AN IMPROVED MT5 MODEL FOR CHINESE TEXT SUMMARY GENERATION

Fuping Ren[2], Jian Chen[1] and Defu Zhang[1]

[1]School of informatics, Xiamen University, Xiamen,361005, China
[2]Shenzhen Comtech Technology Co. Ltd, Shenzhen 518063, China

ABSTRACT

Complicated policy texts require a lot of effort to read, so there is a need for intelligent interpretation of Chinese policies. To better solve the Chinese Text Summarization task, this paper utilized the mT5 model as the core framework and initial weights. Additionally, In addition, this paper reduced the model size through parameter clipping, used the Gap Sentence Generation (GSG) method as unsupervised method, and improved the Chinese tokenizer. After training on a meticulously processed 30GB Chinese training corpus, the paper developed the enhanced mT5-GSG model. Then, when fine-tuning the Chinese Policy text, this paper chose the idea of "Dropout Twice", and innovatively combined the probability distribution of the two Dropouts through the Wasserstein distance. Experimental results indicate that the proposed model achieved Rouge-1, Rouge-2, and Rouge-L scores of 56.13%, 45.76%, and 56.41% respectively on the Chinese policy text summarization dataset.

KEYWORDS

Natural Language Processing, Text Summarization, Transformer model

1. INTRODUCTION

The purpose of text summarization is to extract essential information from a given text or set of texts, commonly used for tasks like automatic report generation, news headline creation, and structured search previews. Text summarization methods are broadly categorized into Extractive Summarization and Abstractive Summarization. Abstractive Summarization can make full use of context information to achieve the coherence of summarization and conform to the thinking form of human natural language, but designing a good Abstractive Summarization method exists certain challenges.

Early Abstractive Summarization method was largely impractical. The seq2seq framework[1] was introduced in 2014 and garnered attention; however, this framework was plagued by some problems such as generating inaccurate and duplicate information. To solve these concerns, the Pointer-Generator Network (PGN)[2] proposed a hybrid pointer generation network to address word duplication and out-of-vocabulary words. Additionally, it employed a coverage mechanism to prevent the duplication of information. It is worth noting that most preceding text summarization models were based on RNN networks, leading to difficulties in parallelization.

The Transformer model[3] was proposed in 2017 and it marked a significant milestone in the field of text summarization. However, conventional Transformer models did not exhibit dominance, paving the way for large-scale models to dominate both Extractive and Abstractive Summarization. The MASS model[4], introduced in 2019, addressed the limitations of the BERT model for generative tasks, proposing the use of continuous segments as masking objects and employing an entire Encoder-Decoder structure. Similarly, the BART model[5], also introduced in

2019, utilized an arbitrary noise function to perturb and reconstruct text within the Encoder-Decoder framework, making it more suitable for text summarization than previous methods. our model uses an abstractive method based on PEGASUS with a copy mechanism to generate the final summary from the bridging document. SUMOPE is proposed for long text summary generation, computational results show that SUMOPE outperforms the state-of-the-art methods in terms of ROUGE scores and human evaluation.

Although most of the aforementioned models were designed for English text, their application to Chinese text summarization remains a challenge due to limited research and model availability. Large-scale models such as T5[8] and its multilingual variant mT5[9] have shown promising results for Chinese text summarization, albeit with time efficiency limitations. Google's PEGASUS model[10], proposed in 2020, specifically focused on sentence masking as an unsupervised task within an Encoder-Decoder framework, demonstrating excellent performance particularly on small sample datasets. However, PEGASUS faces some limitations regarding parameter scale and representation ability, particularly in the context of Chinese summarization.

The contributions of this paper are as follows:
(1) Based on GSG, an enhanced mT5 model is proposed for Chinese Text Summary Generation, the proposed model shows superior performance compared to other models.
(2) An improvement to the Dropout mechanism is developed, resulting in enhanced performance through the execution of Dropout twice.
(3) The proposed model is applied to Chinese policy text summarization and makes promising results.

2. AN IMPROVED PRE-TRAINING MODEL MT5 BASED ON GSG AND MLM

This paper utilized the mT5 model as the foundational framework and initial weight, subsequently employing the GSG method for an unsupervised task. The mT5 model represents an enhanced multilingual version derived from the T5 model. The original T5 model was trained only on English text data, making it less suitable for use with other languages. T5 is a general model designed primarily for all text-based NLP tasks, utilizing a unified "seq2seq" format to effectively fine-tune any downstream task using the same set of hyperparameters. One of the most important aspects of T5 is guiding decision-making in various stages of pre-training, providing considerable reference value for practical applications. What makes T5 particularly noteworthy is its scale, with the size of its pre-trained model ranging from 60 million to 11 billion parameters. These models have been pre-trained on approximately 1 trillion word tokens. Unlabelled data is sourced from the C4 dataset, which comprises roughly 750GB of English text obtained from the Common Crawl website. This paper made adjustments to the foundational layer of the framework, pruned a portion of parameters to reduce the model size, and enhanced the Chinese tokenizer. Ultimately, approximately 30G of the Chinese training corpus was utilized for training, leading to the development of the mT5 pre-training model using the GSG method.

