Global Optimization under Length Constraint for Neural Text Summarization

Takuya Makino and **Tomoya Iwakura**
Fujitsu Laboratories, Ltd.
{makino.takuya,iwakura.tomoya}@fujitsu.com

Hiroya Takamura
AIST
takamura.hiroya@aist.go.jp

Manabu Okumura
Tokyo Institute of Technology
oku@pi.titech.ac.jp

Abstract

We propose a global optimization method under length constraint (**GOLC**) for neural text summarization models. GOLC increases the probabilities of generating summaries that have high evaluation scores, ROUGE in this paper, within a desired length. We compared GOLC with two optimization methods, a maximum log-likelihood and a minimum risk training, on CNN/Daily Mail and a Japanese single document summarization data set of The Mainichi Shimbun Newspapers. The experimental results show that a state-of-the-art neural summarization model optimized with GOLC generates fewer overlength summaries while maintaining the fastest processing speed; only 6.70% overlength summaries on CNN/Daily and 7.8% on long summary of Mainichi, compared to the approximately 20% to 50% on CNN/Daily Mail and 10% to 30% on Mainichi with the other optimization methods. We also demonstrate the importance of the generation of in-length summaries for post-editing with the dataset Mainich that is created with strict length constraints. The experimental results show approximately 30% to 40% improved post-editing time by use of in-length summaries.

1 Introduction

Automatic text summarization aims at generating a short and coherent summary of a given text. In text summarization, while the generated summaries should contain the important content of the input text, their *lengths* should also be controlled, e.g., the summary should be as long as the width of target devices such as smart-phones and digital signage. Therefore, editors have to summarize a source text under a length constraint by reordering and paraphrasing.

For summarization, both extractive and abstractive methods have been widely studied. Extractive methods are based on selection of sentences from source texts without using reordering or paraphrasing. In contrast, abstractive methods generate summaries as new sentences. Therefore, abstractive methods can rely on the reordering and paraphrasing required for summary and title generation. However, most abstractive summarization methods are not able to control the summary length.

To this problem, Kikuchi et al. (2016) and Liu et al. (2018) proposed abstractive summarization models with a capability of summary length control. One is an LSTM based summarization model, and the other is a CNN based one. They proposed to enforce the desired length in the decoding of training and generation. Their models, however, leave much room for improvement, at least with regard to two aspects. One aspect is that the summarization performance is still worse than other state-of-the-art models. The other is that their models sometimes fail to control the output length.

In this paper, we address these two issues by incorporating global training based on a minimum risk training (MRT) under the length constraint. MRT (Och, 2003) is used to optimize a model globally for an arbitrary evaluation metric. It was also applied for optimizing the neural summarization model for headline generation with respect to ROUGE (Ayana et al., 2017), which is based on an overlap of words with reference summaries (Lin, 2004). However, how to use MRT under a length constraint was an open problem; thus we propose a global optimization under length constraint (GOLC) for neural summarization models. We show that neural summarization models trained with GOLC can control the output length better than the existing methods. This is because our training procedure makes use of overlength summaries. While the probabilities of generating sum-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1039–1048
Florence, Italy, July 28 - August 2, 2019.

maries that satisfy the length constraint increase, overlength summaries are penalized and hence the probabilities of generating such summaries decrease.

We conducted experiments on CNN/Daily Mail and a Japanese single document summarization dataset of the Mainichi Shimbun Newspapers. Models trained with GOLC showed better ROUGE scores than those of maximum log-likelihood based methods while generating summaries satisfying the length constraint. In contrast to the approximately 20% and 50% of overlength summaries generated by the other state-of-the-art models, our method only generated 6.70% of overlength summaries on CNN/Daily and 7.8% on long summary of Mainichi while improving ROUGE scores.

We also demonstrate the importance of the generation of in-length summaries for post-editing. The experimental results of post-editing generated summaries showed that generated in-length summaries contributed to an approximately 30% to 40% improved post-editing time.

2 Related Work

There are many models for text summarization such as rule-based models (Dorr et al., 2003) and statistical models (Banko et al., 2000; Zajic et al., 2004; Filippova and Strube, 2008; Woodsend et al., 2010; Filippova and Altun, 2013). Recently, abstractive summarization models based on neural encoder-decoders have been proposed (Rush et al., 2015; Chopra et al., 2016; Zhou et al., 2017; Paulus et al., 2018). There are mainly two research directions: model architectures and optimization methods.

