This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization

Ming-Hsiang Su, Chung-Hsien Wu, *Senior Member IEEE*, and Hao-Tse Cheng

***Abstract*—This study proposes a two-stage method for variable-length abstractive summarization. This is an improvement over previous models, in that the proposed approach can simultaneously achieve fluent and variable-length abstractive summarization. The proposed abstractive summarization model consists of a text segmentation module and a two-stage Transformer-based summarization module. First, the text segmentation module utilizes a pre-trained Bidirectional Encoder Representations from Transformers (BERT) and a bidirectional long short-term memory (LSTM) to divide the input text into segments. An extractive model based on the BERT-based summarization model (BERTSUM) is then constructed to extract the most important sentence from each segment. For training the two-stage summarization model, first, the extracted sentences are used to train the document summarization module in the second stage. Next, the segments are used to train the segment summarization module in the first stage by simultaneously considering the outputs of the segment summarization module and the pre-trained second-stage document summarization module. The parameters of the segment summarization module are updated by considering the loss scores of the document summarization module as well as the segment summarization module. Finally, collaborative training is applied to alternately train the segment summarization module and the document summarization module until convergence. For testing, the outputs of the segment summarization module are concatenated to provide the variable-length abstractive summarization result. For evaluation, the BERT-biLSTM-based text segmentation model is evaluated using ChWiki_181k database and obtains a good effect in capturing the relationship between sentences. Finally, the proposed variable-length abstractive summarization system achieved a maximum of 70.0% accuracy in human subjective evaluation on the LCSTS dataset.**

***Index Terms*— Abstractive summarization, text segmentation, variable-length summarization, BERT, bidirectional LSTM.**

I. Introduction

In recent years, a large number of text resources have appeared on various websites, such as social network, news websites, content farm websites and Wikipedia. Because of the rapid growth of information, how to efficiently process and utilize these text resources have become an increasingly crucial challenge to address. While the human reading rate is limited to 300 words per minute [1], people in modern times have to face the problem of how to quickly understand these large amounts of information in a limited time. Unlike the quality of articles that are strictly controlled in Wikipedia, content farm websites quickly grow with low-quality articles for high click-through rates. For example, Demand Media, a content farm website, published one million articles per month in 2009, four times as many as English Wikipedia [2]. Content farms have even caused the proliferation of fake news [3]. How to quickly filter out the parts in articles without informative contents has become a problem that cannot be delayed in modern times. It is a time-consuming and impractical thing to read the article in its entirety to see if it is what you need. If people can quickly understand the content and focus of the article, they can quickly get what they need. The main goal of an automatic summarization system is to produce shorter content than the original text but retain important information. Therefore, an automatic summarization system can help people understand the important content of the article, so that people can read the abstract directly to get the information needed, saving a lot of time.

In the real world, humans will consider the length of the input text and the application environment when writing the summary, and then determine the length of the summary required. It is obviously improper to generate summaries with the same length for the documents which contain quite different quantity of information. If the summarization system can generate summaries of different content lengths based on the quantity of information and different summary requirements from the users, it can save more manpower and time spent in practical applications. Therefore, a variable-length summarization system is required to generate summaries according to the user's requirements. For example, by applying variable-length summarization and specifying the length of the summary to be 3 sentences, a variable-length summary of the article in TABLE I can be obtained. Compared with the headline summary, the variable-length summary successfully extracts more information in the input article.

Automatic summarization methods can be roughly divided into two categories [4], the extractive methods [5-11] and the abstractive methods [12-18]. The former extracts important parts from the input text, while the latter makes a proper understanding of the input text and applies the generative model to generate the summary. In comparison, abstractive summaries

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract No. MOST 108-2221-E-006-103-MY3.

M.-H. Su is with the Department of Computer Science and Information Engineering, National Cheng Kung University (e-mail: huntfox.su@gmail.com).

C.-H. Wu is with the Department of Computer Science and Information Engineering, National Cheng Kung University (e-mail: chunghsienwu@gmail.com).

H.-T. Cheng is with the Department of Computer Science and Information Engineering, National Cheng Kung University (e-mail: top30339@gmail.com).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

are more consistent with human summaries [19] and are more fluent. In the past, most summarization systems used the extractive method because the abstractive method involved more difficult techniques, such as semantic representation and natural language generation. But the advances in summarization technology in recent years have led to more and more research on abstractive summarization, aiming to produce more human-like and fluent summaries [12-18]. Most of abstractive summarization systems generate summaries through a data-driven end-to-end model, which makes it impossible to dynamically adjust the length of the summary based on the length of the input text or special demands. Besides, the existing abstractive databases [14, 20] are mostly from the news or social media. The reference summaries of these corpora are mostly single-sentence titles, so the models trained using these corpora only produce a single sentence as the summary.

Although abstractive summaries are more fluent than extractive summaries, most of the existing abstractive systems cannot obtain variable-length summaries. Instead, the extractive system can change the length of the summary according to the status by specifying the number of paragraphs to be extracted. The segments can firstly be extracted by using a text segmentation model. Then, a variable-length summary can be maintained by doing abstractive summarization for each segment, while the summary can be kept fluent. The main motivation of this study is to combine two summarization methods and a text segmentation model to generate a variable-length and fluent summary.

TABLE I
SUMMARY EXAMPLE

Input Article	中山市教育系統單位今年公開招聘 143 名教職員。被聘用者爲事業單位人員編制，享受國家政策規定的薪酬待遇。欲應聘者，可在本月 28 日 17:30 前登錄中山教育信息港進入招聘系統網絡報名。(The Zhongshan Education System agencies publicly recruited 143 faculty members this year. The hiring is prepared for the personnel of the institution and enjoys the remuneration package stipulated by the state policy. If you want to apply, you can log in to the Education Information Network in Zhongshan Education Information Port before 17:30 on the 28th of this month.)
Headline Summary	中山招 143 名事業編制教職員。(Zhongshan recruits 143 career staff.)
Variable-Length Summary (Specified length is 3)	中山招 143 名事業編制教職員，被聘用者享用國家政策規定的薪資，本月 28 日 17:30 前可登入報名。(Zhongshan recruits 143 business staff, and the hired person enjoys the salary set by the state policy. You can log in before 17:30 on the 28th of this month.)

As the existing abstractive summarization systems are trained in a data-driven manner, these systems can only generate fixed-length summaries. They cannot dynamically generate variable-length summaries based on the user's demands or other conditions. However, if the text segmentation task is invoked before summarization, the input text can be dynamically segmented to obtain a specified number of segments. Furthermore, the summary of each segment can be combined to obtain a variable-length summary. As the existing abstractive summarization database contains only the headline summaries, the database with variable-length summaries that can be used directly to train the abstractive summarization model is still not available. In addition, it is very difficult and laborious to construct such a database, because the specified length of the summary varies greatly in the face of different situations or different input articles. It is almost impossible to consider all the circumstances for building a variable-length summary database.

Based on the above discussions, the existing problems can be summarized as follows. First, most of the existing abstractive summarization models are designed to generate fixed-length summaries rather than variable-length summaries. Second, the available summarization databases only contain headline summaries, which cannot be directly applied to train a variable-length summarization system. Accordingly, to solve the above problems, this study proposes a two-stage Transformer-based [21-22] approach to construct a variable-length summarization system using the database with only headline summaries.

Even though the proposed method can be used to train the variable-length abstractive summarization model using only the existing summarization database, the model still faces the model convergence problems. The reason is that as there is no segment-based summary, the segment-based summarization model in the two-stage model cannot be trained by the teacher forcing technique, which makes the result of the first stage segment summarization model unacceptable. Besides, in the training of the two-stage model, the error of the first stage will affect the result of the second stage. Therefore, if only the loss value of the headline summary is used in the second stage for training the two-stage model, it is difficult to converge to an acceptable model.