2.1. Framework of the Proposed Model

The mT5 model is a versatile framework that utilizes a unified "seq2seq" format to tackle a wide range of text-based NLP problems. Before pre-training, it's crucial to carefully assess the overall architecture. Built upon the Transformer model, mT5 incorporates several transformer architectures, such as Encoder-Decoder, Language Model, and prefix-based language models (Prefix LM). Through experimental comparison, it was determined that the Encoder-Decoder model delivered the most effective overall performance among these three frameworks. Therefore, the mT5 model adopts the Encoder-Decoder framework in this paper.

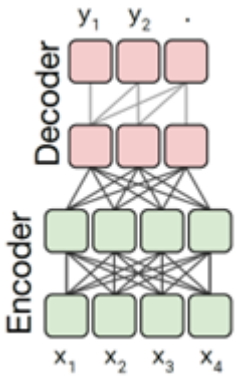

Figure 2.1 Encoder-Decoder architectures

With the overall framework in place, the next step is to make policy choices for the training process. There are various strategies available to guide the model, one such example being the Masked Language Model (MLM) method. For pre-training, the T5 model found that the BERT-like pre-training method was the most effective. This method primarily involves randomly destroying part of the content and then restoring it. When selecting the text masking strategy, the T5 model determined that the "Replace spans" method, which considers entire words, is the optimal choice. Typically, the recommended text masking ratio in the MLM model is 15%, and it was observed that a subsection replacement length of 3 proved to be the most effective.

Taking into account these practical considerations, this paper ultimately adopts the Gap Sentence Generation (GSG) method. By leveraging the advantages and flexibility of the mT5 model, the model's basic framework and weights can be directly utilized.

2.2. Perfect Tokenizer

In NLP tasks, the input is a piece of text, but the actual calculations primarily use word embeddings, requiring an intermediate conversion process. In English, the Tokenizer's role is to segment a piece of text into a list of words and then assign each word a word vector, ultimately forming the word embedding table used in calculations. However, in Chinese, more attention is given to lexical information. For example, the text “我爱祖国” (I love my country) would first be segmented into “我 爱 祖 国 祖国” (I love ancestor country), resulting in four individual characters and one word. These five tokens are then converted into their corresponding word vectors, forming the word embedding table. The term Tokenizer generally translates to "分词工具" (word segmentation tool), and the commonly used word segmentation tool for Chinese is Jieba.

Before training, this paper needs to improve the word segmentation (tokenization). Both T5 and mT5 models use the sentencepiece[11] Tokenizer, which is a C++-written word segmentation library known for its efficiency and lightweight nature. Unfortunately, it is not particularly friendly for Chinese. Firstly, sentencepiece forcibly converts some full-width symbols to half-width symbols, which may be unacceptable in certain cases and could potentially affect the evaluation results of tasks. Secondly, although sentencepiece’s built-in algorithm is capable of tokenizing Chinese words, it is still not sophisticated enough for Chinese word segmentation. Lastly, being written in C++, despite being open-source, for those familiar with Python, working with C++ can feel like dealing with a black box, making it difficult to read the source code or make modifications.

Based on these issues, it was decided to switch the Tokenizer to BERT's Tokenizer. However, simply replacing the Chinese BERT Tokenizer is not sufficient. First, as mentioned earlier, enhanced lexical information can lead to better performance for Chinese natural language processing models. Second, even when considering individual characters, the vocab.txt of Chinese BERT is incomplete, omitting some common punctuation marks (such as quotation marks) and Chinese characters (such as “琊” and others). Therefore, it is necessary to further improve the vocab.txt. Specifically, the plan is to first add the first 200,000 words from the original Chinese BERT token dictionary obtained using Jieba word segmentation. Then, modify the logic of the Tokenizer to enable word segmentation, a relatively straightforward operation that only requires consideration of the Jieba word segmentation table during tokenization. Subsequently, this modified Tokenizer will be used to traverse the segmented training corpus, counting the frequency of each token. Finally, only the top 100,000 most frequent words and 50,000 characters will be retained and added to vocab.txt, thus constructing the final Tokenizer.