Pointer networks (Vinyals and Le, 2015; Gulcehre et al., 2016; See et al., 2017) and copy mechanisms (Gu et al., 2016; Zeng et al., 2016) have been proposed for overcoming the unknown word problem. Other methods for the improvement of abstractive summarization models include use of existing summaries as soft templates with a source text (Li et al., 2018) and extraction of actual fact descriptions from a source text (Cao et al., 2018). Although summary length control of abstractive summarization has been studied, previous studies focus on incorporation of a length controlling method to neural abstractive summarization models (Kikuchi et al., 2016; Fan et al., 2018; Liu et al., 2018; Fevry and Phang, 2018; Schumann, 2018). In contrast, our research focuses on a global optimization method.

Optimization methods for optimizing a model with respect to evaluation scores, such as reinforcement learning (Ranzato et al., 2015; Paulus et al., 2018; Chen and Bansal, 2018; Wu and Hu, 2018) and minimum risk training (Ayana et al., 2017), have been proposed for summarization models based on neural encoder-decoders. Our method is similar to that of Ayana et al. (2017) in terms of applying MRT to neural encoder-decoders. There are two differences between our method and Ayana et al.'s: (i) our method uses only the part of the summary generated by a model within the length constraint for calculating the ROUGE score and (ii) it penalizes summaries that exceed the length of the reference regardless of its ROUGE score.

3 Summary Length Controllable Models

In this section, we describe two summarization models that are optimized by GOLC for generating summaries within length constraints. These two models are also optimized with maximum log-likelihood estimation (MLE) that is widely applied for training neural encoder-decoders of the original papers of the summarization models and a minimum risk training (MRT).

3.1 LSTM based Model (PG w/ LE)

Kikuchi et al. (2016) proposed *LenEmb* (LE) that is a variant of LSTM that takes into account the remaining length of a summary in training and generation. The remaining length of a summary is initialized as the length of the reference summary in training and as the desired length in generation. For each time step in decoding, the length of a generated word is subtracted from the remaining length of a summary.

We integrate LE into a pointer-generator network (See et al., 2017), which is a state-of-the-art neural summarization model. A pointer-generator consists of a pointer network and an LSTM encoder-decoder. A pointer network can copy words of a source text into a summary even if they are out-of-vocabulary. Probability of generating a word is calculated based on linear interpolation between probability distribution of vocabulary, and attention distribution of source words. We replaced an LSTM decoder of a pointer-

generator with LenEmb, which we call this model PG w/ LE.

3.2 CNN based Model (LC)

Liu et al. (2018) proposed CNN based encoder-decoders for controlling summary length. This model uses the variant of a CNN decoder that takes into account the desired length of a summary. In CNN based encoder-decoders, the representations of words are the concatenation of word embeddings and position embeddings (Gehring et al., 2017). This model is trained to generate <EOS>, which is the end of a sentence, when the number of generated words in a summary is the desired length.

We note that the length of a summary is the number of words in the summary in Liu et al. (2018), while the length of a summary is the number of characters in the summary in Kikuchi et al. (2016),

$\Delta(\boldsymbol{y}, \boldsymbol{y}') = -\text{ROUGE1}($
$\langle$malaysia, markets, closed, for, holiday$\rangle$,
$\langle$markets, in, malaysia, closed, for, holiday$\rangle$
$)$=-1.0

(a) Example of the original $\Delta(\mathbf{y}, \mathbf{y}')$.

trim($\boldsymbol{y}'$, byte($\boldsymbol{y}$)):
$\langle$markets, in, malaysia, closed, for $\rangle$

$\tilde{\Delta}(\boldsymbol{y}, \boldsymbol{y}') = -\text{ROUGE1}($
$\langle$malaysia, markets, closed, for, holiday$\rangle$,
$\langle$markets, in, malaysia, closed, for$\rangle$
$) + \max(0, 38 - 35) = -0.8 + 3.0 = 2.2$

(b) Example of the proposed $\tilde{\Delta}(\mathbf{y}, \mathbf{y}')$.