In this study, the proposed system combines a text segmentation model and a two-stage abstractive summarization model. The text segmentation model is used to divide the input text into a specified number of segments, and each segment uses the abstractive summarization model to obtain a sentence-based summary. The concatenation of the sentence summaries forms the variable-length abstractive summary. For text segmentation, there is no Chinese database available for text segmentation model training. This study introduces a large Chinese text segmentation database ChWiki_181k and applies the advanced language representation model, the Bidirectional Encoder Representations from Transformer (BERT), to the text segmentation task.

The two-stage model, proposed in this study, contains two sequence-to-sequence models (e.g., Transformer). The general training method of the sequence-to-sequence model needs the target output to help train the decoder using teacher forcing, which makes the model output more stable and easier to train. However, the two-stage model only has the headline summary as the target output, so teacher forcing can only be performed in the second stage, which makes the first stage generation output very unstable and difficult to converge. Therefore, this study adopts the extractive model to generate the segment-based extractive summary as the target output in order to train the first stage using teacher forcing. In the two-stage model, the first stage produces an extractive summary of each segment and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

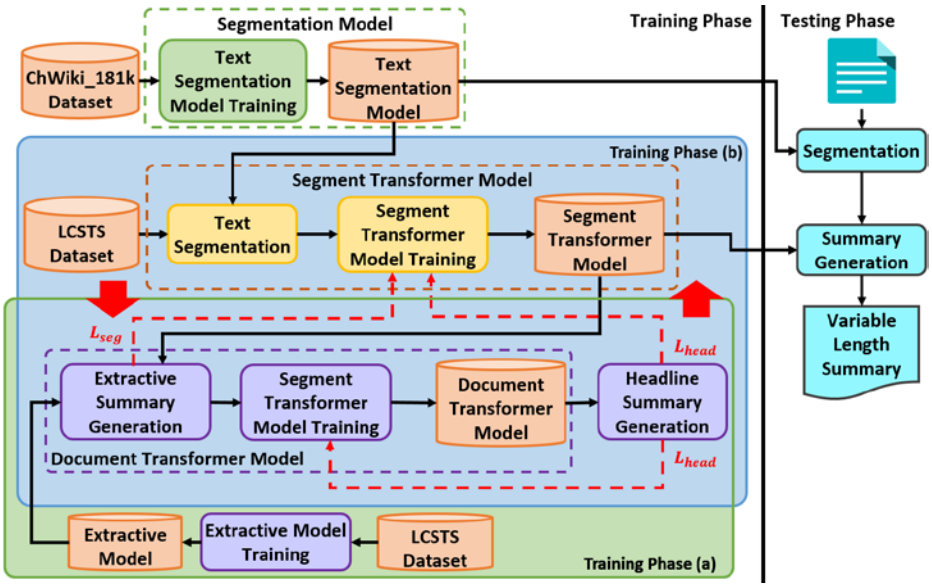

Fig. 1. Framework of the variable-length abstractive summarization system.

calculates the loss value of the segment-based extractive summary. The second stage uses the summary of each segment to generate a headline summary that can be used to compare with the reference summary in the database to calculate the loss value. The two loss values are combined and used to train the entire two-stage model. In the test process, a variable-length abstractive summary can be generated by specifying the number of divided segments in the first stage.

The main contribution of this study is to build a summarization system that can generate variable-length abstractive summaries according to the user's demand. This is an improvement over previous models, in that the proposed approach can simultaneously achieve fluent and variable-length abstractive summarization. The variable-length abstractive summarization model is divided into a text segmentation module and a two-stage Transformer-based summarization module. The proposed text segmentation module, which utilizes BERT and Bidirectional LSTM, shows improved performance over existing methods. The two-stage Transformer-based summarization module combines extractive and abstractive methods to produce fluent and variable-length abstractive summaries.

II. System Framework

The main goal of this study is to build a variable-length abstractive summarization system that can generate variable-length abstractive summaries according to the user's demands. The system framework is shown in Fig. 1, which is divided into the training phase and the testing phase. Four models are constructed during the training phase, namely the text segmentation model, the extractive model, the document transformer model and the segment transformer model. The text segmentation model divides the input text into *m* segments; the value of *m* is determined according to the user's setting. The extractive model is an extractive summarization model, acting as the function of finding the key sentence of the input text. The segment transformer model and the document transformer model form the abstractive summarization model in which the former generates a sentence-based summary from each segment, and the latter generates a headline summary of the entire text input. The testing phase includes a text segmentation model and a segment transformer model. Text segmentation divides the input text into a specified number of segments, and the segment transformer model is used to generate a variable-length summary. In Fig. 1, the extractive model and the document transformer model are used in the training phase only as the pre-trained models to help train the segment transformer model, and they are not activated during the testing phase. The testing phase uses only the text segmentation model and the segment transformer model to obtain the summarization results.

A. Text Segmentation Model

The goal of text segmentation is to divide text into segments, such that each segment is topically coherent, and segmentation points indicate a change of topic. The number of segments can be specified by the user or can be set according to the desired ratio of the input length. Before text segmentation, the input text goes through a sentence tokenizer, which divides the input text into sentences of appropriate length. The text segmentation model then finds the appropriate segmentation points in these sentences. In the English system, the Natural Language Tool Kit (NLTK) [23] is mostly used to tokenize sentences. However, there is no currently available tool for tokenizing Chinese sentences.

Therefore, this study uses a simple way to perform sentence segmentation of Chinese text. First, this method uses punctuations marks as delimiters, including commas, periods, semicolons, exclamation marks, and question marks, whereby each input text can be divided into n sentences $(sent_1, sent_2, \dots, sent_n)$. Then, the existing text segmentation model encodes each sentence to obtain a sentence representation and determines whether each sentence is the starting point or end point of the segment by using a classifier [24-25]. This study proposes another different viewpoint in the representation of segmentation points. For an input text with n

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

sentences $(sent_1, sent_2, \ldots, sent_n)$, the representation vector of the i-th segmentation point defined as the boundary between the i-th and the $(i+1)$-th sentences is calculated, as shown in (1).

$$r_i = BERT(sent_i, sent_{i+1}) \tag{1}$$

where $1 \le i \le n-1$ and *BERT* is the BERT-based segmentation point embedding model.

Recent language representation models, including BERT [26], use pre-trained methods to enhance the performance of subsequent NLP tasks. However, unlike other models, BERT applies a masked language model (MLM) to pre-train the language representation model to achieve the goal of a bidirectional language model. MLM aims to solve a serious problem in the bidirectional language model. In the bidirectional language model, each word can indirectly see itself in all layers. Therefore, the bidirectional language model can simply copy the input to the output during the training process, which makes the model unable to learn the correct representation vector to describe the contextual information. In the comparison of a bidirectional and unidirectional model, the former can capture more complete contextual information and provide better representation vectors for subsequent models. Many NLP tasks need to capture relationships between sentences [22, 27-30], such as question answering (QA) systems, natural language inference (NLI) systems, and text segmentation tasks. BERT applies next sentence prediction (NSP) technique to pre-train the language representation model, which is to predict whether sentence B is the next sentence of sentence A. BERT pairs all the sentences in the training set and randomly select 50% sentence B as the next sentence of sentence A, while the other 50% is not. From MLM and NSP, it can be noticed that BERT is a language representation model that suits both sentence-level and token-level NLP tasks.