2.3. GSG and MLM Method

This paper selected the GSG method that is suitable for text summarization tasks. Given three sentences, the middle sentence is masked and then restored by this method. However, this method does not utilize all of the text content. Only using the GSG method is equivalent to utilizing only one-third of the text content. The method is shown in Figure 2.2. When using the GSG method in this paper, the MLM method is also added to make use of the remaining text content. There are a total of 3 sentences in Figure 2.2, the middle one "我住在厦门" is masked as Gap, marked as "[MASK1]". The surrounding text randomly selects words as the masking objects. In the Figure 2.2, the words "祖国" and "求学" were randomly selected and marked as "[MASK2]", and the proportion was still 15%. The masked middle text from the GSG method needs to be input as target text into the decoder for text restoration. The words masked in the surrounding context are done in a BERT-like manner, allowing for text restoration operations within the encoder section.

The GSG method is proposed based on the assumption that models closely aligned with downstream tasks can achieve superior performance. It has shown strong performance in text summarization tasks[9]. Indeed, the GSG method is a type of masking technique, wherein sentences are masked. This aligns with the random substitution strategy employed by the T5 model, with the primary difference being the expansion of the small paragraph length into a full sentence. GSG offers three masking strategy options based on a given document $D = \{x_i\}_n$, where n represents the number of sentences and each sentence is denoted as x_i. The three strategies are as follows [9].

(1) Random: Randomly select m sentences as Gap Sentences.

(2) Lead: Select the previous m sentences as Gap Sentences.

(3) Principal: Select the previous m sentences as Gap Sentences according to the level of importance.

Among three strategies, Principal stands out as a relatively reasonable choice and is therefore adopted in this paper. Two methods for assessing the importance of sentences are as follows:

(1) Independent discrimination (Ind): The ROUGE1-F1 score is independently calculated for each sentence as an importance score for sorting, utilizing the calculation expression shown in formula (2.1).

$$s_i = rouge(x_i, D \backslash \{x_i\}), \forall_i \quad (2.1)$$

where, s_i represents the score of the i-th sentence, and the formula represents the relationship between the current sentence and the remaining text.

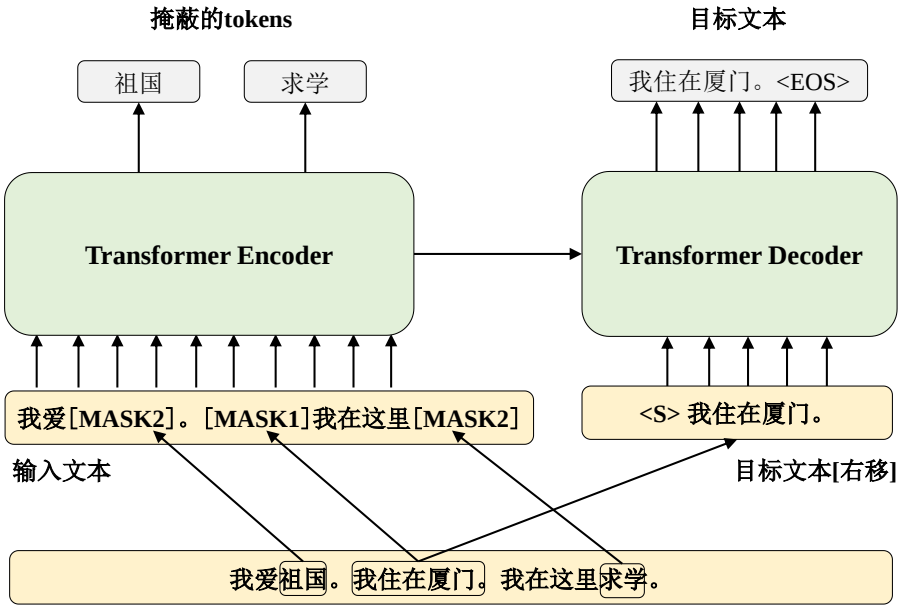

Figure 2.2 Schematic diagram of the GSG method

(2) Sequential discrimination (Seq): This method involves selecting the ROUGE1-F1 score of $S \cup \{x_i\}$ and the remaining text $D \backslash (S \cup \{x_i\})$ through greedy strategy, until m sentences are selected. The process is shown in Algorithm 2.1.

