


Article

An Abstract Summarization Method Combining Global Topics

Zhili Duan, Ling Lu *, Wu Yang, Jinghui Wang and Yuke Wang 

College of Computer Science and Engineering Chongqing, Chongqing University of Technology,
Chongqing 400050, China

* Correspondence: ll@cqut.edu.cn; Tel.: +86-139-8379-1161

Featured Application: Provide a technical basis for the ongoing application of sentiment analysis, opinion extraction, automatic question answering system, etc.

Abstract: Existing abstractive summarization methods only focus on the correlation between the original words and the summary words, ignoring the topics' influence on the summaries. To this end, an abstract summarization method combining global topic information, ACGT, is proposed. A topic information extractor, based on Latent Dirichlet Allocation, is constructed to extract key topic information from the original text, and an attention module is built to fuse key topic information with the original text representation. The summary is then generated by combining a pointer generation network and coverage mechanism. With evaluation metrics of ROUGE-1, ROUGE-2, and ROUGE-L, the experimental results of ACGT in the English dataset CNN/Daily Mail are 0.96%, 2.44%, and 1.03% higher than the baseline model, respectively. In the Chinese dataset, LCSTS, ACGT shows a higher performance than the baseline method by 1.19%, 1.03%, and 0.85%, respectively. Our results demonstrate that the performance of summaries is significantly correlated with the number of topics that are introduced. Case studies show that the introduction of topic information can improve both the coverage of original text topics and the fluency of summaries.



Citation: Duan, Z.; Lu, L.; Yang, W.; Wang, J.; Wang, Y. An Abstract Summarization Method Combining Global Topics. *Appl. Sci.* **2022**, *12*, 10378. <https://doi.org/10.3390/app122010378>

Academic Editor:
Rafael Valencia-García

Received: 6 August 2022
Accepted: 5 October 2022
Published: 14 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: natural language processing; automatic text summarization; topic model; pointer generation network; BERT

1. Introduction

Automatic summarization [1] technology is a technique that uses computers to understand and analyze text in order to generate concise summaries covering topics of the original text. It is an important research direction in Natural Language Processing (NLP), and also a pre-task for many downstream applications, such as automated question and answer systems [2] and news headline generation [3].

Existing automatic summarization methods can be divided into two main categories: extractive and abstractive. Extractive methods constitute a summary by extracting important text units from the original text. The abstractive method is more similar to manual summarization methods. It restates the original text via techniques such as synonymous substitution and sentence abbreviation, thus resulting in summaries that mostly contain words or phrases other than the original text, and also with better fluency than summaries produced by extractive methods.

The Sequence to Sequence [4] (Seq2Seq) summary generation framework incorporating deep neural networks has been widely studied in recent years. Moreover, the Seq2Seq method with an attention mechanism [5] purposefully addresses the problem of gradient disappearance caused by excessively long sentences, thus improving the performance of generated summaries.

The performance of summary generation can also be affected by out-of-vocabulary (OOV) and redundant words. To this end, See et al. [6] proposed a pointer generator network. The method suggests copying words from the original text or generating new

words from a fixed vocabulary, and uses a coverage mechanism to alleviate the word redundancy problem, thus improving the performance of summary generation.

In general, the Seq2Seq framework combined with an attention mechanism provides the basis for studies of automatic summaries. However, existing approaches consider summary generation as a translation process from the original text to the summary; moreover, most of the attention mechanism is built between characters of the original text and the summary characters, with less research from the perspective of the topics of the original text. In terms of evaluation metrics of automated summaries, ROUGE [7], which is widely used, is a recall-based evaluation metric. It evaluates the n-tuple contribution statistics of the generated summary and the reference summary, without evaluating the topic consistency between the summary and original text, which may result in topics of the summaries deviating from the original text despite high ROUGE scores.

We argue that the process of manually writing summaries is to read and understand the full text to find the words and phrases that are most relevant to the original topic, and then write a summary that matches the original topic. Therefore, we propose an Abstract Summarization method Combining Global Topics (ACGT) to improve the performance of the summary by constructing a global topic information extractor and an attention module that combines topic information. Our main achievements in this paper are as follows:

1. We propose a summary generation method incorporating global topic information, model the topic of the document, and update the document representation by fusing the topic information of the text with word embeddings through the information fusion module.
2. We propose the nTmG (n-TOPIC-m-GRAM) method to extract the key topic information in the original text. The essential method is used to avoid the noise caused by the introduction of the topic.
3. Empirical studies show that the proposed method demonstrates a more advanced performance than baseline methods. It also shows that the number of incorporated topics is tightly correlated with the performance of generating summaries, which provides empirical evidence for subsequent automatic summary studies combining topics.

2. Related Works

The mainstream automatic summarization methods can be divided into two types: extractive and abstractive. Extractive methods involve extracting words and sentences and other semantic units from the original text. Representative extractive methods include semantic-information-based methods [8] and structural-information-based methods [9,10]. Abstractive summarization methods are closer to manual summarization, which restates the original text with words, sentences, and phrases that are different from the original text. It is a challenge to the model's ability of understanding, representing, and generating. In recent years, the application of deep neural networks and attention mechanisms in machine translation research has promoted the research of abstractive summarization methods with an encoder–decoder structure. In 2015, Rush et al. [5] first introduced an encoder–decoder structure and attention mechanism to the abstractive summarization task. Later, Nallapati et al. [11] combined an attention mechanism with RNN, and utilized information such as stop words and document structure, and its ROUGE-L improved by 1.47% compared to Rush on DUC-2004 and gigaword datasets.

In 2018, Gehrmann S et al. [12] used a content selector as a bottom-up attention step to constrain the model to likely phrases. This approach improves the ability to compress text, while still generating fluent summaries. Celikyilmaz A et al. [13] introduced deep communication agents in an encoder–decoder architecture; these are divided into cooperative agents, each of which is responsible for representing a part of the text, to solve the challenges of representing a long document for abstractive summarization. Empirical results show that multiple communication encoders produce higher quality summaries compared to baseline methods.

In addition, the generation of OOV words and redundant words also has a significant impact on summary generation. To this end, Gulcehre et al. [14] divided OOV words into two strategies, one is to find similar words in the preset vocabulary, the other is to use original text words instead, and the decoding method is judged by a two-layer perceptron during generation. Gu et al. [15] proposed COPYNET, which added the probabilities of the output words of two modules, Generate-Mode and Copy-Mode, at the decoder to obtain the distribution of the final words. Vinyals et al. [16] presented a pointer network that uses the weights of the input sequence as a pointer, and outputs the word probability distribution for the input sequence. Furthermore, See et al. [6] proposed a method combining generation and replication to solve the problem of OOV words, and introduced a coverage mechanism to solve redundant words.

In order to improve the performance of the abstract, researchers have analyzed the relationship between the original text and the abstract from different perspectives. Ruan Q et al. [17] proposed a novel approach to formulate, extract, encode, and inject hierarchical structure information explicitly into an extractive summarization model. The HiStruct model outperforms baseline collectively on CNN/Daily Mail, PubMed, and arXiv. Mao Z et al. [18] present a dynamic latent extraction approach for abstractive summarization. The model treats the extracted text segments as latent variables and employs dynamic segment-level attention weights during decoding. Experimental results show that DYLE outperforms all existing methods on GovReport and QMSum.