Figure 1: Examples of $\Delta(\mathbf{y}, \mathbf{y}')$ of the original MRT and $\tilde{\Delta}(\mathbf{y}, \mathbf{y}')$ of GOLC where ROUGE-1 recall is calculated based on unigrams. In the two examples, a reference $\mathbf{y}$ is $\langle$malaysia, markets, closed, for, holiday$\rangle$ and a sampled summary $\mathbf{y}'$ is $\langle$markets, in, malaysia, closed, for, holiday$\rangle$ and $c_b(\mathbf{y}) = \texttt{len(' '.join(}\mathbf{y}\texttt{))} = 38$ and $c_b(\mathbf{y}') = \texttt{len(' '.join(}\mathbf{y}'\texttt{))} = 35$.

4 Conventional Optimization Methods

In this section, we describe MLE, and MRT that are used for training summarization models. We denote a source sentence as $\mathbf{x} = \langle x_1, ..., x_N \rangle$, where $x_i (1 \leq i \leq N)$ is a word in $\mathbf{x}$ and a summary as $\mathbf{y} = \langle y_1, ..., y_M \rangle$, where $y_j (1 \leq j \leq M)$ is a word in $\mathbf{y}$.

4.1 Maximum Log-likelihood Estimation

MLE aims at maximizing log-likelihood on training data D:

$$L_{MLE}(\theta) = \sum_{(\mathbf{x},\mathbf{y}) \in D} \log p_\theta(\mathbf{y}|\mathbf{x}), \qquad (1)$$

where $p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{M} p(y_t|\mathbf{y}_{<t}, \mathbf{x})$. For each time step in decoding, a model calculates the probability of generating a target word in a reference summary, then, the target word is used as the next input of a decoder. We see that a model never generates overlength summaries since words in a reference summary are used as inputs of a decoder. Thus, the way of decreasing the probability of generating overlength summaries is not trivial.

4.2 Minimum Risk Training

In MRT, unlike MLE, the probability of a word at each step is calculated using previously generated words as in the test phase. MRT aims at optimizing a model for an evaluation metric by minimizing the expected loss on D:

$$L_{MRT}(\theta) = \sum_{(\mathbf{x},\mathbf{y}) \in D} \sum_{\mathbf{y}' \in \tilde{S}(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x}) \Delta(\mathbf{y}, \mathbf{y}'), \qquad (2)$$

where $Q_\theta(\mathbf{y}'|\mathbf{x}) \propto p_\theta(\mathbf{y}'|\mathbf{x})^\gamma$. $\Delta(\mathbf{y}, \mathbf{y}')$ is a loss function of the negative ROUGE between a reference summary $\mathbf{y}$ and a summary to be evaluated $\mathbf{y}'$, γ is a smoothing factor and $\tilde{S}(\mathbf{x}) = S(\mathbf{x}) \cup \{\mathbf{y}\}$ (Shen et al., 2016). $S(\mathbf{x})$ is a set of summaries that can be generated by a model for a given $\mathbf{x}$. Including reference summaries into the set of sampled summaries can increase the probabilities of generating reference summaries, which will be analyzed in Section 6.

From Equation (2), we see that the probability of generating a summary is weighted by its ROUGE score. Since MRT optimizes summarization models in terms of a ROUGE score, the length of summaries generated by models depends on the type of a ROUGE score, i.e., summary lengths will be long if we choose ROUGE recall as Δ, while summary lengths will be short if we choose ROUGE precision as Δ. By choosing the ROUGE F score as Δ, the length of a generated summary will be balanced, though there is no relation with whether or not the summary is overlength.

Therefore, output length controllable models lose the ability of generating summaries with a desired length. These models assume generating

<EOS> when the remaining length of a summary is 0, or the length of a summary reaches the desired length by using words in a reference summary in MLE.

5 Global Optimization under Length Constraint

Compared to conventional methods, the proposed method (GOLC) optimizes models under length constraint. To take into account the length constraint, we modify Δ of the original MRT to $\tilde{\Delta}$ that has an overlength penalty. We formalize loss function for optimization under length constraint as follows:

$$L_{GOLC}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \sum_{\mathbf{y}'\in\tilde{S}(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x})\tilde{\Delta}(\mathbf{y},\mathbf{y}'), \tag{3}$$

where $Q_\theta(\mathbf{y}'|\mathbf{x}) \propto p_\theta(\mathbf{y}'|\mathbf{x})^\gamma$. $\tilde{\Delta}(\mathbf{y},\mathbf{y}')$ is formalized as follows:

$$\begin{aligned}\tilde{\Delta}(\mathbf{y},\mathbf{y}') = &-\texttt{ROUGE}(\mathbf{y},\texttt{trim}(\mathbf{y}',c_*(\mathbf{y}))) \\ &+ \max(0, c_*(\mathbf{y}') - c_*(\mathbf{y})),\end{aligned} \tag{4}$$

where ROUGE calculates the ROUGE score between two texts. We used ROUGE-L recall as a score function. $\texttt{trim}(\mathbf{y}', c_*(\mathbf{y}))$ extracts the longest subsequence of words in $\mathbf{y}'$ under the length constraint $c_*(\mathbf{y})$. $c_*(\mathbf{y})$ denotes the length of $\mathbf{y}$. The number of characters in a summary $c_b(\mathbf{y})$ is used for PG w/ LE: $c_b(\mathbf{y}) = \texttt{len('' ''.join}(\mathbf{y}))$ for English, and $c_b(\mathbf{y}) = \texttt{len(''.join}(\mathbf{y}))$ for Japanese[1]. The number of words in a summary is used for LCs: $c_w(\mathbf{y}) = |\mathbf{y}|$.

The first term in Equation (4) decreases the loss when a part of a generated summary within the length constraint covers word n-grams of the reference summary. The part of a generated summary that exceeds the length constraint does not affect the calculation of the ROUGE score. The second term increases the loss if a generated summary is longer than the reference summary. Figure 1 shows examples of $\Delta(\mathbf{y},\mathbf{y}')$ of the method by Ayana et al. (2017) (a) and our loss function (b).

[1]A difference between calculating the number of characters in an English summary and that in a Japanese one is whether or not the length of space between words is counted.

6 Analysis of GOLC

In this section, we argue that GOLC is more suitable for training neural encoder-decoders under a length constraint by comparing our objective function with the existing ones. In addition, we analyze the contribution of reference summaries in MRT. $L_{MRT}(\theta)$ of Equation (2) can be written as:

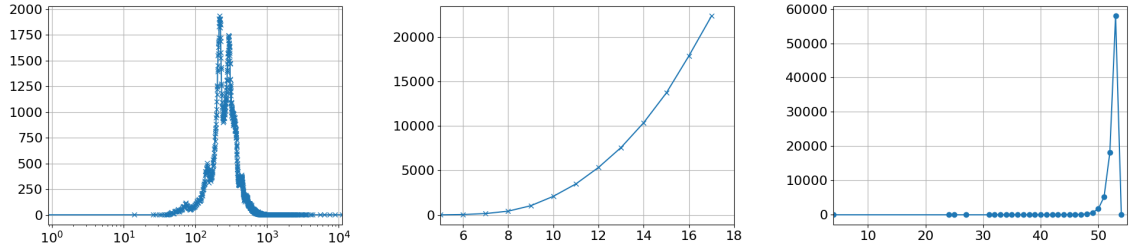
$$\begin{aligned}L_{MRT}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \{&-Q_\theta(\mathbf{y}|\mathbf{x}) + \\ &\sum_{\mathbf{y}'\in S(\mathbf{x})} Q_\theta(\mathbf{y}'|\mathbf{x})\Delta(\mathbf{y},\mathbf{y}')\},\end{aligned} \tag{5}$$

because $\Delta(\mathbf{y},\mathbf{y}) = -1$ for a reference summary $\mathbf{y}$. From this equation, if negative ROUGE recall is used as the loss function, we observe that the probability of each reference summary, which has the best ROUGE score and readability, largely increases. However, the probability of generating overlength summaries may increase from decreasing $L_{MRT}(\theta)$ because a longer summary tends to result in a higher ROUGE recall score.