When computing the ROUGE1-F1 score, n-grams are classified into two types: Non-repetitive n-gram set (Uniq) and Repeated n-gram set (Orig). The Non-repetitive n-gram set (Uniq) processes the sentence set first, removing any repeated n-grams, and then utilizes ROUGE1-F1 for calculation. Meanwhile, the Repeated n-gram set (Orig) maintains the original sentences and allows for n-gram repetition. This paper explored six combinations of principal method and n-gram, specifically Ind-Uniq, Ind-Orig, Seq-Uniq, Seq-Orig, Random, and Lead. In this paper, the Ind-Orig combination was chosen. Furthermore, the selection of gapped sentences is proportional and referred to as Gap Ratio, with the most effective ratio identified as 30%.

Algorithm 2.1 Sequential Discrimination Algorithm

Algorithm 2.1 Selection of Gap Sentences for Sequential Discrimination

1: $S := \emptyset$

2: **for** $j \leftarrow 1$ to m **do**

3: $s_i := rouge\left(S \cup \{x_i\}, D \backslash (S \cup \{x_i\})\right), \quad \forall_i \ s.t. \ \ x_i \notin S$

4: $k := argmax_i \{s_i\}_n$

5: $S := S \cup \{x_k\}$

6: **end for**

In conclusion, this paper selected the mT5 model as the initial weight and fundamental framework, adhering to the standard Encoder-Decoder structure. Additionally, a BERT-like

method was employed for pre-training. The masking strategy involved the use of a small segment mask (Replace spans), while the GSG method differed slightly by masking sentences. When considering the issue of masking ratio, a 30% Gap Ratio for the GSG method produced the most favorable outcomes. As it pertains to sentence masking, the length of the span was no longer taken into account. For simplicity, the proposed model is subsequently denoted as mT5-GSG.

2.4. Training Corpus and Parameters

In this paper, when referring to literature, we searched through various channels and compiled approximately 30GB of meticulously processed Chinese corpora. The main sources include CSL dataset, LCSTS dataset, Weibo dataset, NLPCC2017 dataset, Sogou dataset, and the majority of the data comes from Chinese texts on the Common Crawl website. As it is a pre-training corpus, it only contains text content without any labels. Pre-training is essentially unsupervised training.

In terms of parameter settings, practical considerations need to be taken into account. The training machine is equipped with 12 Nvidia RTX 3090 graphics cards, which is not an extensive amount of resources, so the parameters need to be compact. Among them, the number of layers for the Encoder and Decoder is set at L=12, which represents the depth of the Transformer. The hidden layer size is H=768, the feed-forward neural network layer size is F=1024, and the number of attention heads in the self-attention layer is A=12. The total number of training steps is 500K, the pre-training learning rate is 0.02, the pre-training batch size is 256, the combination of Gumbel-Softmax Gradient (GSG) method is set as Ind-Orig. The corpus consists of approximately 30GB of text, the maximum length of input tokens is 512, and the maximum length of target text tokens in the GSG method is 256. The dropout ratio during pre-training is 0.5, and the activation function chosen is the GELU function, which considers both randomness and sufficiency. As for the optimizer, AdamW optimizer is chosen to ensure sufficient training of the model.

Additionally, it is important to note that the initial weights of mT5 have different dimensions. Therefore, in this paper, dimension reduction was performed by pruning. For example, while the dimension size of the feed-forward neural network layer (FFN) in mT5 is 4096, this paper reduces it to 1024, and other parameters are subsequently reduced to the dimensions mentioned above. Although deleting some parameters may reduce effectiveness, it is entirely acceptable according to the needs of this paper. If the model were trained according to the dimensions of mT5, and setting aside the issue of computational resources, the resulting model size alone would be impractical (approximately 1.5GB), whereas the final size of the previous Named Entity Recognition (NER) task model is only about 500MB. The model trained in this paper for the text summarization task is only around 370MB in size.

2.5. Improved Dropout

In 2021, a straightforward improvement to Dropout, termed "Dropout Twice," was introduced in SimCSE[12]. This modification involves the execution of Dropout twice to enhance its effectiveness. The underlying rationale for this approach stems from addressing the inconsistency issue between the training and inference stages, which arises from the inherent randomness introduced by Dropout. To implement this improvement, consider a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, the purpose of training is to obtain a model $P^w(y|x)$, which n represents the number of training samples, (x_i, y_i) represents a labelled sample pair, x_i represents the input data, and y_i is the label. Using "Dropout Twice" yields two distribution models P_1 and P_2, which can be combined using similarity metrics such as KL divergence [13], JS divergence [14], and Wasserstein distance adopted in this paper.