In recent years, researchers have considered the summary generation process as machine translation, and have proposed many models. However, there are still significant differences between summary generation and machine translation. It is sufficient for the summary to retain only the key information of the original text, rather than all of it. Thus, it would be better if the summary had good coverage of the topics of the original text. It is inadequately represented in current evaluation metrics since the widely used ROUGE metric is a recall-based approach.

Therefore, researchers have carried out topic-oriented research. Li J et al. [19] proposed UCTOPIC, a novel unsupervised contrastive learning framework for context-aware phrase representations and topic mining. It outperforms the state-of-the-art phrase representation model by 38.2% NMI on average on four entity clustering tasks. Bahrainian S A et al. [20] introduced the first topical summarization corpus NEWTS, based on the well-known CNN/Daily Mail dataset, and annotated via online crowd sourcing. The goal was to create datasets that support topic-focused summarization tasks, and then study the relationship between original text topics and summaries. Li M et al. [21] proposed a hierarchical contrastive learning mechanism to unify the mixed granularity of semantic meaning in the input text, including common vocabulary and topic vocabulary.

Subsequently, researchers introduced the Latent Dirichlet Allocation (LDA) topic model [22] to the summarization task. Wu D et al. [23] proposed an extractive summarization method based on LDA, which calculates the sentence weights according to the position and title of the sentence in the document, and extracts sentences according to the weights to form summaries. Liu Na et al. [24] presented a multi-document extractive method based on LDA important topics. In terms of abstractive summarization, Yang Tao et al. [25] proposed a hybrid summarization model based on topic awareness, adding document topics to help in summary generation. Guo Ji-Feng et al. [26] used LDA to obtain topic words, constructed a composite attention mechanism, and combined it with a Generative Adversarial Network (GAN) [27] to generate summaries. These past methods have constructed many fusion modules and performed multistep attention processes, making the work more complicated. Yang Tao et al. [25] proposed a topic-aware summary generation method for long texts, which produces summaries with better ROUGE scores than their baseline methods, but introduce noise by including all topics of the document.

Comparing previous works, many fusion modules are constructed, multistep attention is carried out, and the work is more complicated. Specifically, in 2021, Yang Tao et al. [25] proposed a long text summary generation method based on topic awareness. Although

the ROUGE value of the generated summary is better than his baseline method, noise is introduced due to the introduction of all the topics of the document into the model, which makes the generated summary redundant. This inspired us to propose a summary generation method that combines global topic information. Our method focuses on topic information that has a significant impact on the original text, and succinctly integrates it into the representation of the original text to improve the performance of summary generation. The proposed model is effectively validated with two standard datasets.

3. Motivation

The purpose of the summary is to generate a short overview text that clarifies the important points of the article. The summary should cover the global topic information of the original text. The global topic information mentioned in this paper refers to the part of the document topics that have an important impact on summary generation. The relationship between global topic information and a summary is analyzed below.

In 2003, Blei et al. [22] proposed the LDA, which provides a method for discovering the underlying topics of documents. In recent years, LDA has been introduced into automatic summarization tasks by many researchers, and has achieved advanced results [28,29]. LDA is also the basis for our proposed ACGT model. We believe that for any document d , its word sequence is encoded as V , and the document topic distribution vector T of d generated by the LDA model reflects the global topic of d , and T can be integrated into V to obtain a fusion of global topic information V' , which is used in the decoder. Therefore, we need to focus on two issues, one is to avoid noise caused by the introduction of the topic, and the other is to improve the effectiveness of the topic's introduction.

In this paper, we analyze more than 280,000 pieces of data in the CNN/Daily Mail dataset, and the topic relevance of the original text and summary is shown in Figure 1. It shows that there are 183,794 items of the TOP 1 topic in the original text that are also TOP 1 in the target summary, accounting for 63%. This suggests that the TOP 1 topic of the original text also appears in the summary with a high probability, which inspired us to propose a method to guide summary generation with the original text TOP N topics.

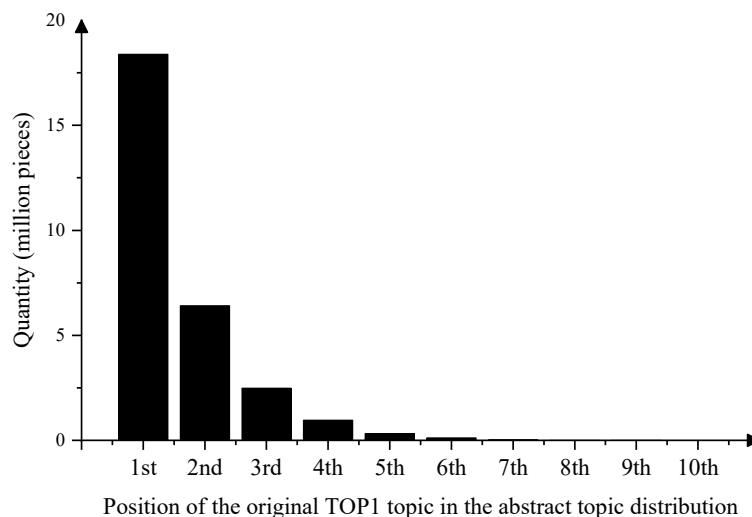


Figure 1. Distribution of original text TOP 1 topics in summary topics. The horizontal axis is the order of the original text TOP 1 topics in its summary topic, and the vertical axis is the number of the original text TOP 1 topics in summary topic distribution.

To avoid the noise caused by the introduction of topics, we propose the nTmG (n-TOPIC-m-GRAM) method, which chooses TOP m items with the highest probability from the n topics with the largest probability distribution in the original text. Then, to enhance the effectiveness of nTmG introduction, we fuse the mean vector of words with the original representation by an attention mechanism.

4. Approach

The ACGT summary generation model proposed in this paper is depicted in Figure 2. The encoder mainly includes the Global Topics Extractor and the Global Topics with combined Attention Module. It outputs the context vector containing the global topic information to the decoder, which is used to guide summary generation. The decoder uses the pointer network and coverage mechanism to solve the problem of OOV words and redundant words.