This study proposes the BERT-biLSTM model which is a text segmentation model combining BERT and the bidirectional LSTM. In BERT-biLSTM model, BERT produces the representation sequence (r_1, r_2, …, r_{n-1}), and then the representation is discriminated by a classification model to determine if the segmentation point is the boundary of a segment. BERT-biLSTM model adopts the bidirectional LSTM as the classification model, which takes the representation sequence as input and outputs the probability sequence (p_1, p_2, …, p_{n-1}). Equations (2) and (3) calculate the hidden outputs of the forward and backward layers in the biLSTM, respectively. Then the forward output and the backward output are concatenated, as shown in (4). Finally, the BERT-biLSTM model outputs the probability sequence ($p_1, p_2, \ldots, p_{n-1}$) through a sigmoid function, as shown in (5).

$$\vec{h}_i = LSTM(r_i, \vec{h}_{i-1}, \vec{c}_{i-1}) \tag{2}$$

$$\overleftarrow{h}_i = LSTM(r_i, \vec{h}_{i+1}, \overleftarrow{c}_{i+1}) \tag{3}$$

where $\overrightarrow{h_i}$ is the forward output, $\overrightarrow{c_{i-1}}$ is the forward cell state, $\overleftarrow{h}$ is the backward output, and $\overleftarrow{c_{i+1}}$ is the backward cell state.

$$h_i = \vec{h}_i \oplus \overleftarrow{h}_i \tag{4}$$

$$p_i = \sigma(h_i) \tag{5}$$

where $0 \le p_i \le 1$ and p_i indicates the probability that the i-th segmentation point is a boundary of the segment. The loss value is then calculated by the binary cross-entropy function to train the BERT-biLSTM model, as shown in (6).

$$J(\theta) = \sum_{i=1}^{n-1} [-y_i \log p_i - (1-y_i)\log(1-p_i)] \tag{6}$$

where y_i is the target label, which is binary and takes only values 0 or 1.

B. Extractive Model

The extractive model takes all sentences of a segment as the input and outputs the corresponding probability sequence indicating the probability of each sentence for extraction. In this study, the BERT-based summarization model (BERTSUM) [11] is adopted to construct the extractive model. BERTSUM consists of two input format adaptation methods. The first technique is to encode multiple sentences. This technique uses the [CLS] token and the [SEP] token to wrap all the input sentences. BERTSUM can generate representations for all sentences by adding [CLS] token before all sentences, and these representations consider the contextual information of the entire input segment. The second technique is the interval segment embedding. BERTSUM employs E_A or E_B according to whether the sentence position is odd or even. The method successfully uses only two values of segment embedding to encode segments in multiple sentences. Using the above two techniques, BERTSUM can obtain the representation sequence (T_1, T_2, …, T_n) for the article with n sentences ($sent_1$, $sent_2$, …, $sent_n$) by BERT. With the representation sequence through BERTSUM, a classifier is constructed to obtain a probability sequence (p_1, p_2, …, p_n), where $0 \le p_i \le 1$, and p_i indicates the probability for the extraction of the i-th sentence. This study uses a fully connected layer classifier, because the previous research has shown that a fully connected layer in the extractive model performed almost the same as other complex classifiers [11]. For training the fully connected classifier, the input is the representation sequence (T_1, T_2, …, T_n) of the article with n sentences and the target output sequence is ($y_1, y_2, \ldots, y_n$), where y_i is 0 or 1 indicating whether the i-th sentence is the sentence to be extracted from the segment.

C. Document Summarization Model

Nowadays, most sequence transformation systems are based on the sequence-to-sequence model with an encoder-decoder architecture. In previous studies, encoders and decoders were usually built using RNNs or CNNs [31]. The emergence of Transformer considering the attention mechanism becomes a popular alternative to RNNs or CNNs. Moreover, because of the architecture of Transformer, the training process can be more parallelized and the training time can be reduced. Transformer can be applied to any sequence-to-sequence task, in which the encoder converts the input sequence into a

representation sequence, and the decoder further generates the output sequence based on the representation sequence. Transformer is composed of an *N*-layer encoder and an *N*-layer decoder, just like the traditional sequence-to-sequence model. Each encoder consists of two sublayers, which are a multi-head self-attention layer and a fully connected feed-forward layer. Each decoder also includes a multi-head self-attention layer and a fully connected feed-forward layer, but it additionally incorporates a masked multi-head self-attention mechanism. The purpose of masking is to make the model refer to the information only before the *i*-th position when predicting the *i*-th position. Besides, the residual connection [32] and the layer normalization [33] are attached in all sub-layers of the encoders and decoders.

An attention function is defined as mapping a query and a set of key-value pairs to an output [21]. The query, keys, values and output are all vectors. The output is a weighted sum of the values, where the weight is calculated by a compatibility function of the query with the corresponding key [21]. There are two basic attention mechanisms, namely additive attention and dot-product attention (multiplicative attention). Because the query, key, value, and output are all vectors, the dot-product attention can apply the optimized multiplication of vectors to make the operation faster and save memory space in practice. However, past research has shown that additive attention is better than dot-product attention [34]. In order to solve this problem while maintaining the advantages of dot-product attention, Transformer proposes the scaled dot-product attention, as shown in (7). In practice, the collection of the query, key, and value will be packed into Q, K and V. The traditional dot-product attention is that the result of multiplying Q and K and passing the softmax function is directly considered as the weight of V, while the scaled dot-product attention divides the weight before the softmax function by $\sqrt{d_k}$, thereby solving the problem that the value is too large before the softmax function.

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

In addition, Transformer finds that after Q, K and V are linearly projected h times, the result of the attention mechanism is better. Thus, the multi-head attention is proposed, as shown in (8). The projection result of each head is passed through the scaled dot-product attention, and the results of the respective heads are connected to obtain the output.

$$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \tag{8}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{9}$$

where the trainable parameters include $W_i^Q \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^K \in \mathbb{R}^{d_{model}\times d_k}$, $W_i^V \in \mathbb{R}^{d_{model}\times d_v}$, and $W^O \in \mathbb{R}^{hd_v\times d_{model}}$. W_i^Q, W_i^K, and W_i^V indicate the linear projection of the input to each head, and W^O indicates the linear projection of the concatenation of each head to the original dimension. Then all the encoders and decoders in the Transformer pass through a fully connected feed-forward layer, which contains two linear transformations and a rectified linear unit (ReLU), as shown in (10).

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{10}$$

where x is the input of a fully connected feed-forward layer, and W_i and b_i are the parameters of the i-th linear transformation.

The document summarization model outputs the probability $p_{c,j}$ which indicates the probability of generating the j-th word at the c-th step. The loss value is determined by the cross-entropy loss function as shown in (11) and used to train the document Transformer model.

$$J(\theta) = \sum_{c=1}^{C}\sum_{j=1}^{N} -y_{c,j}\log p_{c,j} \tag{11}$$

where $y_{c,j}$ is the target label of the j-th word at the c-th step with binary values 0 or 1, N is the number of the words, and C is the number of steps.

D. Segment Summarization Model

Although a Transformer-based abstractive summarization model can be established, it can only generate headline summaries of a single sentence rather than variable-length summaries using the currently available database. To this end, this study proposes a two-stage model. The first stage is the segment summarization model, which combines the text segmentation model, the extractive model and the segment transformer model to achieve the goal of generating variable-length summaries. Moreover, the document transformer model is employed in the second stage to increase the abstraction of the summary and solve the problem of the lack of corpus with segment summaries. The process of the two-stage model is shown in Fig. 2.

