Length Control in Abstractive Summarization by Pretraining Information Selection

Yizhu Liu[1] **Qi Jia**[2] **Kenny Q. Zhu**[3]*
Shanghai Jiao Tong University
Shanghai, China
{[1]liuyizhu, [2]Jia_qi}@sjtu.edu.cn, [3]kzhu@cs.sjtu.edu.cn

Abstract

Previous length-controllable summarization models mostly control lengths at the decoding stage, whereas the encoding or the selection of information from the source document is not sensitive to the designed length. They also tend to generate summaries as long as those in the training data. In this paper, we propose a length-aware attention mechanism (LAAM) to adapt the encoding of the source based on the desired length. Our approach works by training LAAM on a summary length balanced dataset built from the original training data, and then fine-tuning as usual. Results show that this approach is effective in generating high-quality summaries with desired lengths and even those short lengths never seen in the original training set.

1 Introduction

Abstractive summarization (Nallapati et al., 2016; See et al., 2017; Çelikyilmaz et al., 2018; Dong et al., 2019; Lewis et al., 2020; Liu et al., 2021; Dou et al., 2021) aims at reproducing the semantics and topics of the original text in a concise and fluent summary by paraphrasing. In order to display the summary on different mobile devices or websites with space limitations, we have to produce summaries in different lengths. Length-controllable summarization is a multi-objective optimization problem, including generating complete summaries within desired lengths and selecting proper information to summarize based on desired lengths. The existing length-controllable summarization based on encoder-decoder models can be divided into two categories: (1) *early-stop during decoding* and (2) *information selection before encoding*.

Early-stop during decoding methods (Kikuchi et al., 2016; Liu et al., 2018; Makino et al., 2019; Yu et al., 2021) focus on when to output *eos* (end of sequence), indicating the end of the summary. An ad-hoc method (Rush et al., 2015) generates the *eos* by assigning a score of $-\infty$ to all candidate words at the position of the desired length during test. Ad-hoc can be applied to any seq2seq model. Others learn the relationship between length and the decoder state at training time. However, these methods simply add length requirements to the decoder and ignore the fact that encoding the content, or the information selection, from the source document must also adapt to different length requirements. Table 1 gives an example. The content of the reference summary with 10 tokens is the celebration of iranians. The reference summary with 30 tokens contains the reason for the celebration. Some generated summaries with short desired lengths are likely to be incomplete, similar to the truncated version of summaries generated by models without length constraints. The summaries of ad-hoc and LenAtten in Table 2 are not complete and lose the information about "deal".

Source Document
... iranians erupted in celebration as young people waved flags from their sunroofs , blasted music from stereos and chatted online with the hashtag #irantalks . the excitement came after a breakthrough nuclear deal with the united states and other world powers ...

Length	**Reference Summary**
10	iranians celebrate the deal online and in the streets .
30	after a breakthrough nuclear agreement deal with the united states and other world powers , celebration broke out in iranians . young people waved flages and chatted online .

Table 1: The reference summaries of one source document with lengths as 10 and 30.

Generated Summaries (Desired Length=10)
BART (Lewis et al., 2020) + Ad-hoc (Rush et al., 2015) (10 tokens)
iranians erupted in celebration as young people waved flags from
LenAtten (Yu et al., 2021) (12 tokens)
the agreement on the final day of persian new year festivities ,
LPAS (Saito et al., 2020) (22 tokens)
iranians erupted in celebration . the excitement came after a breakthrough nuclear deal with the united states and other world powers .

Table 2: The summaries generated by different models.

Methods based on *information selection* are *two-*

*Kenny Q. Zhu is the corresponding author, and is partially supported by NSFC Grant No. 91646205, and SJTU-CMBCC Joint Research Scheme.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6885 - 6895
May 22-27, 2022 ©2022 Association for Computational Linguistics

stage methods (See et al., 2017; Sarkhel et al., 2020; Saito et al., 2020). One prominent example is LPAS (Saito et al., 2020), which in the first stage, extracts top l most important tokens from the source document as a prototype summary where l is the desired length, and in the second stage encodes the original source document and prototype summary by a dual-encoder. On the one hand, such two-stage approaches suffer from noises introduced in the intermediate results. On the other hand, the second stage of these methods does not have first-hand length information, which weakens the length control. Table 2 shows that LPAS contains redundant information about "deal" and its length is much longer than the reference summary.

In this paper, we propose a **length-aware attention mechanism** (LAAM) which extends a transformer seq2seq model with the ability to select information in the context according to the length constraint. LAAM re-normalizes the attention between encoder and decoder to boost the tokens with higher attention scores based on the desired length, helping with selecting length-aware information from source document. The number of boosted tokens decreases step by step until *eos* gets the highest attention score, which is helpful in stopping the decoding process at desired length. LAAM can be thought of as a hybrid approach between the two types of previous approaches.

We observe that there is a big difference in the number of summaries within different length ranges in the original training set in any summarization dataset. The shorter reference summaries are especially rare. As shown in Table 1, given a short desired length, the summaries of the previous methods and LAAM still select redundant information. To balance the distribution of summaries in different length ranges, we propose a heuristics to create a length-balanced dataset (LBD) by prepredefining the length ranges and constructing extractive summaries within different length ranges, which helps model to select different information from source document via desired lengths.