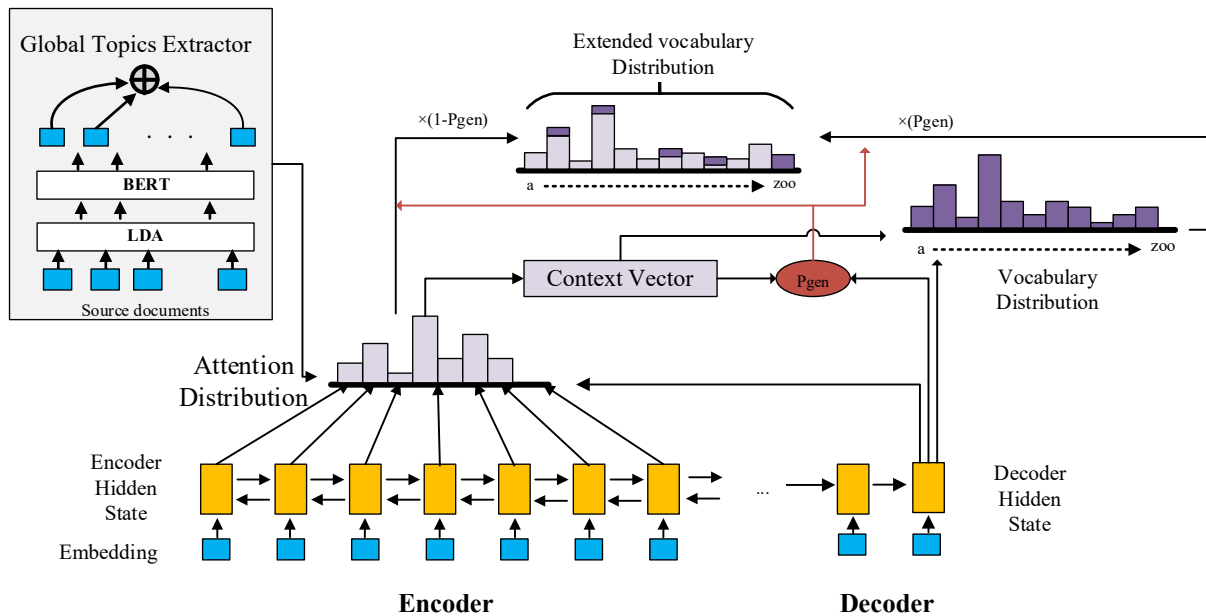


Figure 2. An abstract summarization method combining global topics.

4.1. Global Topics Extractor

To extract the key topic information of the document, we construct a topic information extractor based on LDA. We use LDA combined with the nTmG method to extract the key topic words of the original text, and then re-encode the key topic information through the information fusion unit to obtain the global topic vector and then update attention. The probabilistic graphical model of LDA-generated documents is shown in Figure 3.

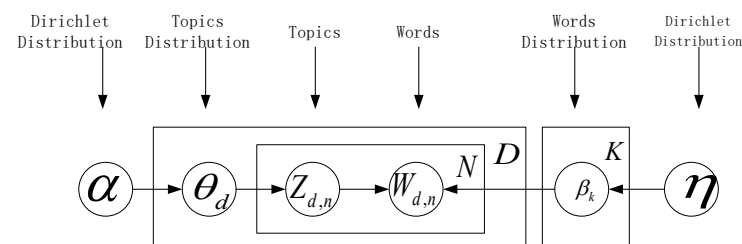


Figure 3. Probabilistic model diagram of LDA.

For the input document set $D = \{d_1, d_2, \dots, d_i, \dots, d_s\}$, the latent variable K represents the number of topics implied in D , α represents the strong and weak relationship between the hidden topics in D , and η is the probability distribution of all implied topics; α and η both obey the Dirichlet distribution. When LDA generates a document, it selects the document according to the prior probability, determines the number of characteristic words of the document, samples the topic distribution of the document, selects the topic of the words in the document, and then samples the word distribution of the current topic, and finally samples the generated word. According to the LDA probability graph model, for the

document $d_i \in D$, the generation probability of the n -th word $W_{d,n}$ is calculated according to:

$$\theta_d = \text{Dirichlet}(\alpha) \quad (1)$$

$$Z_{d,n} = \text{multi}(\theta_d) \quad (2)$$

$$\beta_k = \text{Dirichlet}(\eta) \quad (3)$$

where θ_d is the topic distribution of the document d_i , β_k is the word distribution of the current topic, then θ_d and β_k are sampled from the Dirichlet distribution α and η , respectively. $Z_{d,n}$, the topic of $W_{d,n}$ is sampled from the multinomial distribution of θ_d .

In this paper, we use the Gibbs sampling algorithm to train the parameters of the topic distribution and word distribution β of the LDA model. First, determine the number of topics K , and randomly assign a topic number Z to each word in the corpus; Rescan the corpus, and update the topic number for each word with the Gibbs sampling formula, And repeat the aforementioned sampling steps until convergence. Finally, count the word topics in the corpus, get the topic distribution θ_d of the document, count the distribution of each topic word, and get the word distribution β_k of the topic.

We use the LDA model to obtain the topic probability distribution of document set D and the word probability distribution under each topic. For the document $d_i \in D$, we take the topic with the probability of TOP n , then choose the terms with the probability of TOP m under the topic of TOP n to form the key topic information word set $Top = \{t_{11}, t_{12}, \dots, t_{1m}, \dots, t_{nm}\}$. Experimental demonstrated that BERT has achieved state-of-the-art performance in the summary task. We use the BERT pretraining model to obtain the embeddings representation of Top , and finally, take its mean vector G_T as the key topic information vector of the document. These are calculated according to:

$$X = f_{BERT}(Top) \quad (4)$$

$$G_T = \frac{1}{nm} \sum_{i=1}^{nm} X_i \quad (5)$$

where f_{BERT} is the nonlinear equation of BERT and $X = \{x_{11}, x_{12}, \dots, x_{1m}, \dots, x_{nm}\}$ is the vector set of key topic words encoded by BERT. In $G_T \in R^{(b*l*d)}$, b represents the model training batch, l represents the number of key topic words, and d represents the dimension of the last hidden layer of BERT.

4.2. Global Topics with Combined Attention Module

Based on the way the human brain processes information overload, Rush et al. [5] introduced an attention mechanism to improve the ability of neural networks to process information.

The essence of the attention mechanism is an addressing process. Its calculation can be divided into three steps. The first step is information input, the second step is to calculate the attention distribution α , and the third step is to calculate the weighted average of the input information according to the attention distribution α . However, traditional attention mechanisms generally pay attention to document information within a certain distance, while not paying attention to some important information. To this end, we propose an attention module that combines global topic information, and its structure is shown in Figure 4.

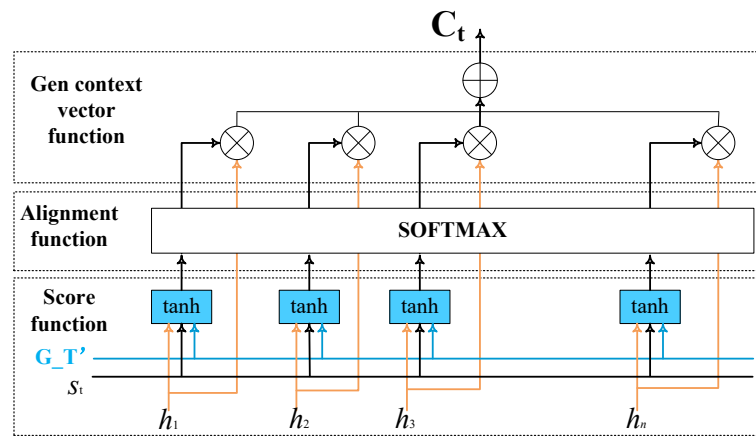


Figure 4. Global Topics with combined Attention Module.