In the two-stage model, the first stage accepts the segment sequence $(seg_1, seg_2, \dots, seg_m)$ of length m as the input and outputs m sentence as the segment summaries $(sum_1, sum_2, \dots, sum_m)$, where $sum_i \in \mathbb{R}^{length_i\times d_{model}}$ is a one-sentence summary for the i-th segment seg_i, and d_{model} is the dimension of the model. Then all the representation vectors are concatenated to obtain the intermediate summary, sum_{inter}, as the input feature embedding of the document transformer model, as shown in Fig. 2. The segment transformer model should be trained with teacher forcing technique. Otherwise, it will affect the convergence speed in the training process and even the quality of the generated summary. However, the existing database does not have segment summaries, so there is no target output in the segment transformer model for teacher forcing-based training. The second problem is that only the loss value generated by the predicted headline summary of the document transformer model could be used to update the parameters of the whole two-stage model. This causes the result to be very divergent and makes it difficult to converge for the training of the whole model.

In order to deal with the above problems, the proposed system uses the extractive model BERTSUM to extract the most informative sentence for each segment as the segment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

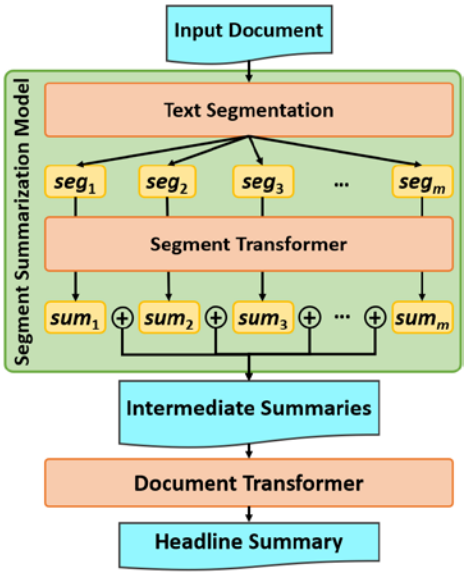

Fig. 2. The framework of the two-stage summarization model.

extractive summary. The segment extractive summary, serving as the target output, is used to train the segment transformer model with teacher forcing to solve the above problem. Besides, the two-stage model can determine the loss value after the first stage by considering the segment extractive summary to help train the segment transformer model, making it easier to converge and solve the second problem.

E. Training of The Two-Stage Transformer-based Model

The training and testing block diagram of the proposed variable length summarization system is shown in Fig. 3. In the training phase (a), we first train a BERTSUM-based [11] extractive model to extract the important sentences from the segment using a sub-database selected from the LCSTS corpus. The process for constructing the sub-database applies a greedy algorithm to find the set of sentences with the highest ROUGE score [35]. Then, we use collaborative training to train the two-stage Transformer-based model. The training process begins with the document Transformer in the second stage using the LCSTS corpus. The input of the document Transformer is the outputs of the extractive model and the output of the document Transformer is the headline summary of LCSTS corpus, as shown in training phase (a) of Fig. 3. The trained document Transformer is regarded as the original document summarization module for further training. The segment Transformer in the first stage is then trained by simultaneously considering the outputs of the segment Transformer and the trained document Transformer by using the segments of LCSTS corpus, as shown in training phase (b) of Fig. 3. The input of the segment Transformer is the segments of the LCSTS corpus obtained from the segmentation model, and the output of the segment Transformer is the variable-length summaries. The target output of the segment Transformer for the first-stage training is the sentence output of the extractive model. The variable length summaries then are fed to the document Transformer. The proposed two-stage model uses two loss values to train the two Transformer model. The first is the loss value determined from the summarization output of the segment Transformer model, and the second is the loss value estimated from the headline summary generated by the document Transformer model. Finally, the training process takes turns to train the segment Transformer model and the document Transformer model until convergence, as shown in training phase (b) and (c) of Fig. 3. In the testing phase, by specifying the length of the desired abstractive summary, the input text is divided into the desired number of segments. Given the divided segments, the segment Transformer generates a sentence-based abstractive summary for each segment. Finally, the variable-length abstractive summary is obtained by concatenating the sentence-based summaries, as shown in the testing phase of Fig. 3.

When training the two-stage model, only the parameters of the segment Transformer model in the first stage will be updated by using L_{seg} and L_{head}, as shown in Fig. 3. The document transformer model in the second stage helps calculate the headline summary loss value and does not update its parameters to avoid the destruction of the stability of the pre-trained model. Then the parameters of the document transformer model in the second stage will be updated by using only L_{head}. The training process is shown in (b) and (c) of Fig. 3. The training process will take turns to train the segment Transformer model and the document Transformer model until convergence. The required loss values, namely L_{seg} and L_{head}, are calculated using (12). The total loss value L_{total} is calculated by

$$L_{total} = \alpha \times L_{seg} + (1-\alpha) \times L_{head} \tag{12}$$

where $0 \le \alpha \le 1$, and α represents the proportion of L_{seg} in L_{total}.

The idea about the setting of the hyperparameter α in (12) is to decide the contribution of the document Transformer model and the segment Transformer model. If the segment Transformer model contributes more, the loss generated by the segment Transformer model will be considered more for the overall training. Otherwise, the loss generated by the document transformer model will be a lower proportion for the overall training. Finally, L_{total} is backpropagated to the segment Transformer model to complete the two-stage model training. In the testing phase of Fig. 3, after the input text is passed through the text segmentation model to obtain a specified number of segments, all segments pass through the trained segment Transformer model to obtain the specified number of summaries, thereby achieving the goal of "variable-length abstractive summarization". The variable-length abstractive summary is regarded as the final output of the summarization model.

III. Experimental Results and Discussion

This study combines the text segmentation model BERT-biLSTM, the extractive model BERTSUM and two Transformer models to construct the variable-length abstractive summarization system. During the training phase, the above four models sequentially complete the training process, and the experiments in this section will verify whether each model can achieve the expected goals and confirm its effect.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

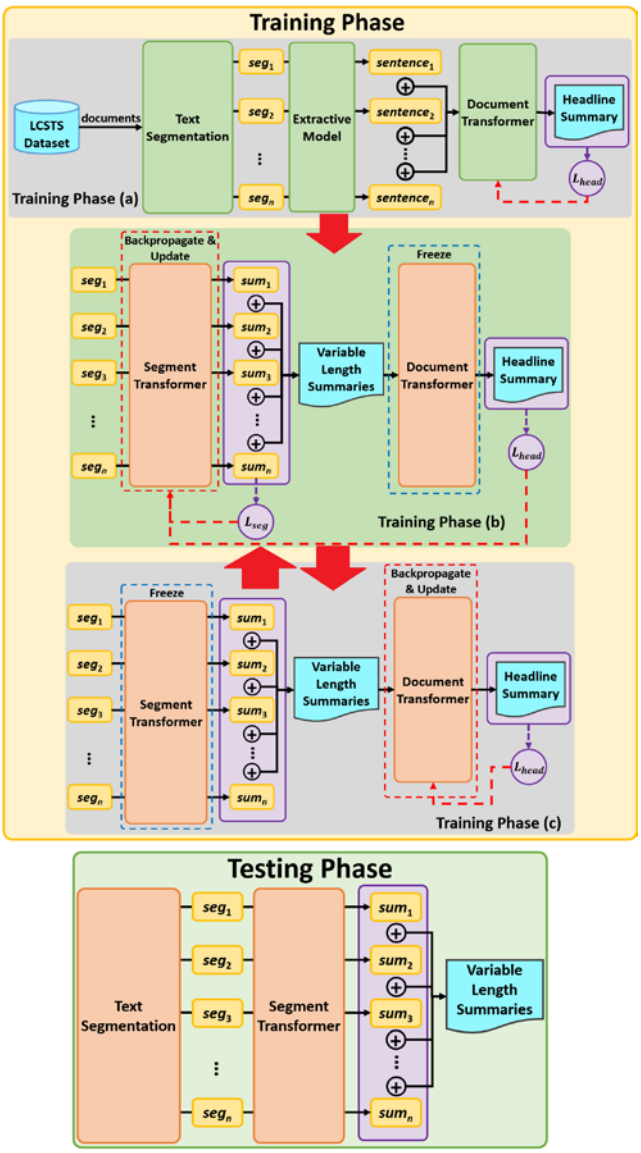

Fig. 3. The training and testing processes of the proposed variable length summarization system.