In our approach, we can create an LBD from original summarization dataset. We first train LAAM on such LBD to enhance the ability of LAAM on information selection with length constraints. Then we fine-tune the pretrained LAAM on original dataset to learn to paraphrase the selected information as abstractive summaries in different lengths. The task of generating short summaries by the models fine-tuned on datasets without short reference summaries can be seen as a *zero-shot* problem. Benefiting from the pretraining with LBD, our approach can solve the zero-shot length control problem.

Our contributions are as follows:

1. We propose a new length-aware attention mechanism (LAAM) to generate high-quality summaries with desired length. LAAM outperforms the state-of-the-art length-controllable methods on CNN/Daily Mail and XSUM in terms of ROUGE scores, length variance and human evaluation (Table 5).

2. We design a heuristics to create a length-balanced dataset (LBD) from original dataset. After pretraining LAAM on LBD, the pretrained LAAM performs better than LAAM and can effectively solve the zero-shot length control problem (Table 10).

2 Approach

In this section, we first introduce the length-controllable summarization (LCS) problem, then introduce the length-aware attention mechanism (LAAM), which attends the existing transformer seq2seq models, and finally explain how to create a length-balanced dataset (LBD) for pretraining.

2.1 Preliminaries

In LCS, the model takes the source document $\mathbf{x} = (x_0, x_1, ..., x_m)$ and the desired length l as input and the summary $\mathbf{y} = (y_0, y_1, ..., y_n)$ as output. x_i is the i^{th} token of document and y_t is the t^{th} token of summary. x_m and y_n are *eos* tokens. The goal is to estimate the conditional probability $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}, l) = \prod_{t}^{n} p(y_t|y_1, y_2, ..., y_{t-1}, \mathbf{x}, l) \qquad (1)$$

We take the transformer seq2seq model (Vaswani et al., 2017) as our basis. Suppose that the encoder output is $\mathbf{h} = \{h_0, h_1, ..., h_m\}$, $\mathbf{h} \in \mathbb{R}^{m \times d}$, and the output of the decoder's masked self-attention sub-layer is $\mathbf{z} = \{z_0, z_1, ..., z_n\}$, $\mathbf{z} \in \mathbb{R}^{n \times d}$. The normal cross attention is calculated as:

$$\mathbf{A} = \text{softmax}(\mathbf{z} \cdot \mathbf{h}^T) \qquad (2)$$

where $\mathbf{A} \in R^{n \times m}$ is an attention matrix. $A_t = \{a_{t,0}, a_{t,1}, ..., a_{t,m}\}$ shows the attention scores of y_t. $a_{t,i}$ is the attention score between y_t and x_i.

2.2 Length-aware Attention Mechanism

In the transformer seq2seq model, the cross attention of an output token y_t is likely to *summarize* those tokens with high attention scores in the input (source document). By formulating the cross attention as a function of the desired length l, we can manipulate the input information selection according to l. This is the intuition behind LAAM, which is illustrated in Figure 1.

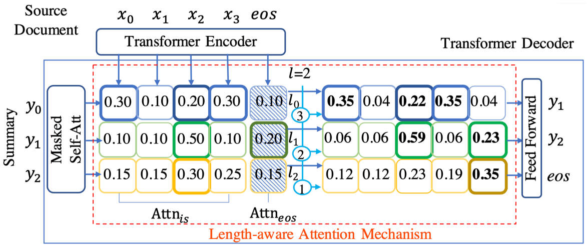


Figure 1: Overview of LAAM on Transformer Seq2seq. The bold values are boosted attention scores. The shadow boxes denote the attention scores of *eos*.

LAAM is made up of two parts: *attention for input selection* ($Attn_{is}$) and *attention for eos token* ($Attn_{eos}$), each optimized for *information selection* and *length control*, the two objectives in LCS.

$\boldsymbol{Attn_{is}}$. At decoding, given the initial desired length l, $l+1$ is the number of tokens in the output with eos, the remaining length budget (l_t) decreases as more tokens are generated. Specifically, at step t,

$$l_t = \begin{cases} l+1-t, & 0 \leq t \leq l \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, at each decoding step, the decoder should plan its output y_t given the remaining number l_t of tokens it will generate. Our key idea is to increase the attention scores of the top l_t tokens with the highest attention scores in A_t, which gives a boost to the chance of these tokens to be selected and summarized. The interesting effect of this is that i) the longer l, the more source information will be selected for summarization; and ii) as the decoder generates more tokens,the number of tokens to be mainly attended in input decreases. We use one-hot vector $\mathbf{p} = \{p_0, p_1, ..., p_m\}$ to label the indices of the top l_t tokens with the highest attention scores in A_t as 1 and others as 0, and then the length-aware attention score is computed as:

$$a'_{t,i} = w_{t,i} \times a_{t,i} \tag{4}$$

$$w_{t,i} = \begin{cases} 1, & p_i = 0 \\ l_t, & p_i = 1 \end{cases} \tag{5}$$

where $w_{t,i}$ is the *weight for boosting the attention* between x_i and y_t. According to Eq. (5), the weight for cross attention decreases as the remaining length decreases, resulting in a decrease in the gap between the enhanced tokens and other tokens. This makes the model evenly attend to tokens related to the enhanced tokens and output general words to end the decoding. The model can learn to select information to be summarized by desired length.