We perform attention computation using Bahdanau attention [30], where the hidden layer state of the encoder and the key topic vector are jointly taken as the input of the attention module. First, linearity G_T into the same dimension as the original word embedding to obtain the key topic information vector $G_{T'}$:

$$G_T \rightarrow G_{T'} \tag{6}$$

where $G_{T'} \in R^{(b \times l \times d)}$, in which b denotes the model training batch, l denotes the document length, and d denotes the hidden layer dimension of the word vector.

Next, the key topic information vector $G_{T'}$, the encoder hidden layer state h_i , and the decoder hidden layer state s_t pass through a linear layer and tanh activation function, and then the correlation weights are obtained. Finally, the Score function is calculated according to:

$$Att(h_i, s_t, G_{T'}) = v^T \tanh(W_h h_i + W_s s_t + W_{g_t} G_{T'} + b_{attn}) \tag{7}$$

The attention distribution \hat{a} is obtained by alignment function as follows:

$$\begin{aligned} \hat{a} &= softmax(Att(h_i, s_t, G_{T'})) \\ &= \frac{\exp(Att(h_i, s_t, G_{T'}))}{\sum_{k=1}^T \exp(Att(h_i, s_t, G_{T'}))} \end{aligned} \tag{8}$$

where Att is additive attention using a single layer feedforward neural network, W_h , W_s , and W_{g_t} are learnable parameter matrices, and b_{attn} is bias term.

We take the weighted summation of the attention distribution (\hat{a}) and the encoder hidden layer state of each word $h = \{h_1, h_2, \dots, h_i, \dots, h_n\}$, to obtain the context vector C_t that fuses the key topic information to generate context vector functions as follows:

$$C_t = \sum_{i=1}^T \hat{a}_i h_i \tag{9}$$

4.3. Pointer Network and Coverage Mechanism

For the sequence of target summary words $R = [r_1, r_2, \dots, r_t, \dots, r_k]$, the decoder uses a single unidirectional LSTM to calculate the decoder's hidden layer state s_t :

$$y_t = e^w(r_t) \tag{10}$$

$$s_t = LSTM(s_{t-1}, y_{t-1}, C_t) \tag{11}$$

where e^w is the word embeddings representation, y_t is the representation of the summary words r_t at time step t , $y_t \in R^m$, m is the word embedding dimension, and s_t is the decoder hidden layer state at time step t ; in $s_t \in R^n$, n is the hidden layer state dimension.

For a given sequence, the conditional probability $P(w)$ of the output of each word in the vocabulary is calculated according to:

$$P_{vocab} = softmax(V'(V[s_t; C_t] + b) + b') \tag{12}$$

$$P(w) = P_{vocab}(w) \tag{13}$$

where $[;]$ denotes vector splicing, P_{vocab} is the probability distribution of all words in the preset vocabulary, V' and V are the learnable parameter matrix, and b and b' are bias terms.

4.3.1. Pointer Network

The traditional Seq2Seq model cannot solve the problem of OOV words; therefore, See et al. [6] proposed the method of pointer generation network, which searches for the element with the largest weight of the current input sequence in each step of prediction, when the output sequence is completely derived from the input sequence, so it can adapt to the change of the length of the input sequence and solve the problem of OOV words.

We use a pointer network to solve the OOV words problem. At each decoder time step t , the pointer network calculates the generation probability $P_{gen} \in [0, 1]$ using the context vector C_t , the decoder hidden layer state s_t and the input y_t at the decoder, which represents the probability of generating words from a fixed vocabulary, and helps the model in determining whether to generate words from the vocabulary or copy words from the input sequence of the original text. P_{gen} is calculated as shown:

$$P_{gen} = \sigma(W_{ct}C_t + W_{st}s_t + W_{yt}y_t + b_{ptr}) \tag{14}$$

where W_{ct} , W_{st} , and W_{yt} are learnable parameter matrices, b_{ptr} is the bias term, and σ is the sigmoid activation function.

For each document, P_{gen} weighted sums of the vocabulary distribution and the attention distribution are used to obtain an extended vocabulary on which the probability of generating word w is as follows:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \hat{\alpha}_i^t \tag{15}$$

where P_{vocab} is the probability of generating words on the predefined vocabulary. If w is a word outside the vocabulary, $P_{vocab}(w)$ is zero; if w does not appear in the original text, then $\sum_{i:w_i=w} \hat{\alpha}_i^t$ is zero.

4.3.2. Coverage Mechanism

To alleviate the problem of generating repetition words, a coverage mechanism is introduced. The coverage vector, which is the sum of all attention distributions, is defined to record the coverage of words received from the attention mechanism up to the current time step. c^t is calculated according to:

$$c^t = \sum_{t'=0}^{t-1} \hat{\alpha}^{t'} \tag{16}$$

where $\hat{\alpha}^t$ is the attention distribution at time step t .

To take into account the attention weights within a certain step size before the current time step, we introduce an additional input when calculating the attention distribution, changing Equation (7), as follows:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + W_{g-t} G_{-T'} + W_c c_i^t + b_{attn}) \tag{17}$$

where W_h , W_s , W_c , and W_{g-t} are learnable parameter matrices, and b_{attn} is bias term.

4.4. Loss Function

During training, the loss at time step t uses the negative log likelihood of the target word w_t , as follows:

$$loss_t = -\log P(w_t) \quad (18)$$

The overall loss of the input sequence is:

$$loss = \frac{1}{T} \sum_{t=0}^T loss_t \quad (19)$$

where T is the total length of the target summary.

Furthermore, an additional coverage loss is calculated in the coverage mechanism to penalize words with too much attention:

$$covloss_t = \sum_i \min(\hat{\alpha}_i^t, c_i^t) \quad (20)$$

where $covloss_t \leq \sum_i \hat{\alpha}_i^t = 1$. Therefore, coverage loss is weighted by the hyperparameter λ is added to the loss function to obtain the final loss function is calculated according to:

$$loss_t = -\log P(w_t) + \lambda \sum_i covloss_t \quad (21)$$

5. Experiments

5.1. Dataset

Experiments are conducted on the English long paragraph dataset and Chinese short text dataset, respectively. The English dataset is CNN/Daily Mail (CNN/DM) [31], which contains 287,227 training data and 11,490 validation data. The basic statistics are shown in Table 1. The average length of the original text in the training set is 766, with a total of 29.74 sentences, and the average length of the target abstract is 53, with a total of 3.72 sentences, and the ratio of the length of the abstract to the original text is 1/14.45.

Table 1. Length statistics of CNN/DM.

Avg Length	CNN			DM		
	Train	Verify	Test	Train	Verify	Test
Text	762	763	716	813	774	780
Sum.	66	67	67	67	66	65

The Chinese dataset is LCSTS (Large-scale Chinese Short Text Summarization dataset), which is contributed by Hu [32] based on the content published by authoritative certified users, such as China Daily on Weibo, with a scale of more than 2 million. The dataset consists of three parts, as shown in Table 2. Part I is the training set, and part II is randomly sampled from part I, with 1 to 5 manual scores added; 1 indicates the lowest correlation between the document and the abstract, and 5 indicates the highest. Part III is independent of the first two parts, and also has 1–5 manual scores. To make the comparison experiment fair, referring to the dataset used by the baseline model [32], this paper takes part I as the training set, and the data with more than 3 points of part III as the test set, for the experiment.