A. Evaluation Metrics

The sub-tasks of the proposed variable-length abstractive summarization system can be divided into two categories, namely the text segmentation task and the summarization task. For the two tasks, different metrics were used to evaluate the effect of the proposed model. P_k indicator [36] was employed for evaluating the text segmentation model, while ROUGE [35] and subjective evaluation [17] were applied for summarization model evaluation.

Consider the error metric P_k [36] that is the error probability evaluating that two sentences in the text are divided into the same segment or different segments using the segmentation model. More formally, given reference segment (*ref*) and hypothesized segment (*hyp*) obtained by the segmentation model, the value of P_k is defined in (13).

$$P_k(ref, hyp) = \frac{1}{N-k}\sum_{i=1}^{N-k}[\delta_{ref}(i, i+k) \neq \delta_{hyp}(i, i+k)] \quad (13)$$

where N is the length of the reference segment and k is the window size. $\delta_{ref}(i,j)$ is the indicator function which evaluates to one if the boundary with sentence indices i and j are in the same segment, and zero otherwise. Similarly, $\delta_{hyp}(i,j)$ is one if the two indices are hypothesized to be in the same segment, and zero otherwise. Then, in (13), the outputs of $\delta_{hyp}(i,j)$ and $\delta_{ref}(i,j)$ are judged whether they are not equal, and the results are summed and divided by the number of all possible combinations to obtain P_k. P_k is a probability value between 0 and 1. From (13), it can be found that the higher the value of P_k, the more the false prediction is.

In this study, another metric used to evaluate text summarization models is Recall-Oriented Understudy for Gisting Evaluation [35] (ROUGE). ROUGE compares the generated summaries by the model with the reference summaries in the corpus to determine the similarity as a score. The common and relatively basic ROUGE has two kinds, namely ROUGE-N and ROUGE-L. The former compares the n-gram repetition rate of the generated summaries and the reference summaries, as shown in (14). The latter uses the length ratio of the longest common subsequence (LCS) to compare the generated summaries with the reference summaries.

$$\text{ROUGE-N} = \frac{\sum_{S\in\{\text{Re}\,ferenceSummaries\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{\text{Re}\,ferenceSummaries\}}\sum_{gram_n\in S} Count(gram_n)} \quad (14)$$

where $gram_n$ denotes n-gram, $ReferenceSummaries$ is the reference summary in the corpus, $Count_{match}(gram_n)$ is the number of n-grams that appear simultaneously in the generated summary and the reference summary, and $Count(gram_n)$ is the number of n-grams in the reference summary. Most of the summarization systems adopt ROUGE-1, ROUGE-2, and ROUGE-L as the evaluation metrics. In this study, ROUGE-1 uses the unigram to evaluate the amount of information contained in the generated summary, ROUGE-2 uses the bigram to measure the fluency of the generated summary, and ROUGE-L measures the relationship between the generated summary and the reference summary.

In this study, we also employed a subjective evaluation method [17] to compare the generated summary with the input text based on three criteria, namely fluency, relatedness, and faithfulness, and judged whether the quality of the generated summary is good or not. First, 100 pairs of input texts and generated summaries were randomly presented to the annotators for labeling, and the results were used to confirm the inter-annotator agreement. After confirming that the agreement was high enough, all the generated results that need to be evaluated were presented to the annotators. Each generated summary was only evaluated once, and the generated results from the same input were presented to the same annotator.

B. Dataset Description

For text segmentation dataset, WIKI-727K [25], collected from the articles of Wikipedia, provides a good benchmark for

evaluating text segmentation models. The existing text segmentation corpus is only available in English, so we also prepared a large Chinese text segmentation corpus, called ChWiki_181k, for evaluation. The articles in Wikipedia mark the hierarchy of various segments, so the granularity can be determined when arranging the dataset. The so-called granularity refers to the degree of division of segments, and this study used the finest one; that is, all classes were completely divided. Based on the above process, this study prepared two corpora, ChWiki_181k and Wiki_300 [24], from the Wikipedia with 90% as the training set and 10% as the test set as shown in Table II.

TABLE II
DETAILS OF THE WIKI_181K AND WIKI_300 DATABASE

	Wiki_181k	Wiki_300
Number of documents	181,926	300
Number of sentences	14,471,224	58,071
Number of segmentation points	331,097	2,234

For summarization dataset, the LCSTS dataset [14] was used directly for training the abstractive summarization model and the extractive model BERTSUM [11]. The data description of LCSTS is shown in TABLE III.

TABLE III
DETAILS OF LCSTS DATABASE

	Wiki_181k	Wiki_300
Number of documents	181,926	300
Number of sentences	14,471,224	58,071

C. Evaluation of the Text Segmentation Model

This study proposed the BERT-biLSTM model for the text segmentation task. In this model, the BERT model was first used to represent the boundary between two sentences with a segmentation point embedding, and then the bidirectional LSTM was applied to determine whether each boundary was a segmentation point or not based on the segmentation point embedding. In the implementation of the text segmentation model, the pre-trained Chinese BERT model released by Google Research was employed and the hyperparameters were set the same as the downloaded model, where the hidden state dimension was 768, the number of encoder layers was 12, and the number of heads was 12. Besides, the threshold α of BERT-biLSTM was set to 0.5, which meant that the segmentation points with a probability greater than 0.5 were regarded as a segment boundary. Recently, text segmentation had been formulated as a supervised learning task [24-25]. Previous research had proven that supervised learning was better than unsupervised learning in the text segmentation task. Therefore, this study proposed the BERT-biLSTM model based on supervised learning. In our experiment, Wiki_300, a small English text segmentation corpus, was used to train the BERT-biLSTM model, that made it easy to be compared to the most advanced text segmentation systems.

This experiment was to compare the performance of our proposed method and CNN-biLSTM with or without attention mechanism [24] using P_k indicator, as shown in TABLE IV. The results showed that the proposed BERT-biLSTM had a lower P_k than CNN-BiLSTM with or without attention mechanism [24]. This illustrates that BERT [26], the latest language representation model adopted by BERT-biLSTM, can improve the effectiveness of text segmentation.

TABLE IV
THE PERFORMANCE OF OUR PROPOSED METHOD AND CNN-BILSTM METHOD ON WIKI_300 DATASET

Method	P_k
CNN-BiLSTM [24]	0.328
CNN+Attn-BiLSTM [24]	0.344
BERT-biLSTM using sentence representation (proposed)	**0.284**
BERT-biLSTM using relation representation (proposed)	**0.299**

We also evaluated whether the model which considered the relationship between two sentences, represented as the segmentation point embedding, could improve the text segmentation result. In the experiment, BERT was used to represent a single sentence and a sentence pair, separately. The results in TABLE IV and TABLE V show that the effect of using single-sentence information as the segmentation point embedding was better. We think the reason is that using single-sentence information only needs to consider whether the previous sentence is the end of the segment or the latter sentence is the beginning of the segment to determine whether a sentence boundary is a segmentation point. Therefore, considering the relationship between sentences before and after the segmentation point did not improve the result. However, in TABLE V, it could be found that the effects of the two representation methods were almost the same, which implies that BERT-biLSTM also has a good ability in capturing the relationship between sentences.