The Wasserstein distance, also known as earthmover's distance, measures the dissimilarity between two probability distributions and is given by the formula (2.2).

$$D_{ws}(P_1, P_2) = inf_{\theta \sim \Pi(P_1, P_2)} \mathbb{E}_{(x,y)\sim\theta}[||x - y||] \quad (2.2)$$

where, inf denotes the largest lower bound, $\Pi(P_1, P_2)$ is a set of all possible joint distributions combining with the P_1 and P_2 distribution, and $\mathbb{E}_{(x,y)\sim\theta}[||x - y||]$ calculates the distance between two samples *x* and *y* sampled from the joint distribution θ. Therefore, the expectation of this sample pair distances under the joint distribution θ can be calculated. and the lowest attainable bound on this expectation across all possible joint distributions is the Wasserstein distance.

Based on the above, this paper applied the Wasserstein distance to integrate the distributions derived from the two Dropouts. Prior to this, it is crucial to emphasize that the principal objective of model training is to minimize the negative log-likelihood loss function, as denoted in formula (2.3).

$$L_{nll} = \frac{1}{n}\sum_{i=1}^{n} -logP^{w}(y_i|x_i) \quad (2.3)$$

For "Dropout Twice", the sample x_i is repeatedly input into the feedforward neural network, and obtains two distributions, denoted as $P_1^w(y_i|x_i)$ and $P_2^w(y_i|x_i)$. For the same input (x_i, y_i), two unequal probability distributions are obtained. After two Dropouts, the negative log-likelihood function is shown in formula (2.4).

$$L_{nll}^{i} = -logP_1^w(y_i|x_i) - logP_2^w(y_i|x_i) \quad (2.4)$$

By considering the Wasserstein distance between the two Dropout distributions, we derive the formula (2.5).

$$L_{ws}^{i} = D_{ws}\left(P_1^w(y_i|x_i), P_2^w(y_i|x_i)\right) \quad (2.5)$$

Following the computation of the aforementioned formulas (2.4) and (2.5), values are obtained using the negative log-likelihood function and the Wasserstein distance. To mitigate the influence of the Dropout module, an enhancement to the previous loss function is analogized by introducing influencing factors for adjustment. The final model incorporates the Wasserstein distance, as illustrated in formula (2.6).

$$\begin{aligned} L^i &= L_{nll}^{i} + \beta L_{ws}^{i} \\ &= -logP_1^w(y_i|x_i) - logP_2^w(y_i|x_i) + \beta D_{ws}\left(P_1^w(y_i|x_i), P_2^w(y_i|x_i)\right) \end{aligned} \quad (2.6)$$

Among them, β is the impact factor or weight balancing coefficient, and its value is set to 5 after experimentation. Similar to the impact factor of the previous activation function, it needs to be tested by modifying parameters multiple times. To simplify the calculation, its value range is set to positive integers. The reason for this is that more attention needs to be paid to the influence of the two dropouts at this point. If only the loss function itself is considered, then the value should be set between 0 and 1.

3. EXPERIMENTAL ANALYSIS

3.1. Evaluation Indicators & Data

The most widely used evaluation method in the text summarization domain is the ROUGE[15] evaluation metric, which commonly includes $ROUGE-N$ and $ROUGE-L$. These metrics can be computed as follows:

$$ROUGE-N=\frac{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\}}\sum_{gram_n\in S}Count(gram_n)} \quad (3.1)$$

where n denotes $n-gram$, $Count(gram_n)$ denotes the number of occurrences of one $n-gram$, and $Count_{match}(gram_n)$ denotes the number of co-occurrences of one $n-gram$. Usually, the *N* values commonly range from 1 to 4, and for this paper, 1 and 2 are selected.

$$ROUGE-L=\frac{(1+\beta^2)Rec_{lcs}Pre_{lcs}}{Rec_{lcs}+\beta^2 Pre_{lcs}} \quad (3.2)$$

$$Rec_{lcs}=\frac{LCS(X,Y)}{m} \quad (3.3)$$

$$Pre_{lcs}=\frac{LCS(X,Y)}{n} \quad (3.4)$$

Where X denotes the candidate abstract, Y represents the reference abstract, $LCS(X,Y)$ represents the length of the longest common subsequence of X and Y. and *m* and *n* represent the lengths of *Y* and *X* respectively. Additionally, Rec_{lcs} represents the recall rate, and Pre_{lcs} represents the precision rate. β is an influence factor typically set to a large value.

Furthermore, in text summarization or text generation tasks, the decoder module usually employs a search algorithm during decoding. Commonly used methods include Greedy search and Beam Search [16], with this paper utilizing Beam Search.

Table 3.1 shows the experimental parameter settings during fine-tuning.