Table 2. Statistics of LCSTS.

Set	Text, Summary) Pairs	
part I	2,400,591	
part II	number of pairs	10,666
	manual score 1	942
	manual score 2	1039
	manual score 3	2019
	manual score 4	3128
part III	manual score 5	3538
	number of pairs	1106
	manual score 1	165
	manual score 2	216
	manual score 3	227
	manual score 4	301
	manual score 5	197

5.2. Evaluation Metrics

We use the standard ROUGE-1, ROUGE-2, and ROUGE-L metrics [33] to measure summary qualities. The ROUGE-N is calculated according to:

$$R_{ROUGE-N} = \frac{\sum_{S \in \{Ref\}} \sum_{N_{n-gram} \in S} Count_{match}(N_{n-gram})}{\sum_{S \in \{Ref\}} \sum_{N_{n-gram} \in S} Count(N_{n-gram})} \quad (22)$$

where $n - gram$ represents an n-gram, $\{Ref\}$ represents the reference abstract, $Count_{match}(N_{n-gram})$ represents the number of n-grams that appear in the generated abstract and the reference abstract at the same time, and $Count(N_{n-gram})$ indicates the number of n-grams appearing in the reference summary.

ROUGE-L is used to measure the readability of the generated summary, and its calculation is shown in the following equations:

$$R_{lcs} = \frac{LCS(X, Y)}{m} \quad (23)$$

$$P_{lcs} = \frac{LCS(X, Y)}{n} \quad (24)$$

$$ROUGE - L = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2P_{lcs}} \quad (25)$$

where $LCS(X, Y)$ is the length of the longest common subsequence of X and Y, and m and n are the lengths of the reference abstract and the generated abstract. R_{lcs} and P_{lcs} represent the recall rate and precision rate, respectively. Since ROUGE cannot directly evaluate Chinese abstracts, when evaluating Chinese abstracts in this paper, Chinese characters are first converted into numbers before evaluation.

5.3. Experimental Setup

The experiments in this paper are conducted with the PyTorch deep learning framework on a graphics device, NVIDIA GeForce RTX 3090 TI. Training is performed using the ADAGRAD [34] optimizer, with a learning rate set to 0.15. For CNN/DM dataset, we follow the processing method of See et al., and use the non-anonymized version of the data. Word separation is performed using the Stanford University Toolkit core NLP pair, setting

the original text length to 400, the summary length to 100 for training, and 120 for testing, and the preset vocabulary is set to 50 k. For LCSTS dataset, four types of characters are inserted into the document first, including <PAD> as a complementary character, <UNK> as an OOV words, <s> and </s> as sentence start and end identifiers. The vocabulary size is set to 40 k at (CHARACTER-BASED) and 50 k at (WORD-BASED) using the JIEBA word splitting tool. In the coverage mechanism, the weight of coverage loss is set to 1.

In our approach, we use a bidirectional LSTM on the encoder and a unidirectional LSTM on the decoder, with the hidden layer dimension being 256 in both. Additionally, our model has 128 dimensional word embeddings. The batch size is set to 16. We use beam search to get the summaries and the beam size is 4.

5.4. Experimental Results

We conducted four experiments, and the final experimental results were taken from the arithmetic mean of the four experiments, and the standard deviations of the four experimental results are presented in Tables 3–5.

Table 3. ROUGE score of each model on CNN/DM dataset.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3	35.10	14.51	34.38
ABS	31.33	11.81	28.83
PGEN	36.44	15.66	33.42
PGEN + Cov	39.53	17.28	36.38
Key-inf-guide	40.34	17.70	36.57
ACGT(Ours)	40.49 (std = 0.043)	19.72 (std = 0.031)	37.41(std = 0.027)

Table 4. ROUGE score of each model on LCSTS dataset (word-based).

Model	ROUGE-1	ROUGE-2	ROUGE-L
Hu.RNN	17.7	8.5	15.8
Hu.RNN context	26.8	16.1	24.1
Copy Net	35.0	22.3	32
PGEN	36.68	21.39	31.12
PGEN + Cov	37.16	24.67	33.96
ACGT(Ours)	38.35(std = 0.039)	25.70(std = 0.018)	34.81(std = 0.011)

Table 5. ROUGE score of each model on LCSTS dataset (character-based).

Model	ROUGE-1	ROUGE-2	ROUGE-L
Hu.RNN	21.5	8.9	18.6
Hu.RNN context	29.9	17.4	27.2
Copy Net	34.4	21.6	31.3
PGEN	36.57	22.14	31.46
PGEN + Cov	37.15	24.00	34.05
ACGT(Ours)	38.72(std = 0.031)	24.80(std = 0.025)	34.92(std = 0.016)

We chose eight representative state-of-the-art baseline models for comparison, and the pointer generator network with coverage mechanism as our baseline.

Lead-3 [35]: A traditional simple extractive summary model, extracting the first three sentences of the article as the summary.

RNN [33]: RNN is used as the encoder and decoder, and the final hidden layer vector is used as the input of the decoder.

RNN context [32]: Using RNN as the encoder and decoder. The weighted summation of all hidden vectors on the encoding side is used to decode the summary.

ABS [5]: Generates the summary using an attention-mechanism-based encoder–decoder structure, such as RUSH.

Copy Net [15]: A hybrid mechanism is used to obtain information concerning the memory unit and encode the content and location of the text, mainly for solving OOV words.

PGEN [16]: A Seq2Seq + Attention structure with a pointer network that allows copying words from the original text or generating new words from a fixed vocabulary.

PGEN + Cov [6]: Combines a pointer network with an encoder–decoder based on the attention mechanism, and alleviates the problem of generating redundant words with a coverage mechanism.

Key information guide model [36]: Fusing key information of documents, including people, time, and place, in the form of keywords or key sentences, into the generation module using a multi-view attention approach to guide summary generation.

The comparison experiments show that the proposed ACGT outperforms all other baseline methods on both CNN/DM and LCSTS datasets. For CNN/DM dataset, ACGT yields gains of 0.96/2.44/1.03 of ROUGE-1/2/L scores compared to PGEN + Cov. On the LCSTS dataset (word-based), the ROUGE-1/2/L score of ACGT is improved by 1.19/1.03/0.85 compared to PGEN + Cov, and it also improved by 1.57/0.80/0.87 on the LCSTS dataset (character-based). It can be demonstrated that the summary generation method of combining topics proposed by ACGT is effective.

5.5. Ablation Experiment

In the ablation experiment, to further illustrate the influence of the introduction of key topic words in ACGT on abstracts, this paper experimentally analyzes correlation between summaries quality and the number of topic terms. The number of terms extracted from each topic in the CNN/DM dataset and LCSTS dataset is 1–10, and the ROUGE values of ACGT generated abstracts are shown in Figures 5 and 6.

