_SEP	[CLS] s1 s2 **[SEP]** s3 [SEP] s4	0.6668	0.0076
NO_AUX_SEP	[CLS] s1 s2 **[SEP]** s3 s4	0.6604	0.0174
Vanilla	**[CLS]** s1 s2 [SEP] s3 s4	0.6684	0.0089
COUNTER_SEP	**[CLS]** s1 [SEP] s2 s3 [SEP] s4	0.6789	0.0125

test dataset contained 500 unused articles from the Daily News dataset.

The conclusions drawn from Table 10 are as follows. First, the comparison of NO_AUX_SEP and Vanilla indicates that without auxiliary [SEP], using [CLS] as the pooled output led to higher performance than using [SEP]. Second, auxiliary [SEP] tokens improved the performance of the model; the performance improvement was more significant in the case of using the left-side auxiliary [SEP]. Surprisingly, the setting of COUNTER_SEP led to excellent performance. It is remarkable that the model could infer the learning objective without explicit identifiers.

7. Conclusion

This paper investigates automatic paragraph segmentation on Daily News and Novel datasets. Based on the work of Iikura et al. [1], we further improved the model performance by introducing auxiliary loss. According to the experimental results, the average F1-score obtained using the architecture of Iikura et al. was 0.6704 on the Daily News dataset. Meanwhile, the average F1-score obtained by our proposed architecture was 0.6801, improving the performance by approximately 1%. The performance improvement was also confirmed on the Novel dataset. The results of the two-tailed paired t-test indicated that the difference between the results obtained by the two architectures was statistically significant. For the Daily News dataset, the difference between the architecture using only the auxiliary loss without the focal loss and the architecture of Iikura et al. was also statistically significant.

Auxiliary loss is effective for the following reasons. First, knowing whether there are paragraph segmentation points can help the current judgment. Second, the auxiliary loss can train the model to pay attention to the surrounding paragraph segmentation information. Over focusing on nearby information and ignoring distant information is a disadvantage of neural network models. We can explicitly instruct the model to pay attention to a wider range of information by introducing auxiliary loss.

In addition, the experimental results indicated that the architecture of Iikura et al. achieved poor results on the Daily News dataset. This indicates that focal loss was not suitable for the Daily News dataset because the class imbalance of the this dataset was not as high as that of the Novel dataset.

Our research results can be used in auxiliary writing systems and to organize web texts. The use of auxiliary loss is not limited to paragraph segmentation; it can also be used for other text segmentation tasks. In future work, we will use auxiliary loss to study automatic paragraph segmentation for larger window sizes. It is generally believed that increasing the window size can improve model performance. We will also use the results of this study to develop paragraph segmentation tools. Finally, we will use auxiliary loss for research on other text segmentation tasks.

References

[1] R. Iikura, M. Okada, and N. Mori, "Improving bert with focal loss for paragraph segmentation of novels," Int. Symp. Distributed Computing and Artificial Intelligence, pp.21–30, Springer, 2020.
[2] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Proc. IEEE Int. Conf. Comput. Vis., pp.2980–2988, 2017.
[3] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, vol.30, pp.5998–6008, 2017.
[5] K. Clark, U. Khandelwal, O. Levy, and C.D. Manning, "What does bert look at? an analysis of bert's attention," arXiv preprint arXiv:1906.04341, 2019.
[6] I. Pak and P.L. Teh, "Text segmentation techniques: a critical review," Innovative Computing, Optimization and Its Applications, pp.167–181, 2018.
[7] M. Naili, A.H. Chaibi, and H.H.B. Ghezala, "Comparative study of word embedding methods in topic segmentation," Procedia Computer Science, vol.112, pp.340–349, 2017.
[8] M.A.K. Halliday and R. Hasan, Cohesion in english, Routledge, 2014.
[9] M.A. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic passages," Computational linguistics, vol.23, no.1, pp.33–64, 1997.
[10] F.Y.Y. Choi, "Advances in domain independent linear text segmentation," arXiv preprint cs/0003083, 2000.
[11] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," Proc. 41st Annual Meeting of the Association for Computational Linguistics, vol.1, pp.562–569, July 2003.
[12] O. Ferret, "Improving text segmentation by combining endogenous and exogenous methods," Proc. Int. Conf. RANLP-2009, pp.88–93, 2009.
[13] M. Riedl and C. Biemann, "How text segmentation algorithms gain from topic models," Proc. 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.553–557, June 2012.
[14] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," J. Am. Soc. Inf. Sci., vol.41, no.6, pp.391–407, 1990.
[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
[16] J. Pennington, R. Socher, and C.D. Manning, "Glove: Global vectors for word representation," Proc. 2014 Conf. empirical methods in natural language processing (EMNLP), pp.1532–1543, 2014.
[17] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, "Text segmentation as a supervised learning task," arXiv preprint arXiv:1803.09337, 2018.
[18] P. Badjatiya, L.J. Kurisinkel, M. Gupta, and V. Varma, "Attention-based neural text segmentation," European Conf. Information Re-