$\boldsymbol{Attn_{eos}}$. At each decoding step t, to enhance the ability of model to generate eos at the desired length, we modify the attention score between y_t and eos in source document x_m as follows:

$$a'_{t,m} = (l+1-l_t) \times a_{t,m} \tag{6}$$

The length-aware attention of eos increases step by step, which demonstrates the probability of stopping decoding will increase as the length of the output close to the desired length.

Finally, we re-normalize the modified attention scores $A'_t = \{a'_{t,0}, a'_{t,1}, ..., a'_{t,m}\}$ to get the context vector $\mathbf{c_t}$ and compute the probability distribution of predicted tokens via:

$$p(y_t|y_{<t}, \mathbf{x}, l) = \text{softmax}(W\mathbf{c}_{t-1} + b) \tag{7}$$

$$\mathbf{c_t} = \sum_{0}^{m} \tilde{a}_{t,i} h_i \tag{8}$$

$$\tilde{a}_{t,i} = \frac{a'_{t,i}}{\sum_{i=0}^{m} a'_{t,i}} \tag{9}$$

where W and b are trainable parameters.

2.3 LBD Creation for Pretraining LAAM

Since the summary lengths of a training dataset may be highly concentrated in a small range (see Table 4), neural-based abstractive summarization models tend to select source information according to the summary lengths they have seen in training data and generate summaries with similar lengths. In order to make the model learn to select proper information according to different desired lengths, we propose a heuristics to create a length-balanced dataset (LBD) by extracting summaries with various lengths from each document in original dataset and makeing lengths of these extractive summaries evenly distributed in different ranges.

Given an abstractive summarization dataset D, which consists of a training set T and a validation set V, we create the training set T' and validation set V' of LBD. To create T', we set the discrete bins $B = \{b_1, b_2, ..., b_k\}$ to represent the ranges of summary length of T'. k is the number of the bins. For example, $B = \{(0, 10], (10, 20], ...\}$ and $b_0 = (0, 10]$. For each document src and its reference summary ref in T, we produce length-controllable pairs (LCPs) consisting of src and its extractive summaries in various length ranges. Let e be the extractive summary of length $b \in B$. We apply a greedy approach, where we add one sentence at a time incrementally to the e, until the length of e is within the proper range of b and has the highest ROUGE-1 (R-1) recall with respect to ref. Generally, the more training data, the greater the impact on the model. To make T' effective, the number of samples in T' should be close to $|T|$. $S(b)$ is the subset of T', including LCPs with extracted summaries with length in b. We add top $\lceil |T|/k \rceil$ extractive summaries (length $\in$ b) with the highest R-1 recall and their source documents to $S(b)$, which makes the summaries equally distributed in the bins or length ranges. The details are in Algorithm 1.

Algorithm 1 Creating Training Set of LBD

```
Input: the training set $T$
Output: the training set $T'$

$rec()$ computes the R-1 recall score between two texts.
$len()$ computes the length of token sequence.
for each training pair $(src, ref) \in T$ do
    $src = \{s_0, s_1, ...\}$, where $s_t$ is the $t^{th}$ sentence in $src$.
    for $i = 0 \rightarrow k$ do
        $min$ and $max$ denote minimum and maximum length of length
        range $b_i$, respectively.
        $e_i \leftarrow \emptyset$
        while $S = \{s | s \in src \cap len(e_i \cup s) \leq max\}$ do
            Select the $s_{sel}$ with best $rec(e_i \cup s_{sel}, ref)$ from $S$.
            if $rec(e_i \cup s_{sel}, ref) > rec(e_i, ref)$ then
                $e_i \leftarrow s_{sel}$; $src \leftarrow src - s_{sel}$
            else
                break
        if $len(e_i) > min$ then
            Add $(src, e_i, rec(e_i, ref))$ to $S(b_i)$.
$S(b_i) \leftarrow$ top $\lceil |T|/k \rceil$ samples from $S(b_i)$ sorted by $rec(e_i, ref)$
$T' \leftarrow S(b_1) \cup S(b_2) \cup \cdots \cup S(b_k)$
return $T'$
```

For V', we create an extractive reference summary by selecting one sentence at a time until we get a subset of sentences from src that maximizes the R-1 F1 with respect to ref. Given an original source document and reference summary pair, R-1 recall computes the similarity between extracted sentences and reference without considering the length of extracted sentences. This meets our requirements for creating T', that is, we can extract multiple summaries within different length ranges for one document. To evaluate the model at training, each document in V' only needs one extractive summary. R-1 F1 considers the difference between the lengths of compared summaries, which can select an extractive summary most similar to the reference in length and content.

In this paper, we first pretrain LAAM on LBD for the ability to select information from source document to be summarized according to length constraint. Then we fine-tune the pretrained LAAM (**PtLAAM**) on original dataset. At this stage, armed with the ability to select information from source document, the model further learns to paraphrase the selected information into abstractive summaries with desired length.