For constructing a Chinese abstractive summarization system, ChWiki_181k, a larger text segmentation database compared to Wiki_300, was also used to train the proposed system. As ChWiki_181k was constructed by this study, there were no other systems based on this database. Therefore, this study proposed BERT-biLSTM as the benchmark model of ChWiki_181k for underlying research and comparisons, and the results are shown in TABLE V. In the results, BERT used the fine-tuning method for feature extraction, in which the latter did not need to fine-tune BERT during the training phase, so the training time was less. But it could be observed from TABLE V that the former method was still better, so the former was used in the proposed system.

TABLE V
THE PERFORMANCE ON CHWIKI_181K DATASET

Method	P_k
BERT-biLSTM with BERT fine-tuning and using sentence representation	**0.252**
BERT-biLSTM with BERT fine-tuning and using relation representation	**0.255**
BERT-biLSTM without BERT fine-tuning and using sentence representation	0.275
BERT-biLSTM without BERT fine-tuning and using relation representation	0.278

D. Evaluation on the Effect of the Extractive Model

The effect of BERTSUM was performed in the experiment, and the result is shown in TABLE VI. The document transformer model was a Transformer-based abstractive summarization model in which the purpose was to generate the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

headline summary in the database. Most of the hyperparameters used were the same as the original Transformer, except for the hidden dimension which was doubled to 1024 because of the longer input. TABLE VI shows that adding the extractive model could greatly improve the performance of the document Transformer model. The reason is that the extractive model reduces the length of the input article so that the document transformer model does not face the problem that the input and output lengths vary greatly, resulting in poor attention performance.

The proposed system consisted of the segment transformer and the document transformer in the two-stage model. Before the segment transformer model, the input text was first divided into segments with a shorter length by the text segmentation model, which was similar to the extractive model that reduces the input length. Therefore, the above two summarization models adopted Transformer because they both had inputs and outputs of similar length.

TABLE VI
ROUGE OF THE DOCUMENT TRANSFORMER MODEL

Method	Rouge-1	Rouge-2	Rouge-L
Document Transformer	23.48	7.78	21.95
BERTSUM+ Document Transformer	**37.00**	**17.86**	**37.72**

E. Evaluation on the Document Transformer Model

We compared the proposed model with the existing models trained using LCSTS [14]. The first benchmark model was the RNN and RNN-context [14]. Both were basic models proposed by LCSTS for subsequent research, using simple sequence-to-sequence models. RNN-context is the model with the attention mechanism. The second benchmark model was the DRGD [15]. For one of the advanced models on LCSTS, it used the variational autoencoder (VAE) to capture the latent structural information. The third benchmark model was the Dual-Train [17]. It applied a regularization approach in the traditional sequence-to-sequence model to enhance the semantic consistency between the generated summary and the input article.

The result is shown in TABLE VII. The proposed model combining the extractive model and the abstractive model successfully defeated RNN and RNN-context. The results proved that Transformer is more powerful than RNNs in the sequence-to-sequence model for the summarization task.

TABLE VII
ROUGE OF THE HEADLINE SUMMARY

Method	Rouge-1	Rouge-2	Rouge-L
RNN	21.50	8.90	18.60
RNN-context	29.90	17.40	27.20
DRGD	**36.99**	**24.15**	**34.21**
Dual-Train	36.20	24.30	33.80
BERTSUM (one sentence)	22.06	11.30	19.85
BERTSUM+ Document Transformer	**37.00**	17.86	32.72

The extractive model BERTSUM was also used for comparison. Because the reference summaries in the database were the headline summaries with a single sentence, the length of the extractive summary generated by BERTSUM was set to one sentence in order to make the length of the generated summary more like other models. By comparison, it can be found that the general abstractive summary model performed better than the extractive summary model, except for the basic RNN. Compared with the advanced models, the proposed model has similar scores of ROUGE-1 and ROUGE-L with DRGD and Dual-Train, which show that it has similar effects to the advanced models. However, the proposed model did not succeed in achieving similar ROUGE-2. This is because the extractive model BERTSUM generated less fluent summary, which was a serious disadvantage of the extractive model, and this also affected the summary generated by the document Transformer model. But in the proposed two-stage model, the variable-length summary was generated by the segment Transformer model, which took the segments instead of the extractive summary as the input, so the input was still fluent and did not encounter the above problem. Moreover, it can also be seen from the results that Transformer could improve the smoothness of the summary generated by the extractive model to a certain level, so Transformer was applicable to the abstractive model of the proposed system.

F. Evaluation of the Variable-Length Abstractive Summary

The final experiment was to evaluate the quality of the generated variable-length summary. TABLE VIII shows the result of ROUGE with different specified lengths and α values. The focus of the variable-length summarization system is that it can dynamically generate variable-length summaries, so we first evaluated the generated summaries with different lengths. The result shows that regardless of the value of α, the smaller the specified length, the higher the ROUGE. This is because that ROUGE is compared with the reference summaries in the corpus, but the reference summaries are headline summaries with only single sentences, which contain a limited amount of information. So, although some information is important, the reference summary may not be included. That is, a longer summary with much information resulting in lower ROUGE value. In addition, there is another reason that the input texts in LCSTS are short articles. Therefore, if the input is divided into too many segments, the amount of information in each segment is insufficient, resulting in a poor quality of the generated summary. However, according to the results from one to five sentences in length, it could be noticed that the generated summaries were stable, and there were no low scores.

TABLE VIII also shows the results by comparing different values of α. The ROUGE score was higher when α was 0.6, which proved that the loss values L_{seg} used in the two-stage model had a greater effects. When the value of α was smaller, it represented a higher proportion of L_{head}, so the generated summary was more like the headline summary in the corpus, which made the ROUGE value higher. However, if L_{head} had high proportion, the training of the two-stage model will be difficult to converge, and the quality of the generated summary will be poor. For traditional headline summary, according to the results in TABLE VIII and TABLE IX, the contribution of the segment Transformer model was greater than the contribution of the document Transformer model, so the proportion of L_{head} was lower than L_{seg} (α=0.6, ROUGE-1 score was 33.00;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3006731, IEEE/ACM Transactions on Audio, Speech, and Language Processing

ROUGE-2 score was 24.79; ROUGE-L score was 30.47). But considering the summaries with a length of 30% of the input length, the contributions of both models were equally important (α=0.5, accuracy is 70.0%).

It can be observed that the ROUGE value of the variable-length summary was lower than the headline summary with single sentence by comparing the results in TABLE VIII with the results in TABLE VII. The reason is related to the weaknesses of ROUGE mentioned above. ROUGE is limited to comparison with the reference summary in the corpus, while the reference summary in LCSTS is the headline summary with a single sentence, so ROUGE value of the headline summary in TABLE VII is higher. But the reference summary in the corpus and the variable-length summary in this study are not consistent, so it is desirable to design a metric that could achieve a more meaningful comparison.