trieval, pp.180–193, Springer, 2018.
[19] J. Li, A. Sun, and S.R. Joty, "Segbot: A generic neural text segmentation model with pointer network," IJCAI, pp.4166–4172, 2018.
[20] Y. Wang, S. Li, and J. Yang, "Toward fast and accurate neural discourse segmentation," arXiv preprint arXiv:1808.09147, 2018.
[21] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," Advances in Neural Information Processing Systems, vol.30, pp.6294–6305, 2017.
[22] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations. arxiv 2018," arXiv preprint arXiv:1802.05365, vol.12, 1802.
[23] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI Technical Report, pp.1–12, 2018.
[24] Y. Wang, Y. Hou, W. Che, and T. Liu, "From static to dynamic word representations: a survey," Int. J. Mach. Learn. Cybern., vol.11, no.7, pp.1611–1630, 2020.
[25] M. Lukasik, B. Dadachev, G. Simões, and K. Papineni, "Text segmentation by cross segment attention," arXiv preprint arXiv:2004.14535, 2020.
[26] A. Solbiati, K. Heffernan, G. Damaskinos, S. Poddar, S. Modi, and J. Cali, "Unsupervised topic segmentation of meetings with bert embeddings," arXiv preprint arXiv:2106.12978, 2021.
[27] L. Xing, B. Hackinen, G. Carenini, and F. Trebbi, "Improving context modeling in neural topic segmentation," arXiv preprint arXiv:2010.03138, 2020.
[28] I.A. Bolshakov and A. Gelbukh, "Text segmentation into paragraphs based on local text cohesion," Int. Conf. Text, Speech and Dialogue, pp.158–166, Springer, 2001.
[29] D. Genzel, "A paragraph boundary detection system," Int. Conf. Intelligent Text Processing and Computational Linguistics, pp.816–826, Springer, 2005.
[30] C. Sporleder and M. Lapata, "Broad coverage paragraph segmentation across languages and domains," ACM Trans. Speech and Language Processing (TSLP), vol.3, no.2, pp.1–35, July 2006.
[31] K. Filippova and M. Strube, "Using linguistically motivated features for paragraph boundary identification," Proc. 2006 Conf. Empirical Methods in Natural Language Processing, pp.267–274, July 2006.

Binggang Zhuo received his B.S. degree in engineering from the University of Electronic Science and Technology of China, Zhongshan Institute, in 2019. He joined the Natural Language Laboratory of Tottori University as a research student in 2020 and began the master's course in 2021.

Masaki Murata received his B.S., M.S., and Ph.D. degrees in engineering from Kyoto University in 1993, 1995, and 1997, respectively. He worked in the Communications Research Laboratory (currently, the National Institute of Information and Communications Technology [NICT]), Japan, from 1998 to 2010. In 2010, he moved to Tottori University, Japan, where he worked as a professor in the Department of Information and Electronics, Graduate School of Engineering. He is also with the Cross-informatics Research Center, Tottori University. His research interests include natural language processing, machine translation, and information retrieval.

Qing Ma received his B.S. degree in engineering from Beihang University, China, in 1983, and his M.S. and DEng degrees in computer science from the University of Tsukuba, Japan, in 1987 and 1990, respectively. He worked in Ono Sokki Co., Ltd., Japan, from 1990 to 1993 and in the Communications Research Laboratory (currently, the National Institute of Information and Communications Technology [NICT]), Japan, from 1993 to 2003. In 2003, he moved to Ryukoku University, Japan, as a professor in the Applied Mathematics and Informatics Course, Faculty of Advanced Science and Technology. His research interests include machine learning and natural language processing.