3 Evaluation

We first introduce the datasets and the experimental setup. We design two experiments, **general length control** and **zero-shot length control**, to compare our approach with baselines. [1] General length control experiment trains and tests the models on the entire original dataset. Zero-shot length control experiment tests the model on a subset of the test set whose summary lengths fall within a certain range, and trains the model on training data with summary lengths outside this range. In each of the two experiments, we evaluate methods' ability to do **length control** and **information selection**.

3.1 Datasets

We use two popular summarization datasets. **CNN/Daily Mail** (CNNDM) (Hermann et al., 2015) consists of pairs of a single source document and a multi-sentence summary. The dataset includes 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs. **XSUM** (Narayan et al., 2018) is composed of article and single-sentence summary pairs. The number of samples in training/validation/test sets are 204,045/11,332/11,334.

3.2 Baselines

The existing length-controllable models with good performance are listed in Table 3.

In the experiments, LAAM and PtLAAM are im-

[1] Data and source code are available at: https://github.com/YizhuLiu/lengthcontrol.

Abbrev.	Description
Exact	Ignore eos before generated summary within the desired length and insert eos at the desired length.
LenEmb	Input remaining length. (Kikuchi et al., 2016)
LC	Take desired length as input. (Liu et al., 2018)
GOLC	Apply length-aware loss. (Makino et al., 2019)
LenAtten	Add length attention unit. (Yu et al., 2021)
LPAS	Extract prototype summary. (Saito et al., 2020)
BLPAS	Apply Prot on top of BART

Table 3: The abbreviation and description of methods.

plemented on top of BART[2], because BART (Lewis et al., 2020) is one of the SOTA models in summarization, and it uses less memory and training time than its peers (Shleifer and Rush, 2020). Exact is not a summarization model but is used here to achieve hard length control on any seq2seq models to produce summaries of exact lengths.

3.3 Experimental Setup

We follow Liu et al. (2018) and Saito et al. (2020) to segment datasets by different length ranges and set the discrete bins B of summary length ranges in Sec. 2.3. The B of CNNDM is $B_c = \{(0, 10], (10, 30], ..., (90, +\infty)\}$ and that of XSUM is $B_x = \{(0, 10], (10, 30], (30, +\infty)\}$. [3] B_x has only 3 ranges as the summaries in XSUM are shorter. In zero-shot length control experiments, test length ranges for CNNDM and XSUM is $(0, 30]$ and $(0, 10]$, containing 488 and 176 samples respectively. The length distribution of the datasets is in Table 4. During training, we set the lengths of gold summaries as desired lengths and take them as input. During test, there are two different setups. The **gold length test** (Saito et al., 2020) asks the models to generate summaries with desired lengths equal to the reference summaries. The **arbitrary length test** asks the models to generate summaries with arbitrary lengths, regardless of the reference summary lengths. The output lengths are set at 10, 30, 50, 70 and 90 for CNNDM and at 10, 30 and 50 for XSUM due to the latter's shorter summaries.

In each experiment, to evaluate the ability to control length, we do **soft length control** tests, which sets $minlen$ and $maxlen$ to 0 and 200 respectively during decoding, covering a very large range. It is up to individual models to generate summaries as close as possible to the target length. To evaluate the ability to select information, we utilize **hard length control** at test, which applies Exact in Table 3 to all competing models at decoding.

Data	Length	Train	Val	Test
CNNDM	$(0, 10]$	421	1	1
	$(10, 30]$	20,429	573	487
	$(30, 50]$	114,521	4,255	4,144
	$(50, 70]$	101,461	4,746	4,380
	$(70, 90]$	31,470	2,321	1,509
	$(90, +\infty)$	18,925	1,472	969
	Total	287,228	13,369	11,491
XSUM	$(0, 10]$	3,049	167	176
	$(10, 30]$	193,237	10,732	10,729
	$(30, +\infty)$	77,60	433	429
	Total	204,046	11,332	11,334

Table 4: Length distributions of two datasets.

Following Lewis et al. (2020), we train our model based on *bart.large* with $lr = 3e\text{-}05$ and warmup $= 500$. We set the dropout as 0.1 and momentum as 0.99, and terminate the training when the $lr < 1.0e\text{-}5$. At test time, the batch size is 32. We set beam size as 4 for CNNDM and 6 for XSUM. All experiments are done on an RTX 2080Ti GPU with 11G RAM.

3.4 Evaluation metrics

ROUGE scores: ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) (Lin, 2004) by F1.

Variance (Var): Variance of the summary lengths against the desired length l:

$$var = 0.001 * \frac{1}{n} \sum_{i=0}^{n} |l_i - l|^2, \qquad (10)$$

where n is the number test cases, and l_i is the length of generated summary for case i.

Human Evaluation: We randomly select 50 samples from CNNDM and 50 samples from XSUM. We ask three human annotators who are native or proficient English speakers to score the generated summaries under 3 aspects: Grammatically correct (Gram.): How grammatical the sentences of a summary are?; Informativeness (Info.): How much important information about the source document is included in summary?; Overall: How good is the overall quality of the summary on you criterion? The score of each aspect will be judged as: Poor (1.0), Barely Acceptable (3.0) and Good (5.0).