Table 3.1 Experimental parameter settings during fine-tuning

Parameter	Value
BERT hidden layer dimension	768
Learning rate when fine-tuning mT5-GSG	1e-5
Batch Size during mT5-GSG training	16
EPOCH	100
STEPS	500K
Optimizer	AdamW

Regarding datasets, this paper considers public Chinese text abstract datasets, including CSL, and NLPCC2017. In particular, a Chinese policy text abstract from a practical project is considered. The specific sample size is shown in Table 3.2 below.

Table 3.2 The size of the dataset samples (unit: pieces)

Data set	Train sample	Dev sample	Test sample
CSL	50000	500	200
NLPCC2017	50000	800	200
Project dataset	8000	100	50

3.2. Effects of Mt5-GSG

In this section, all models do not improve Dropout. Additionally, the proposed mT5-GSG model adopts the Ind-Orig strategy with a Gap Ratio of 30%.

(1) The effect of different models on the CSL dataset

Table 3.3 shows the effect of different models on the CSL dataset. BERT-PGN [17], mT5, and PEGASUS [10] models were selected for comparison because they belong to the state-of-the-art models for Chinese text summary generation. Notably, the beam size significantly influences the performance of models.

The proposed mT5-GSG obtained the best results when the beam size is set to 3. The Rouge-1, Rouge-2 and Rouge-L scores are 70.45%, 60.57% and 68.26 %, respectively. Compared with the mT5 model, the Rouge-1, Rouge-2 and Rouge-L scores of mT5-GSG model are improved by 1.64%, 1.90% and 2.43% respectively.

Table 3.3 Comparison results of the models on the CSL dataset (unit: %)

Model	Beam Size	Rouge -1	Rouge -2	Rouge- L
BERT-PGN (Multidimensional Semantic Features)	2	42.70	16.64	38.44
PEGASUS	2	65.45	54.91	63.81
mT5	2	68.22	57.83	66.38
mT5-GSG	2	69.00	58.74	66.96
BERT-PGN (Multidimensional Semantic Features)	3	44.01	25.73	43.79
PEGASUS	3	66.34	56.06	64.75
mT5	3	68.81	58.67	66.83
mT5-GSG	**3**	**70.45**	**60.57**	**68.26**
BERT-PGN (Multidimensional Semantic Features)	4	43.87	17.50	38.97
PEGASUS	4	66.09	55.75	64.44
mT5	4	68.68	58.50	66.65
mT5-GSG	4	69.19	59.10	67.25

(2) The effect of different models on the LCSTS dataset

Table 3.4 shows the experimental results of different models on the LCSTS dataset. The size of the LCSTS dataset is larger and the samples are more complex, including long texts and summaries. Notably, when utilizing a beam size of 4, mT5-GSG emerged as the top-performing model, achieving Rouge-1, Rouge-2, and Rouge-L scores of 34.12%, 22.23%, and 31.78% respectively.

Table 3.4 Comparison results of the models on the LCSTS dataset (unit: %)

Model	Beam Size	Rouge -1	Rouge -2	Rouge- L
BERT-PGN (Multidimensional Semantic Features)	2	29.57	18.04	27.99
PEGASUS	2	32.90	21.13	31.21
mT5	2	30.75	19.54	28.92
mT5-GSG	2	33.53	21.54	31.47
BERT-PGN (Multidimensional Semantic Features)	3	30.70	19.17	29.20
PEGASUS	3	33.35	21.55	31. 41
mT5	3	31.67	20.40	29.96
mT5-GSG	3	34.00	21.98	31.51
BERT-PGN (Multidimensional Semantic Features)	4	30.95	19.50	29.45
PEGASUS	4	33.72	21.81	31.49
mT5	4	31.97	20.72	30.15
mT5-GSG	4	**34.12**	**22.23**	**31. 57**

(3) The effect of different models on the NLPCC2017 dataset

Table 3.5 shows the experimental results of the models on the NLPCC2017 dataset. For mT5-GSG, the best performance was attained when the beam size is set to 3, resulting in Rouge-1, Rouge-2, and Rouge-L scores of 48.89%, 35.63%, and 43.04% respectively.