Overall, the number of key topic words has an impact on the quality of the abstract. ROUGE is slightly improved when key topic words are added to the model, and as the number of terms increases, the ROUGE value increases. Experiments show that the performance of both datasets remains stable with the change of the number of key topic words, indicating that ACGT is not sensitive to the number of key topic words.

We believe that although this paper fuses TOP m key topic words in Top n topics, the attention mechanism in our paper can sufficiently suppress the noise caused by the introduction of words, so that the summary performance remains stable with the change of the number of key topic words. For CNN/DM, the number of terms used in this experiment is nine. For the LCSTS dataset, six key themes were chosen.

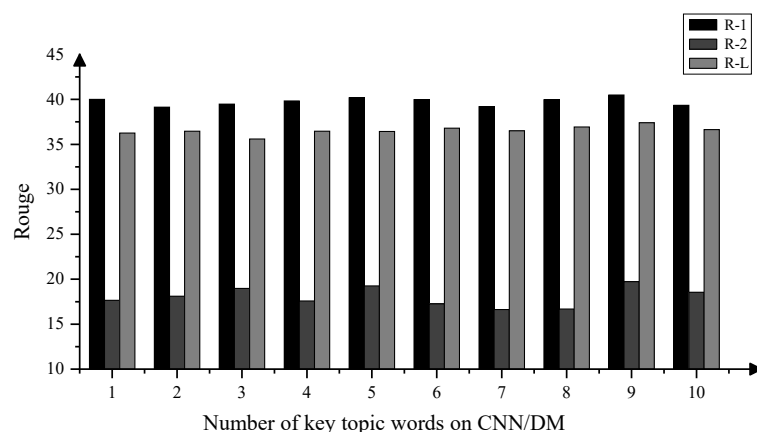


Figure 5. Correlation between summaries quality and the number of topic terms in the CNN/DM dataset (number of topics is 43).

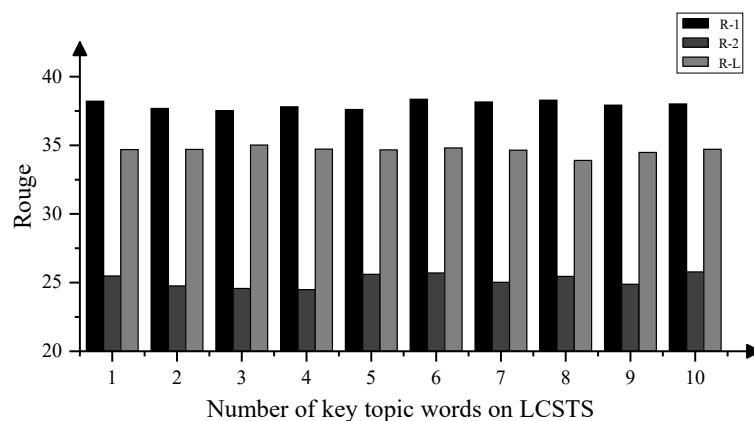


Figure 6. Correlation between summaries quality and the number of topic terms in the LCSTS dataset (number of topics is 18).

6. Analysis and Discussion

6.1. Effect of Number of Topics

To illustrate the effect of introducing topic information into ACGT, we analyze the correlation between the number of topics and summary quality.

When training the LDA, the number of topics in the document set needs to be determined. Blei [23], the proposer of LDA, proposed perplexity as the metric; he suggested that a probability distribution model with low perplexity can predict the original text better. Thus, for CNN/DM dataset, we take the top 10 words with the highest probability for each topic as features and calculate the perplexity value of the LDA model for the number of topics between 1 and 50. The results are shown in Figure 7a. It shows that the average perplexity value of the model is lower when the number of topics is 40~45. Using PGEN + Cov as the baseline method, the ROUGE metrics of ACGT generating summaries when the number of topics is from 1 to 50 are shown in Figure 7b–d. We find that the difference between the ROUGE scores of the ACGT and the baseline method increases significantly as the perplexity value decreases. When the number of topics is 43, the perplexity value is the lowest, at which time the ROUGE difference is the largest, and the performance of ACGT to generate summaries is optimal.

In the LCSTS dataset, the top 10 lexical items with the highest probability in each topic are also chosen as features, and the perplexity is calculated for the number of topics between 1 and 30, and the results are shown in Figure 8a. It shows that the model has the lowest average perplexity when the number of topics is 18 to 20, and overfitting occurs when the number of topics is greater than 20. When the number of topics ranges from 1 to 20, the ROUGE deference values of ACGT generating summaries compared with the baseline methods are shown in Figure 8b–d. It can be demonstrated that the perplexity and the ROUGE difference are negatively correlated. When the number of topics is 18, the ROUGE difference is the best and the performance of ACGT to generate summaries is optimal.

The experiments show that the performance of the ACGT for generating summaries changes significantly with the number of topics introduced. When the perplexity of LDA decreases, the performance of ACGT generating summaries improves, indicating that the topic information extracted by LDA has a positive impact on the performance of summaries, and the ACGT is effective in incorporating topic information into summary generation. The number of topics used by CNN/DM is 43 and LCSTS is 18 in this experiment.

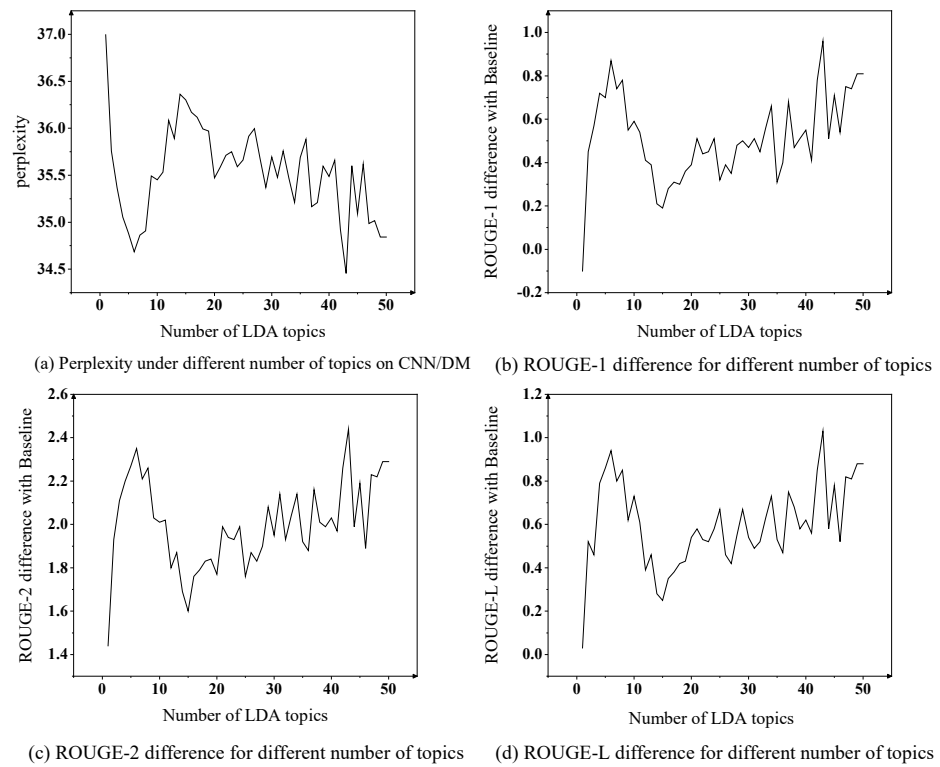


Figure 7. Correlation of summary quality with the number of topics in CNN/DM dataset, where, figure (a) shows the variation of perplexity with the number of topics. In figure (b–d) the vertical axis represents the ROUGE difference between ACGT and the baseline model [4].