3.5 Experiment 1: General Length Control

Length control. We use soft length control here. As shown in Table 5 and Figure 2, LAAM and Pt-LAAM achieve higher ROUGE scores and lower variance than all other approaches, which means our approaches can generate good quality summaries with tighter length control. LAAM and

[2] In rest of this paper, LAAM refers to BART using LAAM as cross-attention, for simplicity.

[3] Because historically, to test length control abilities, the test sets of the datasets are split into some predefined ranges, in this work, we adopt the same ranges in creating the bins.

PtLAAM outperform BART, indicating that by controlling lengths effectively, summary quality can be improved, too. LPAS performs better than LenAtten on ROUGE scores but worse on Var, because LPAS focuses more on information selection under the length constraint and overlooks where to stop decoding. BLPAS is better than LPAS as using the pretrained BART as the basic model. BART and BLPAS are considered the previous SOTA methods for length-agnostic summarization and length-controllable summarization respectively. Therefore, we compare our approaches with BART and BLPAS in the remaining experiments.

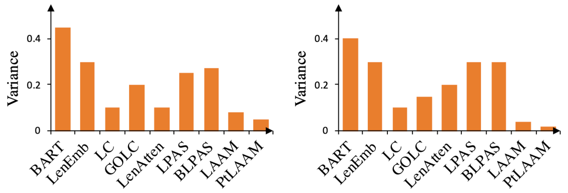
Table 6 also confirms that compared with BART and BLPAS, our best approach PtLAAM gives the best quality summaries by human judges. The summaries generated by PtLAAM achieve better scores in grammatically correct, informativeness and overall. The human evaluation scores of XSUM are lower than those of CNNDM because the summaries in XSUM are much shorter. It is more difficult for a shorter summary to ensure that it is grammatically correct and contains enough information.

	CNNDM			XSUM		
	R-1	R-2	R-L	R-1	R-2	R-L
BART [4]	43.13	20.05	39.32	44.61	21.19	36.00
LenEmb	32.74	13.78	24.50	28.45	8.92	23.13
LC	35.45	14.50	26.02	31.87	11.23	25.94
GOLC	38.27	16.22	34.99	32.94	14.38	26.11
LenAtten	39.82	17.31	36.20	37.20	16.05	31.24
LPAS	42.55	20.09	39.36	43.64	19.81	35.22
BLPAS	42.95	20.29	39.76	44.94	20.31	35.98
LAAM	43.55	20.44	40.63	45.30	21.77	36.64
PtLAAM	**44.17**	**20.63**	**40.97**	**45.48**	**21.80**	**36.84**

Table 5: Gold length test with soft length control. The LAAM and PtLAAM are statistically significantly better than BLPAS with p<0.05 according to t-test.

Data	Model	Gram.	Info.	Overall
CNNDM	Gold	4.6	4.3	4.1
	BART	3.8	2.7	2.2
	BLPAS	3.3	2.9	2.8
	PtLAAM	4.0	3.4	3.3
XSUM	Gold	4.8	3.7	4.5
	BART	3.0	2.9	2.0
	BLPAS	2.1	2.3	2.3
	PtLAAM	3.4	3.0	2.9

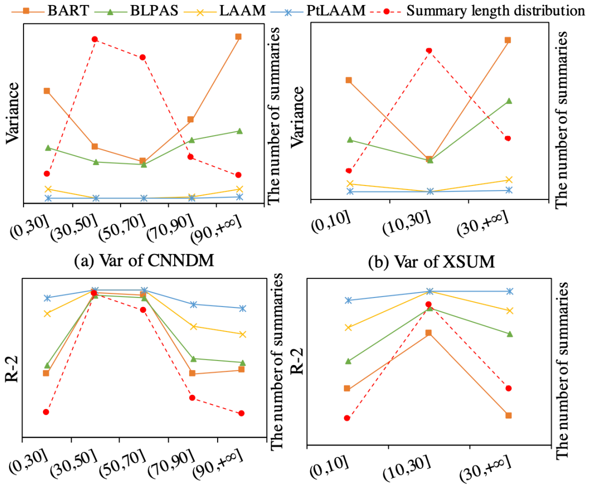
Table 6: Human evaluation. Average Cohen's Kappa is 0.62 among judges, indicating good agreement.



(a) CNNDM (b) XSUM

Figure 2: Variance of generated summary lengths in gold length test with soft length control.

To further test the models' length control ability in different target length ranges, we divide the test data into different sets according to length range in Table 4, and test the models on these sets separately. Figure 3 shows that LAAM and PtLAAM still achieve the lowest Var. For the same length range in Figure 3 and Table 4, the more training data in this range, the lower Var of the generated summaries with respect to the reference summaries within this length range. This denotes that the imbalance length distribution in training data interferes with controlling length. In Figure 3, LAAM and PtLAAM have better and more stable ROUGE scores in all length ranges, illustrating that our approaches are not affected by the summary length distribution in training set and can generate better summaries with desired lengths.