Table 3.5 Comparison results of the models on the NLPCC2017 dataset (unit: %)

Model	Beam Size	Rouge-1	Rouge-2	Rouge-L
BERT-PGN (Multidimensional Semantic Features)	2	41.12	23.55	34.46
PEGASUS	2	47.21	24.56	39.25
mT5	2	47.52	33.51	41.33
mT5 -GSG	2	48.67	33.39	42.07
BERT-PGN (Multidimensional Semantic Features)	3	42.28	23.89	35.63
PEGASUS	3	47.74	25.59	40.82
mT5	3	47.94	34.55	42.73
mT5-GSG	3	**48.89**	**35.63**	**43.04**
BERT-PGN (Multidimensional Semantic Features)	4	41.86	23.62	34.58
PEGASUS	4	47.68	25.27	40.54
mT5	4	47.83	34.47	42.49
mT5-GSG	4	48.78	34.90	42.91

(4) The effect of different models on the Chinese policy text summary dataset

Similarly, Table 3.6 reports the effect comparison of the models on the Chinese policy text summary dataset. Once again, mT5-GSG excelled notably when employing a beam size of 3, achieving Rouge-1, Rouge-2, and Rouge-L scores of 54.63%, 44.18%, and 55.24% respectively.

Table 3.6 Comparison results of the models on the Chinese policy text summary dataset (unit: %)

Model	Beam Size	Rouge -1	Rouge -2	Rouge- L
BERT-PGN (Multidimensional Semantic Features)	2	35.98	17.76	33.63
PEGASUS	2	50.77	35.59	50.95
mT5	2	48.25	21.35	36.69
mT5-GSG	2	53.01	28.27	54.91
BERT-PGN (Multidimensional Semantic Features)	3	36.15	17.54	33.63
PEGASUS	3	52.27	37.98	53.44
mT5	3	50.27	20.15	50.57
mT5-GSG	3	**54.63**	**44.18**	**55.24**
BERT-PGN (Multidimensional Semantic Features)	4	35.47	17.27	33.52
PEGASUS	4	51.91	37.09	50.38
mT5	4	50.03	26.23	49.52
mT5-GSG	4	53.74	41.40	54.85

3.3. Impact Of GSG Strategy And Gap Ratio

Table 3.7 shows the influence of the GSG strategy on the model mT5-GSG for public datasets CSL and LCSTS. The Beam Size of CSL data set is 3, and that of LCSTS data set is 4.

Table 3.7 The influence of GSG's strategy on the mT5-GSG model when used CSL and LCSTS datasets (unit: %)

Strategy	CSL (Beam Size=3)			LCSTS (Beam Size=4)		
	Rouge-1	Rouge-2	Rouge-L	Rouge-1	Rouge-2	Rouge-L
Random	70.37	60.41	65.70	33.92	20.94	31.44
Lead	69.88	60.01	65.46	33.79	20.78	31.35
Ind-Orig	**70.45**	**60.57**	**68.26**	**34.12**	**22.23**	**31.78**
Ind-Uniq	70.41	60.54	68.23	34.09	22.15	31.55
Seq-Orig	70.33	60.31	68.05	34.10	22.07	31.43
Seq-Uniq	70.40	60.24	68.03	34.01	20.53	30.13

As shown in Table 3.7, the Ind-Orig strategy adopted in this paper is the best in all strategies on the two public datasets, which proves the advantage of the Ind-Orig strategy in the GSG method.

Table 3.8 shows the influence of Gap Ratio of the GSG method on the CSL and LCSTS datasets. It was observed that a Gap Ratio of 30% yielded the most favorable results numerically.

Table 3.8 The influence of Gap Ratio in the GSG method on the mT5-GSG model when used CSL and LCSTS datasets (unit: %)

Gap Ratio	CSL (Beam Size=3)			LCSTS (Beam Size=4)		
	Rouge -1	Rouge -2	Rouge- L	Rouge -1	Rouge-2	Rouge-L
5%	70.33	60.27	67.83	34.06	21.91	31.57
10%	70.23	60.17	68.13	34.08	22.08	31.74
15%	**70.48**	60.47	67.97	34.08	22.10	31.73
30%	70.45	**60.57**	**68.26**	**34.12**	**22.23**	**31.78**
45%	70.36	60.49	**68.26**	34.00	22.08	31.63
60%	70.15	60.12	68.05	33.97	21.91	31.34
75%	68.95	59.99	67.93	33.57	21.30	31.36

3.4. Improved Dropout

To further enhance its application, this paper has made improvements to the Dropout method. Table 3.9 illustrates the enhanced effects. The mT5-GSG model itself does not improve Dropout. The improvement of Dropout in this paper is carried out in the fine-tuning stage.