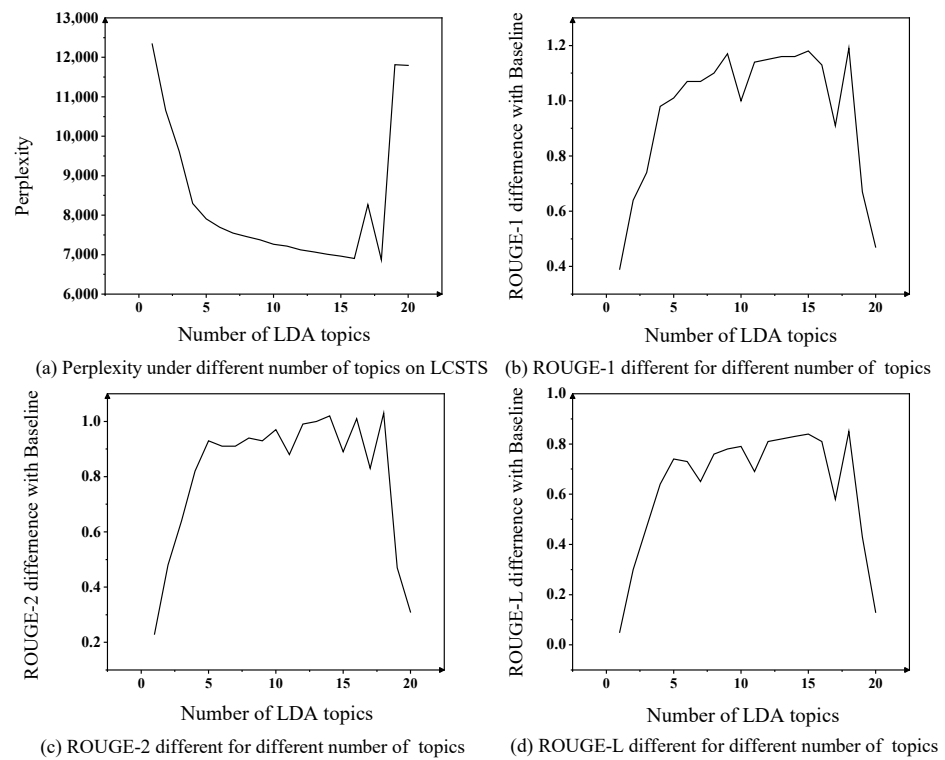


Figure 8. Correlation of summary quality with the number of topics in LCSTS dataset, where, figure (a) shows the variation of perplexity with the number of topics. In figure (b–d), the vertical axis represents the ROUGE difference between ACGT and the baseline model [4].

6.2. Case Studies

In this section, we present the performance of PGEN + Cov and ACGT using examples. The results are shown in Tables 6 and 7, with the gray parts of the table indicating the topic information in each document.

Table 6. Example of summaries generated on CNN/DM dataset.

Original text: A mammoth fire broke out Friday morning in a Kentucky industrial park, sending plumes of thick smoke over the area as authorities worked to contain the damage.
Reference summary: Fire breaks out in a Kentucky industrial park. City official states: no one is believed to be injured or trapped.
PGEN + Cov: A fire broke out Friday morning in an industrial park, and the authorities did not know what had caused the fire.
ACGT(ours): Fire breaks out at the general electric appliance park in Louisville, Kentucky. The authorities said: no one was injured or trapped, but the cause of the fire is unknown.

Table 7. Example of summaries generated on LCSTS dataset.

Original text: At around 15:00 this afternoon, the 2016 International Champions Cup China officially announced that the “2016 International Champions Cup China—Beijing Station—Manchester City vs. Manchester United” originally scheduled to be held at the “Bird’s Nest” at 19:30 on 25 July, Cancelled due to continuous heavy rain in recent days. The Manchester City club said in an official statement that the extreme weather in recent days has made the pitch conditions unsuitable for athletes to compete. Fans can go to the sponsor’s website for a full refund. However, according to Sky Sports UK, the real reason for the cancellation was the poor venue rather than the weather. Just last weekend, Manchester United manager Jose Mourinho called the pitch very bad. On Weibo, some netizens posted the situation of the “Bird’s Nest” lawn, let’s feel it.
Reference summary: Is the Manchester City vs. Manchester United Bird’s Nest match cancelled because of the weather or other reasons?
PGEN + Co: Champions League Manchester City vs. Manchester United reported that the weather caused the venue environment to be too bad?
ACGT(Ours): The real reason for the cancellation of the International Champions Cup China match between Manchester City and Manchester United is that the venue is too bad and the weather is bad?

From Tables 6 and 7, the summaries generated by PGEN + Cov tend to deviate from the topic of the text, while the summaries generated by ACGT are closer to the topic of standard summaries. Additionally, ACGT generates summaries with relatively complete sentences, and its fluency is better than that of the PGEN + Cov method. The case study shows that ACGT incorporates the global topic information of the text by the attention mechanism, which makes the generated abstracts focus on the topic of the original text more effectively and helps to improve the performance of the abstracts.

7. Conclusions

In the task of summary generation, the consistency between the summary and the topics of the original text is crucial. Existing summary generation methods based on the encoder–decoder structure mostly focus on the correlation between the original words and the summary words, while the application of the original topic information is inadequate. Therefore, an abstract summaries method combining global topic information, ACGT, is proposed. By constructing an LDA-based Global Topics Extractor and Global Topics with combined Attention Module, the key topic information of the original text is fused into the document representation for summary generation. Experiments show that ACGT achieves better ROUGE scores than that of baseline methods on both English CNN/DM and Chinese LCSTS datasets, and the performance of the generated summaries is significantly

correlated with the number of topics introduced. The case studies show that ACGT generates summaries with better consistency with the original topics. In the next step, we aim to investigate the document from multiple viewpoints, and extract more key information for document information enhancement.