(a) Var of CNNDM (b) Var of XSUM

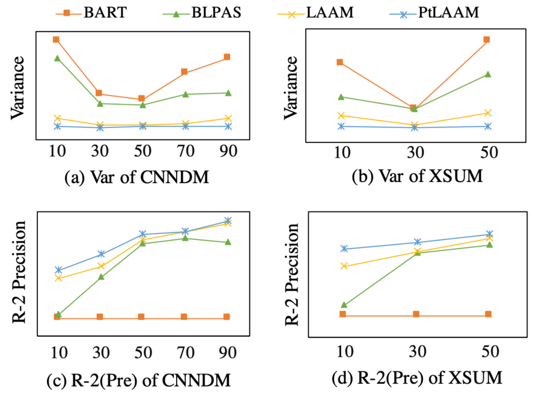
(c) R-2 of CNNDM (d) R-2 of XSUM

Figure 3: Var and R-2(F1) scores of gold length test with soft length control on divided test sets.

The results of arbitrary length test are listed in Figure 4, the lower Var of LAAM and PtLAAM illustrate our approach can control summary length better. As R-2 is the most popular metric in summarization, we report the R-2 related scores of generated summaries. We compute R-2 Precision

[4] We fine-tune the *bart.large* on CNNDM and XSUM via released code in https://github.com/pytorch/fairseq/. Due to incompleteness of the data preprocessing code and possible variance in computing resources and parameters, the results of BART in Table 5 are slightly lower than published version but similar to the numbers reported by others, such as https://github.com/pytorch/fairseq/issues/2541.

(**Pre**) of generated summaries instead of F1, because when the desired length of generated summaries is shorter than reference summary lengths, precision can reflect the accuracy of information selection within that limited budget. In Figure 4, LAAM and PtLAAM get better R-2 (Pre) on both datasets, which means our approaches can select more accurate information. As the desired length increases, the length-controllable models are more likely to select accurate information, causing the gap between our approach and BLPAS to gradually decrease. Bart is not designed to control length, resulting in unchanged R-2 (Pre). Although the arbitrary length test provides a unique perspective in the evaluation of the models, its automatic metric, i.e., R-2 (Pre) is only partial. Therefore, in the rest of the section, we will not do arbitrary length test unless the result is evaluated by human.



(a) Var of CNNDM

(b) Var of XSUM

(c) R-2(Pre) of CNNDM

(d) R-2(Pre) of XSUM

Figure 4: Var and R-2 (Pre) of arbitrary length test with soft length control on complete test sets.

Information selection. Next, we apply hard length control on all models to strictly enforce the exact desired length which is equal to the gold length. The better performance of our proposed approaches in Table 7 indicates that our approaches can cover more important information while producing exactly the same length of the reference summary. Compared to Table 5, our approaches also demonstrate more consistency.

Approach	CNNDM			XSUM		
	R-1	R-2	R-L	R-1	R-2	R-L
BART	43.43	20.11	39.52	44.82	21.34	36.23
BLPAS	43.15	20.52	40.01	45.03	20.57	36.02
LAAM	43.63	20.76	40.63	45.38	21.77	36.64
PtLAAM	**44.21**	**20.77**	**40.97**	**45.53**	**21.82**	**36.85**

Table 7: The ROUGE scores of models in gold length test with hard length control.

As shown in Table 8, the summaries are generated by the SOTA length-controllable approach BLPAS and our best approach PtLAAM with desired length as 10 tokens and 30 tokens. For BLPAS, the summary with desired length as 10 is just the truncated version of the summary with desired length as 30. Different from BLPAS, the content of summaries generated by PtLAAM are changed according to different desired lengths, which denotes that PtLAAM is more effective in selecting information to be summarized by length constraint.

Len	**BLPAS Summaries**	**PtLAAM Summaries**
10	iranians erupted in celebration , as young people waved flages	iranians celebrate online and in the streets after deal .
30	iranians erupted in celebration as young people waved flages , blasted music from stereos and chatted online . the agreement on the final day of persian new year festivities .	the excitement came after a breakthrough nuclear deal with the united states and other world powers . iranians erupted in celebration as young people waved flags and chatted online .

Table 8: Generated summaries of two different lengths from the source document in Table 1.

Ablation Studies. We evaluate the effectiveness of the pretraining LAAM on LBD and length-aware attention mechanism.

Pretraining on LBD. Compared with LAAM only training on original datasets, PtLAAM performs better on R-2 and Var in Figure 4 and Figure 3. The better R-2 scores indicates that the PtLAAM can select more important information with pretrained LAAM on our created dataset LBD. As one source document of LBD may have different extracted summaries within different length ranges, the model trained on LBD can learn to select different information from source document according to the length constraints. Besides, in LBD, the number of summaries with lengths in different ranges is balanced. PtLAAM gets lower Var, which denotes it can control length better. The Var scores in different length ranges are stable, which weakens the negative impact caused by the imbalanced length distribution of training data.

Length-aware attention mechanism, The length-aware attention consists of $Attn_{is}$ and $Attn_{eos}$. Table 9 shows the results of LAAM test on gold length test with soft length control. Compared with LAAM, the LAAM without $Attn_{is}$ has a big drop in ROUGE scores and a small drop in Var score, demonstrating that $Attn_{is}$ mainly focuses on select information with length constraint. The LAAM without $Attn_{eos}$ gets the much lower Var scores but not much difference in ROUGE scores than LAAM, which means that $Attn_{eos}$ is useful in limiting the output length. LAAM outperforms its variant because of the effectiveness of length-aware attention mechanism. Thus, in our experiments, we use PtLAAM model, which trains LAAM with both $Attn_{is}$ and $Attn_{eos}$ on LBD first and then fine-tunes the original datasets, as our best approach.