TABLE VIII
ROUGE SCORE OF THE VARIABLE-LENGTH SUMMARY

	SUMMARY LENGTH	ROUGE-1	ROUGE-2	ROUGE-L
α=1.0	1 SENTENCE	2.76	0.00	2.76
	2 SENTENCES	2.76	0.00	2.76
	3 SENTENCES	2.76	0.00	2.76
α=0.6	1 SENTENCE	**33.00**	**24.79**	**30.47**
	2 SENTENCES	30.01	21.65	25.82
	3 SENTENCES	28.96	20.79	24.38
α=0.5	1 SENTENCE	24.65	13.18	22.10
	2 SENTENCES	21.41	10.37	18.44
	3 SENTENCES	20.03	10.02	17.26
α=0.4	1 SENTENCE	28.67	12.25	25.15
	2 SENTENCES	25.45	10.64	21.42
	3 SENTENCES	22.39	9.39	18.60
α=0.0	1 SENTENCE	2.76	0.00	2.76
	2 SENTENCES	2.76	0.00	2.76
	3 SENTENCES	2.76	0.00	2.76

Currently, there is no automatic evaluation metric to solve the above problem of ROUGE, so the subjective evaluation method [17] was adopted to confirm the quality of the generated variable-length summary. In this study, five native Chinese speakers were invited as annotators, and Fleiss' kappa was used to evaluate the inter-annotator agreement. The value of kappa was 0.76, which was found to be sufficiently consistent [37]. Because the data of LCSTS were short articles, it was not suitable for splitting the text into too many segments. And the above experiment also showed that the longer the specified length, the lower the ROUGE value. Therefore, the specified length was set from 1 to 3 sentences in subjective evaluation. Besides, summaries with a length of 30% of the input length were also considered.

In TABLE IX, the sequence-to-sequence model [17, 38] were based on single layer LSTM with an attention mechanism. The word embedding size was 400, and the hidden state size of the LSTM unit was 500. Self-Train [17] and Dual-Train [17] were implemented based on the he sequence-to-sequence model with two hyperparameters, the temperature τ (τ=2) and the soft training strength α (α=1). The result is shown in TABLE IX. In TABLE IX, we can find that the proposed variable-length abstractive summarization system achieved a maximum of 70.0% accuracy when α was 0.5 under subjective evaluation. This is because of the limited amount of information in the headline summary in the corpus. When α was 0.4, L_{head} had a relatively high proportion, but if the limited information contained in the headline summary was forced to generate a variable-length summary, the information contained in each segment summary was incomplete. Moreover, this may generate the summary with incomplete sentences.

TABLE IX
THE RESULTS OF SUBJECTIVE EVALUATION

Method	Summary length	# Good	# Total	Accuracy
Sequence-to-sequence [17, 38]	1 sentence	360	725	31.0%
Self-Train [17]	1 sentence	316		29.7%
Dual-Train [17]	1 sentence	389		**35.2%**
Proposed method with α=0.6	1 sentence	295		23.9%
	2 sentences	429		47.5%
	3 sentences	490		50.8%
	30% sentences	524		62.1%
Proposed method with α=0.5	1 sentence	333		27.0%
	2 sentences	490		54.3%
	3 sentences	608		63.0%
	30% sentences	**591**		**70.0%**
Proposed method with α=0.4	1 sentence	174		10.5%
	2 sentences	236		13.2%
	3 sentences	276		17.7%
	30% sentences	287		20.1%

When α was 0.5, the effect of the generated summary of one sentence was much worse than the summary with other lengths. The reason is that when the specified length is one sentence, it encounters the problem that the input and output lengths vary greatly, which is the same as the problems the abstractive summarization model faces mentioned above, and this causes the poor quality of the generated summary. However, when the specified length was 2 sentences, 3 sentences, and 30% of the input length, the result was good and successfully defeated the Dual-Train method [17]. One of the loss values, L_{seg}, considered the extractive summary as the target, so the generated summary had the extractive features, which made the consistency between the input article and the output summary be relatively high theoretically. In the future, if there is a corpus with multiple-sentence summaries corresponding to an article, α can be revised to a more appropriate value, so that the generated summary has more abstractive feature while maintaining the same high or even higher consistency.

TABLE X shows an example of the variable-length summary generated by the proposed system when α was 0.5. It can be observed that the variable-length summary in the example has good quality, and its content is fluent and is related to the input article. Moreover, the two-stage model used both extractive and abstractive summaries for training. Therefore, although the abstractive model was used to generate the summary, the generated variable-length summary still had both extractive and abstractive features. Nevertheless, on the other hand, the abstractive model made the two-stage model unrestricted to directly extract sentences from the input article like an extractive model. In the example of TABLE X, the two-stage model has the ability to generate "喬布斯向人們展示 iphone (Jobs shows people iphone)" word by word. If using the extractive summarization model, only the entire sentence of "2007 年喬布斯向人們展示 iPhone 並宣稱它將會改變世界

(In 2007, Jobs showed people the iPhone and declared that "it will change the world")" can be generated, which contains a lot of redundant contents.

TABLE X
AN EXAMPLE OF THE GENERATED VARIABLE-LENGTH SUMMARY

Input document	2007 年喬布斯向人們展示 iPhone 並宣稱「它將會改變世界」，還有人認爲他在誇大其詞，然而在 8 年後，以 iPhone 爲代表的觸屏智能手機已經席捲全球各個角落。未來，智能手機將會成爲「真正的個人電腦」，爲人類發展做出更大的貢獻。(In 2007, Jobs showed people the iPhone and declared that "it will change the world". Others think that he is exaggerating. However, eight years later, the touch-screen smart phone represented by the iPhone has swept the world. In the future, smartphones will become "real PCs" and make greater contributions to human development.)
Reference summary	經濟學人：智能手機將成爲「真正的個人電腦」(The Economist: Smartphones will become "real PCs".)
Variable-length summary (length=2)	喬布斯向人們展示 iphone，智能手機將會成爲真正的個人電腦爲人類發展做出更大的貢獻 (Jobs shows people iphone, smart phones will become real personal computers to make greater contributions to human development.)

IV. CONCLUSION AND FUTURE WORK

The study proposes a variable-length abstractive summarization model composed of a text segmentation model, an extractive summarization model, and an abstractive summarization model. In the proposed model, users can determine the length of the summary according to their needs. Additionally, we propose a two-stage model that can use the existing corpus containing only the single-sentence headline summary to train the variable-length abstractive summarization model. The most advanced sequence-to-sequence model, Transformer and the language representation model BERT based on the encoder of Transformer are adopted in the proposed system. The models in the proposed system achieved a better or comparable performance in both objective and subjective evaluations compared to the advanced models in the previous studies. Besides, a new large Chinese text segmentation corpus ChWiki_181k is constructed, and a benchmark model BERT-biLSTM is proposed for follow-up research and comparisons.

However, although the proposed two-stage model successfully uses the existing corpus to train the variable-length abstractive summarization model, there are still several problems due to the limitations of the corpus. First, the existing Chinese text summarization corpus is a headline summary corresponding to an article, so the model trained by the corpus can only generate headline summaries that contain very limited information. The amount of information is also insufficient for the variable-length summary. Moreover, given an article in the real world, the summary written by different people would be very different. Therefore, if the Chinese summarization corpus can contain multiple-sentence summaries in one article, it can be more realistic and also enhance the performance of variable-length abstractive summaries. The second problem is that most of the input text in LCSTS are short articles, which limit the ability of the variable-length abstractive summarization system. Short article cannot be cut into too many segments and the amount of information contained in each segment may be insufficient. Hence short article may result in a poor quality of variable-length summaries.

In addition, the number of output segments in the proposed variable-length summarization model must be specified by the user. The model cannot automatically detect the amount of information contained in the input article. The current training method treats inputs with longer length as containing more information and is therefore divided into more segments. However, there is no absolute relationship between a longer length and more information. In this way, some segments may contain almost no useful information, causing the model to generate poor results.