Author Contributions: Conceptualization, Z.D. and L.L.; methodology, Z.D.; software, Y.W.; validation, J.W. and Y.W.; formal analysis, W.Y.; investigation, J.W.; resources, W.Y.; data curation, W.Y.; writing—review and editing, Z.D.; visualization, Z.D.; supervision, Y.W.; project administration, L.L.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: The National Social Science Foundation of China “Research on the Emotional Semantic Transformation Mechanism of Online Public Opinion Field Based on Deep Learning” [2017CG29]; the Action Plan for High Quality Development of Postgraduate Education of Chongqing University of Technology [gzlxc20223199].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: It can be accessed on <https://github.com/zoeAC22/ACGT.git>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Luhn, H.P. The automatic creation of literature abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165. [\[CrossRef\]](#)
2. Yuan, D.; Wang, L.; Wu, Q.; Meng, F.; Ngan, N.; Xu, L. Language Bias-Driven Self-Knowledge Distillation with Generalization Uncertainty for Reducing Language Bias in Visual Question Answering. *Appl. Sci.* **2022**, *12*, 7588. [\[CrossRef\]](#)
3. Jwa, H.; Oh, D.; Park, K.; Kang, J.; Lim, H. Exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Appl. Sci.* **2019**, *9*, 4062. [\[CrossRef\]](#)
4. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:1409.3215.
5. Rush, A.M.; Chopra, S.; Weston, J. A neural attention model for abstractive sentence summarization. *arXiv* **2015**, arXiv:1509.00685.
6. See, A.; Liu, P.J.; Manning, C.D. Get to the Point: Summarization with Pointer-Generator Networks. *arXiv* **2017**, arXiv:1704.04368.
7. Lin, C.Y.; Hovy, E. Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Edmonton, AB, Canada, 27 May–1 June 2003; pp. 150–157.
8. Abdel-Salam, S.; Rafea, A. Performance Study on Extractive Text Summarization Using BERT Models. *Information* **2022**, *13*, 67. [\[CrossRef\]](#)
9. Lamsiyah, S.; El Mahdaouy, A.; Espinasse, B.; El Alaoui Ouatik, S. An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Syst. Appl.* **2021**, *167*, 114152. [\[CrossRef\]](#)
10. Rani, R.; Lobiyal, D.K. Document vector embedding based extractive text summarization system for Hindi and English text. *Appl. Intell.* **2022**, *52*, 9353–9372. [\[CrossRef\]](#)
11. Nallapati, R.; Zhou, B.; dos Santos, C.N.; Gulcehre, C.; Xiang, B. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *arXiv* **2022**, arXiv:1602.06023.
12. Gehrmann, S.; Deng, Y.; Rush, A.M. Bottom-up abstractive summarization. *arXiv* **2018**, arXiv:1808.10792.
13. Celikyilmaz, A.; Bosselut, A.; He, X.; Choi, Y. Deep communicating agents for abstractive summarization. *arXiv* **2018**, arXiv:1803.10357.
14. Gulcehre, C.; Ahn, S.; Nallapati, R.; Zhou, B.; Bengio, Y. Pointing the Unknown Words. *arXiv* **2022**, arXiv:1603.08148.
15. Gu, J.; Lu, Z.; Li, H.; Li, V.O.K. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. *arXiv* **2016**, arXiv:1603.06393.
16. Vinyals, O.; Fortunato, M.; Jaitly, N. Pointer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. Available online: <https://proceedings.neurips.cc/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf> (accessed on 15 August 2022).
17. Ruan, Q.; Ostendorff, M.; Rehm, G. Histruct+: Improving extractive text summarization with hierarchical structure information. *arXiv* **2022**, arXiv:2203.09629.
18. Mao, Z.; Wu, C.H.; Ni, A.; Zhang, Y.; Zhang, R.; Yu, T.; Deb, B.; Zhu, C.; Awadallah, A.H.; Radev, D. Dyle: Dynamic latent extraction for abstractive long-input summarization. *arXiv* **2021**, arXiv:2110.08168.
19. Li, J.; Shang, J.; McAuley, J. UCTopic: Unsupervised Contrastive Learning for Phrase Representations and Topic Mining. *arXiv* **2022**, arXiv:2202.13469.
20. Bahrainian, S.A.; Feucht, S.; Eickhoff, C. NEWTS: A Corpus for News Topic-Focused Summarization. *arXiv* **2022**, arXiv:2205.15661.
21. Li, M.; Lin, X.X.; Chen, X.; Chang, J.; Zhang, Q.; Wang, F.; Wang, T.; Liu, Z.; Chu, W.; Zhao, D.; et al. Keywords and Instances: A Hierarchical Contrastive Learning Framework Unifying Hybrid Granularities for Text Generation. *arXiv* **2022**, arXiv:2205.13346.

22. Lei, D.M.B.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
23. Wu, D.; Yuan, Z.; Li, X. Automatic Summarization Algorithm Based on the Combined Features of LDA. *Computer Sci. Appl.* **2013**, *3*, 145–148.
24. Liu, N.; Lu, Y.; Tang, X.-J.; Li, M.-X. Multi-document automatic summarization algorithm based on important topics of LDA. *J. Front. Comput. Sci. Technol.* **2015**, *9*, 242–248.
25. Yang, T.; Xie, Q.; Liu, Y.-J.; Liu, P.-F. Topic-aware long text automatic summarization algorithm. *Comput. Eng. Appl.* **2022**, *34*(Part A), 2651–2665.
26. Guo, J.-F.; Fei, Y.-X.; Sun, W.-B.; Xie, P.-P.; Zhang, J. A PGN-GAN Text Summarization Model Fusion Topic. *J. Chin. Comput. Syst.* **2022**, 1–7.
27. Chou, Y.C.; Kuo, C.J.; Chen, T.T.; Horng, G.J.; Pai, M.Y.; Wu, M.E.; Lin, Y.C.; Huang, M.H.; Su, M.Y.; Chen, Y.C.; et al. Deep-learning-based defective bean inspection with GAN-structured automated labeled data augmentation in coffee industry. *Appl. Sci.* **2019**, *9*, 4166. [[CrossRef](#)]
28. Onah, D.F.O.; Pang, E.L.L.; El-Haj, M. A Data-driven Latent Semantic Analysis for Automatic Text Summarization using LDA Topic Modelling. *arXiv* **2022**, arXiv:2207.14687.
29. Rani, R.; Lobiyal, D.K. An extractive text summarization approach using tagged-LDA based topic modeling. *Multimed. Tools Appl.* **2021**, *80*, 3275–3305. [[CrossRef](#)]
30. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
31. Hermann, K.M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching machines to read and comprehend. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
32. Hu, B.; Chen, Q.; Zhu, F. Lcsts: A large scale chinese short text summarization dataset. *arXiv* **2015**, arXiv:1506.05865.
33. Lin, C.Y. Rouge: A Package for Automatic Evaluation of Summaries. Proceedings of the Workshop on Text Summarization of ACL, Barcelona, Spain, 25–26 July 2004; pp. 74–81. Available online: <https://aclanthology.org/W04-1013.pdf> (accessed on 15 August 2022).
34. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
35. Wasson, M. Using Leading Text for News Summaries: Evaluation Results and Implications for Commercial Summarization Applications[C]//COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics. 1998. Available online: <https://aclanthology.org/C98-2217.pdf> (accessed on 15 August 2022).
36. Xu, W.; Li, C.; Lee, M.; Zhang, C. Multi-task learning for abstractive text summarization with key information guide network. *EURASIP J. Adv. Signal Process.* **2020**, *2020*, 16. [[CrossRef](#)]