Data	Model	R-1	R-2	R-L	Var(%)
CNNDM	LAAM	**43.63**	**20.76**	**40.63**	**0.05**
	w/o $Attn_{is}$	42.77	19.32	39.13	0.06
	w/o $Attn_{eos}$	43.10	20.17	37.45	0.13
XSUM	LAAM	**45.38**	**21.77**	**36.64**	**0.03**
	w/o $Attn_{is}$	43.45	20.64	34.79	0.03
	w/o $Attn_{eos}$	44.62	21.32	35.03	0.08

Table 9: Usefulness of two kinds of attentions.

3.6 Experiment 2: Zero-shot Length Control

In this experiment, we use the modified dataset for zero-shot length control (Sec. 3.3). Zero-shot task can test a model's ability to generalize to summary lengths that it has never seen in the original training data before.

Dataset	Length	Approach	R-1	R-2	R-L	Var(%)
Soft length control						
CNNDM	$(0, 30]$	BLPAS	33.04	14.83	29.42	0.14
		LAAM	33.52	15.20	30.54	0.05
		PtLAAM	**33.65**	**15.77**	**31.26**	**0.03**
XSUM	$(0, 10]$	BLPAS	34.37	19.54	31.66	0.10
		LAAM	34.49	20.07	32.10	0.03
		PtLAAM	**35.16**	**20.55**	**32.47**	**0.02**
Hard length control						
CNNDM	$(0, 30]$	BLPAS	30.25	12.51	26.98	-
		LAAM	33.64	15.23	30.76	-
		PtLAAM	**33.78**	**15.89**	**31.30**	**-**
XSUM	$(0, 10]$	BLPAS	32.55	17.16	29.52	-
		LAAM	34.83	20.15	32.10	-
		PtLAAM	**35.16**	**20.58**	**32.49**	-

Table 10: Results of zero-shot length control.

Table 10 shows the performance of PtLAAM on ROUGE scores and Var on different datatsets are the best. For soft length control experiment, the ROUGE scores of different models are similar, because the lengths of summaries generated by BLPAS are longer than reference summary lengths (BLPAS has higher Var scores), which causes the generated summaries to match more tokens in the reference. Because ROUGE (F1) scores usually penalize summaries with longer lengths, PtLAAM, which controls the length better, is still better than other approaches. The lowest Var of our approaches means that our approach can better control summary length. In the hard length control experiment, the ROUGE scores of BLPAS drop a lot since the hard control shortens the length of summaries generated by BLPAS. The best performance of PtLAAM on ROUGE indicate PtLAAM learns to select information based on desired lengths. The ROUGE scores of our approaches are similar to those in soft length control experiment, which indicates our approaches are stable in controlling length. The LAAM performs worse than PtLAAM on ROUGE and Var denotes that the ability of LAAM to control length is impacted by length distribution of the training data. The pretraining on LBD is useful in generating high-quality summaries under desired summary length since the summaries are balanced in different length ranges of LBD.

3.7 Case Study

In this section, we analyze the performance of different models in controlling length.

Input Document
a gym teacher in new hampshire has been accused of posing as a young girl on a social media site and persuading an elementary school student to share inappropriate images of herself ... police charged 34-year-old paul johnson-yarosevich of acton , maine , on monday with prohibited use of computer after they say they discovered he 'd been fooling a pre-teen girl into sending him inappropriate photos of herself by posing as a young girl on social media . authorities soon learned that the girl was sending the photos to a grown man ...

Len	**Generated summaries**	
-	BART	police charged 34-year-old paul johnson-yarosevich of acton , maine , on monday with prohibited use of computer after they say they discovered he 'd been fooling a pre-teen girl into sending him inappropriate photos of herself by posing as a young girl on social media . authorities soon learned that the girl was sending the photos to a grown man .
10	Exact	police charged 34-year-old paul johnson-yarosevich of acton , maine ,
	BLPAS	police charged 34-year-old paul johnson-yarosevich of acton with prohibited use of a computer .
	LAAM	paul was charged with prohibited use of a computer .
	PtLAAM	Paul was prohibited use of computer for cheating .
30	Exact	*police charged 34-year-old paul johnson-yarosevich of acton , maine ,* on monday with prohibited use of computer after they say they discovered he 'd been fooling a pre-teen girl .
	BLPAS	*police charged 34-year-old paul johnson-yarosevich of acton , on monday with prohibited use of computer* . the investigation started in december . after the father of a pre-teen girl told police about the contact .
	LAAM	police charged 34-year-old paul on monday with prohibited use of computer after discovering he 'd been fooling a girl into sending him inappropriate photos of herself on social media .
	PtLAAM	police charged paul , 34 , on monday with prohibited use of computer after discovering he 'd been fooling a pre-teen girl into sending him inappropriate photos on social media .

Table 11: The generated summaries of Table 1 of various desired length **Len**. The *italicized* tokens repeat significant parts the shorter summaries. The red is the tokens longer than desired length.Here Exact refers to the BART using Exact at test, to be fair.