Table 3.9 The effect of improved Dropout on the Chinese policy text summary dataset (unit: %)

Model	Beam Size	Rouge-1	Rouge-2	Rouge-L
mT5-GSG	2	53.01	28.27	54.91
mT5-GSG $_{(improved)}$	2	54.75	38.50	55.02
mT5-GSG	3	54.63	44.18	55.24
mT5-GSG $_{(improved)}$	3	**56.13**	**45.76**	**56.41**
mT5-GSG	4	53.74	41.40	54.85
mT5-GSG $_{(improved)}$	4	54.07	42.32	55.19

In Table 3.9, the improved model demonstrates its most effective performance when utilizing a beam size of 3. The Rouge-1 score reached 56.13%, indicating a relative increase of 1.50%. Moreover, the Rouge-2 score reached 45.76%, with a relative increase of 1.68%, and the Rouge-score of 56.41% showed a relative increase of 1.17%. These results highlight the model's enhanced effectiveness through the application of "Dropout twice". Furthermore, efforts were made to enhance the impact of Dropout on public datasets. As depicted in Table 3.10, the performance metrics of the improved mT5-GSG model surpassed those of the original model, solidifying the benefits of the enhanced Dropout approach.

Table 3.10 The effect of improved Dropout on the public dataset (unit: %)

Model	CSL (Beam Size=3)	LCSTS (Beam Size=4)	NLPCC2017 (Beam Size=3)
	R1/R2/RL	R1/R2/RL	R1/R2/RL
mT5-GSG	70.45/60.57/68.26	**34.12**/22.23/31.78	48.89/35.63/43.04
mT5-GSG (improved)	**71.29/60.68/68.92**	**34.12/22.41/31.80**	**49.20/36.16/43.80**

4. CONCLUSIONS

This paper introduces a specialized pre-training model mT5-GSG, which utilizes the Gap Sentence Generation (GSG) approach for unsupervised training by integrating the framework and initial weights of mT5. Subsequently, model cropping is employed to reduce the model size, followed by pre-training on a Chinese corpus of approximately 30GB. Ultimately, an mT5-GSG pre-training model with about 370 million parameters is obtained, effectively resolving the challenges encountered by other models. To further enhance the model's performance, this paper proposes the "Dropout Twice" concept, which innovatively combines the probability distributions of two Dropouts using the Wasserstein distance method. The computational results demonstrate that this model outperforms existing models, particularly exhibiting optimal performance on Chinese policy text datasets. The Rouge-1, Rouge-2, and Rouge-L scores are 56.13%, 45.76%, and 56.41% respectively, satisfying the requirements of practical applications.

ACKNOWLEDGEMENTS

This work was jointly supported by the grant from the National Natural Science Foundation of China (61672439) and by the project grant from the Coding (Xiamen) big data science company.

REFERENCES

[1] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 2014, 27.
[2] See A, Liu P J, Manning C D. Get To The Point: Summarization with Pointer-Generator Networks. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 1073-1083.
[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Proceedings of the 31th International Conference on Neural Information Processing System. 2017: 6000-6010.
[4] Song K, Tan X, Qin T, et al. MASS: Masked Sequence to Sequence Pre-training for Language Generation//International Conference on Machine Learning. PMLR, 2019: 5926-5936.
[5] Lewis M, Liu Y, Goyal N, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7871-7880.
[6] Zhao Y, Huang S, Zhou D, et al. CNsum: Automatic Summarization for Chinese News Text. Lecture Notes in Computer Science, 2022, 13472:539-547.
[7] Chang C, Zhou J, Zeng X, Tang Y. SUMOPE: Enhanced Hierarchical Summarization Model for Long Texts. Lecture Notes in Computer Science, 2023,14177:307-319.
[8] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 2020, 21: 1-67.
[9] Xue L, Constant N, Roberts A, et al. mT5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934, 2020.

[10] Zhang J, Zhao Y, Saleh M, et al. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. International Conference on Machine Learning. PMLR, 2020: 11328-11339.
[11] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2018: 66-71.
[12] Gao T, Yao X, Chen D. Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821, 2021.
[13] Barz B, Rodner E, Garcia Y G, et al. Detecting regions of maximal divergence for spatio-temporal anomaly detection. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(5): 1088-1101.
[14] Sra S. Metrics induced by Jensen-Shannon and related divergences on positive definite matrices. Linear Algebra and its Applications, 2021, 616: 125-138.
[15] Lin C Y. Rouge: A package for automatic evaluation of summaries. Text summarization branches out. 2004: 74-81.
[16] Zhao T, Ge Z, Hu H, et al. Generating Natural Language Adversarial Examples through An Improved Beam Search Algorithm. arXiv preprint arXiv:2110.08036, 2021.
[17] Jinyuan Tan, Yufeng Diao, Ruihua Qi, Hongfei Lin. Chinese News Text Automatic Abstract Generation Based on BERT-PGN Model. Journal of Computer Applications, 2021, 41(01):127-132.