REFERENCES

[1] S. Primativo, D. Spinelli, P. Zoccolotti, M. De Luca, and M. Martelli, "Perceptual and cognitive factors imposing "speed limits" on reading rate: a study with the rapid serial visual presentation," *PloS one*, vol. 11, no. 4, p. e0153786, Apr. 2016.

[2] D. Roth, "The Answer Factory: Demand Media and the Fast, Disposable, and Profitable as Hell Media Model," [Online], 2019, Available: https://www.wired.com/2009/10/ff-demandmedia/ (accessed 19 June, 2019).

[3] 運動公社, "從查比高恩斯空難與轉會流言——體育新聞的內容農場現象," [Online], 2019, Available: https://bit.ly/2PXK8ze (accessed 19 June, 2019).

[4] X. Wang, Y. Yoshida, T. Hirao, K. Sudoh, and M. Nagata, "Summarization based on task-oriented discourse parsing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1358-1367, May 2015.

[5] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159-165, Apr. 1958.

[6] K. Y. Chen, S. H. Liu, B. Chen, H. M. Wang, E. E. Jan, W. L. Hsu, and H. H. Chen, "Extractive broadcast news summarization leveraging recurrent neural network language modeling techniques," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1322-1334, May 2015.

[7] S. Yan, and X. Wan, "SRRank: leveraging semantic roles for extractive multi-document summarization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 2048-2058, Dec. 2014.

[8] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, "Query-oriented text summarization using sentence extraction technique," in *Proc. the 4th International Conference on Web Research (ICWR)*, 2018, pp. 128-132.

[9] K. Al-Sabahi, Z. Zuping, and M. J. I. A. Nadher, "A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS)," *IEEE Access*, vol. 6, pp. 24205-24212, Apr. 2018.

[10] K.-Y. Chen, S.-H. Liu, B. Chen, H.-M. Wang, "An Information Distillation Framework for Extractive Summarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 161-170, Jan. 2018.

[11] Y. Liu, "Fine-tune BERT for Extractive Summarization," arXiv preprint [Online], 2019, Available: arXiv:1903.10318.

[12] J. Zhang, Y. Zhou, and C. Zong, "Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 10, pp. 1842-1853, Jun. 2016.

[13] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93-98.

[14] B. Hu, Q. Chen, and F. Zhu, "Lcsts: A large scale chinese short text summarization dataset," arXiv preprint [Online], 2015, Available: arXiv:1506.05865, 2015.
[15] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," arXiv preprint [Online], 2017, Available: arXiv:1708.00625.
[16] H. Zhang J. Xu, and J. Wang, "Pretraining-Based Natural Language Generation for Text Summarization," arXiv preprint [Online], 2019, Available: arXiv:1902.09243, 2019.
[17] B. Wei, X. Ren, Y. Zhang, X. Cai, Q. Su, and X. Sun, " Regularizing output distribution of abstractive Chinese social media text summarization for improved semantic consistency," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 18, no. 3, pp. 31:1-31:15, Jul. 2019.
[18] C. Li, W. Xu, S. Li, and S. Gao, "Guiding generation for abstractive text summarization based on key information guide network," in *Proc. the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 55-60.
[19] H. Wang, L. Jing, and H. Shao, "Research on method of sentence similarity based on ontology," in *Proc. 2009 WRI Global Congress on Intelligent Systems*, vol. 2: IEEE, 2009, pp. 465-469.
[20] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," arXiv preprint [Online], 2016, Available: arXiv:1602.06023.
[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," in *Proc. Advances in neural information processing systems (NIPS)*, 2017, pp. 5998-6008.
[22] M.-H. Su, C.-H. Wu, and L.-Y. Chen, "Attention-based Response Generation Using Parallel Double Q-Learning for Dialog Policy Decision in a Conversational System," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 131-143, Oct. 2019.
[23] E. Loper and S. Bird, "NLTK: the natural language toolkit," arXiv preprint [Online], 2002, Available: arXiv:cs/0205028.
[24] P. Badjatiya, L. J Kurisinkel, M. Gupta, and V. Varma, "Attention-Based Neural Text Segmentation," in Advances in Information Retrieval, Cham, G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, Eds., 2018: Springer International Publishing, pp. 180-193.
[25] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, "Text segmentation as a supervised learning task," arXiv preprint [Online], 2018, Available: arXiv:1803.09337.
[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint [Online], 2018, Available: arXiv:1810.04805, 2018.
[27] M.-H. Su, C.-H. Wu, and Y. Chang, "Follow-Up Question Generation using Neural Tensor Network-based Domain Ontology Population in an Interview Coaching System," in *Proc. INTERSPEECH*, 2019, pp. 4185-4189.
[28] M.-H. Su, C. -H. Wu, K.-Y. Huang, and W.-H. Lin, "Response Selection and Automatic Message-Response Expansion in Retrieval-Based QA Systems using Semantic Dependency Pair Model," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 1, pp. 3:1-3:24, Nov. 2018.
[29] M.-H. Su, C.-H. Wu, K.-Y. Huang, Q.-B. Hong, and H.-H. Huang, "Follow-up Question Generation using Pattern-based Seq2seq with a Small Corpus for Interview Coaching," in *Proc. INTERSPEECH*, 2018, pp. 1006-1010.
[30] M.-H. Su, C.-H. Wu, K.-Y. Huang, and C.-K. Chen, "Attention-based Dialog State Tracking for Conversational Interview Coaching," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6144-6148.
[31] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 1243-1252.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv preprint [Online], 2016, Available: arXiv:1607.06450.
[34] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," arXiv preprint [Online], 2017, Available: arXiv:1703.03906.
[35] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Text summarization branches out*, 2004, pp. 74-81.
[36] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine learning*, vol. 34, no. 1-3, pp. 177-210, Feb. 1999.
[37] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159-174, Mar. 1977.
[38] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Advances in neural information processing systems*, 2014, pp. 3104-3112.

Ming-Hsiang Su received the B.S. degree in computer science and information engineering from the Tunghai University, Taichung, Taiwan, in 2001, the M.S. degree in management information systems from the National Pingtung University of Science and Technology, Pingtung, Taiwan, in 2003, and the Ph.D. degree in computer science and information engineering from Chung Cheng University (NCKU), Tainan, Taiwan, in 2013. He is currently a Postdoctoral Fellow with the Department of Computer Science and Information Engineering, NCKU. His research interests include e-learning, artificial intelligence, machine learning, multimedia signal processing, and personality detection.

Chung-Hsien Wu (SM'03) received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1987 and 1991, respectively. Since 1991, he has been with the Department of Computer Science and Information Engineering, NCKU. He became the Chair Professor in 2017. He served as the Deputy Dean of the College of Electrical Engineering and Computer Science, NCKU, from 2009 to 2015. He also worked at Computer Science and Artificial Intelligence Laboratory of Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in summer 2003, as a Visiting Scientist. He was an Associate Editor of the *IEEE Transactions on Audio, Speech and Language Processing* (2010–2014) and the *IEEE Transactions on Affective Computing* (2010–2014). He is currently an Associate Editor of *ACM Transactions on Asian and Low-Resource Language Information Processing*. Currently, he is the APSIPA BoG Member (2019~2021).
He received 2018 APSIPA Sadaoki Furui Prize Paper Award in 2018, and the Outstanding Research Award of Ministry of Science and Technology, Taiwan, in 2010 and 2016. His research interests include deep learning, affective computing, speech recognition/synthesis, and spoken language processing.

Hao-Tse Cheng received the B.S. degree and the M.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2017. His research interests include multimedia signal processing and natural language processing.