We use the example in Table 11 to analyze different length-controllable methods since the summaries of this example generated by different models are obviously different in *length control* and *information selection*.

As shown in Table 11, BART itself cannot control the length of generated summaries. So, the length of the summary generated by BART is always much longer for covering more information from source document. After adding Exact at test time, BART can generate summary with length exactly the same as desired length. But, as a *early-stop during decoding methods*, Exact always produce incomplete summaries. The summary with 30 tokens of Exact repeats its summary with 10 tokens during generation. Because such methods ignore that the summaries with different lengths of one document should represent different information of source document. BLPAS tends to select more information with length constraints, which may generate summaries with length longer than desired length (the red part in Table 11). The lengths of summaries generated by LAAM and PtLAAM in Table 11 are the same as the desired lengths.

Compared with PtLAAM, given the desired length as 10, LAAM loses the important information about the reason why Paul was charged as there are few training pairs with summary lengths as 10. PtLAAM pretrained on LBD can select information according to various desired lengths as the summary lengths in LBD are evenly distributed in different length ranges. The summaries with desired length as 30 of LAAM and PtLAAM are more similar than their summaries with desired length as 10. This is because there are many more summaries with length about 30 than those with length about 10 in original dataset. Thus, PtLAAM is more effective in generating summaries of lengths that do not appear in the original datasets.

4 Related Work

Previously, most length-controllable approaches in abstractive summarization focused on stoping decoding at a particular time. Ad-hoc (Rush et al., 2015) generated the *eos* token by assigning a score of -∞ to the tokens in vocabulary and generated a fixed number of words. LenEmb and LenInit (Kikuchi et al., 2016) input length embeddings to decoder respectively. Bian et al. (2019) took LenEmb and LenInit as an agent and adjusted the reward incorporating with the desired length. LC (Liu et al., 2018) added the desired length into the first layer of CNN encoder. GOLC (Makino et al., 2019) optimized LenEmb and LC by formalizing loss with an overlength penalty. Fan et al. (2018) predefined some special markers to denote different length ranges and prepended the input with such markers during training and testing. Takase and Okazaki (2019) extended the sinusoidal positional encoding (Vaswani et al., 2017) to take account of stepwise remaining length. LenAtten (Yu et al., 2021) added a length attention unit to exploit proper length information based on the stepwise remaining length.

Other length-controllable approaches decided the content to be summarized by length-aware intermediate summaries. LPAS (Saito et al., 2020) extracted a word sequence with the desired length from source document and generated summary by a non-length-controllable model with document and extracted summary as input. MLS (Sarkhel et al., 2020) generated a general summary and then input it to a length-controllable model.

Compared with previous methods, our approach can effectively control the length of generated summaries by pretraining the length-controllable information selection model on length-balanced dataset. Meanwhile, it can generate summaries with length approximate to the desired length in zero-shot controlling length problem.

Recently, the approaches fine-tune the pretrained transformer seq2seq models (Lewis et al., 2020; Zhang et al., 2020; Dou et al., 2021; Liu and Liu, 2021) on summarization datasets. They achieve outstanding performances on summarization tasks. Our approach is applied to transformer seq2seq model, which is orthogonal to above pretrained transformer models and can be added to them.

5 Conclusion

We present a novel approach to produce summaries in desired length that are fluent and coherent. This approach pretrains a transformer seq2seq model whose cross attention between input and output are re-normalized accordingly to the length requirement. The pretraining is done over synthetic summarization data extracted from the original training set but with summary lengths evenly distributed. Our results show that the framework achieves a good balance between information selection from input documents and length control when producing summaries.

References

Junyi Bian, Baojun Lin, Ke Zhang, Zhaohui Yan, Hong Tang, and Yonghe Zhang. 2019. Controllable length control neural encoder-decoder via reinforcement learning. *arXiv preprint arXiv:1909.09492*.

Asli Çelikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 1065–1072. Association for Computational Linguistics.

Yizhu Liu, Qi Jia, and Kenny Q. Zhu. 2021. Keyword-aware abstractive summarization by extracting set-level intermediate summaries. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2.

Yizhu Liu, Zhiyi Luo, and Kenny Q. Zhu. 2018. Controlling length in abstractive summarization using a convolutional neural network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global optimization under length constraint for neural text summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*.

Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, Atsushi Otsuka, Hisako Asano, Junji Tomita, Hiroyuki Shindo, and Yuji Matsumoto. 2020. Length-controllable abstractive summarization by guiding with summary prototype. *arXiv preprint arXiv:2001.07331*.

Ritesh Sarkhel, Moniba Keymanesh, Arnab Nandi, and Srinivasan Parthasarathy. 2020. Interpretable multi-headed attention for abstractive summarization at controllable lengths. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation.

Sho Takase and Naoaki Okazaki. 2019. Positional encoding to control output sequence length. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*.

Zhongyi Yu, Zhenghao Wu, Hao Zheng, Zhe XuanYuan, Jefferson Fong, and Weifeng Su. 2021. Lenatten: An effective length controlling unit for text summarization. *arXiv preprint arXiv:2106.00316*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*.