In contrast, $L_{GOLC}(\theta)$ of Equation (3) can take into account overlength penalties and can be rewritten with $\tilde{\Delta}(\mathbf{y},\mathbf{y}) = -1$ as:

$$\begin{aligned}L_{GOLC}(\theta) = \sum_{(\mathbf{x},\mathbf{y})\in D} \{&-Q_\theta(\mathbf{y}|\mathbf{x}) \\ &- \sum_{\mathbf{y}^-\in S^-(\mathbf{x})} Q_\theta(\mathbf{y}^-|\mathbf{x}) \left|\tilde{\Delta}(\mathbf{y},\mathbf{y}^-)\right| \\ &+ \sum_{\mathbf{y}^+\in S^+(\mathbf{x})} Q_\theta(\mathbf{y}^+|\mathbf{x})\tilde{\Delta}(\mathbf{y},\mathbf{y}^+)\},\end{aligned} \tag{6}$$

where $S^-(\mathbf{x}) = \{\mathbf{y}'|\mathbf{y}' \in S(\mathbf{x}) \wedge \tilde{\Delta}(\mathbf{y},\mathbf{y}') < 0\}$ and $S^+(\mathbf{x}) = \{\mathbf{y}'|\mathbf{y}' \in S(\mathbf{x}) \wedge \tilde{\Delta}(\mathbf{y},\mathbf{y}') \geq 0\}$. Note that in the second term of the right-hand side, the absolute value $|\tilde{\Delta}(\mathbf{y},\mathbf{y}^-)|$ is used. Since $Q_\theta(\mathbf{y}'|\mathbf{x}) \geq 0$ holds true for any $\mathbf{y}'$ by definition and $\tilde{\Delta}(\mathbf{y},\mathbf{y}') \geq 0$ also holds true for overlength summary $\mathbf{y}'$, we see the following for minimizing $L_{GOLC}(\theta)$. Each $Q_\theta(\mathbf{y}^-|\mathbf{x})$ for summaries shorter than the length constraint increases because $\tilde{\Delta}(\mathbf{y},\mathbf{y}^-) < 0$. In contrast, each $Q_\theta(\mathbf{y}^+|\mathbf{x})$ for overlength summaries decreases because $\tilde{\Delta}(\mathbf{y},\mathbf{y}^+) > 0$. As a result, the possibility of generating overlength summaries is reduced. Of course, the probability of each reference summary in $L_{GOLC}(\theta)$ also largely increases because $\tilde{\Delta}(\mathbf{y},\mathbf{y}) = -1$.



(a) CNN/Daily (b) Mainichi (short) (c) Mainichi (long)

Figure 2: Summary length distributions on CNN/Daily and Mainichi. Summary length is the number of characters.

7 Experimental Settings

We compare our optimization method GOLC with two different optimization methods, MLE and MRT by applying the optimization methods to LSTM and CNN-based summarization models on an English and a Japanese dataset. We implemented summarization models with Chainer (Tokui et al., 2015) and all summarization models were trained on NVIDIA Tesla P100.

7.1 Dataset

CNN/Daily: We created the non-anonymized version of the summarization dataset following See et al. (2017) from the CNN/Daily Mail corpus. These data contain news documents paired with multi-sentence summaries. We obtained 287,226 training pairs, 13,368 development pairs, and 11,490 test pairs. The vocabulary was created by collecting top 500,000 words in terms of their frequency in training data as in (See et al., 2017).

Mainichi: Mainichi contains Japanese news articles with their summaries from 2012 to 2017 of the Japanese newspaper company *The Mainichi Newspapers Co, Ltd*. For each news article which consists of a headline and a body, two summaries are included: a short summary with the maximum length of 17 characters, and that of 54 characters. For tokenizing Japanese texts, we used MeCab[2]. We used the first 200 words of each news article, which is concatenation of a headline and a body, for the input of an encoder. We created training data from all data from 2012 to 2016 and some of the data from 2017. The rest of the 2017 data were used as test data.

[2]https://github.com/taku910/mecab

Hyperparameter	Data	PG	LC
batch size of MLE	C	16	8
	M	30	8
batch size of MRT,GOLC	C, M	5	5
word embedding size	C, M	128	128
hidden state size	C, M	256	256
number of hidden layers	C, M	1	4
sample size of $\tilde{S}$	C, M	10	10
smoothing factor γ	C, M	5e-3	5e-3
gradient clip	C, M	2.0	0.1
dropout	C, M	0	0.2

Table 1: Hyperparamters used in experiments of CNN/Daily (C) and Mainichi (M).

Note that the test data were randomly sampled from the 2017 data. The sizes of the training data and test data are 163,220 and 2,000. Half of the dataset is document-long summary pairs, and the rest of the dataset is document-short summary pairs. The vocabulary was created by collecting words that occur more than two times in the training data. Words that are not included in the vocabulary were replaced with the special token, <UNK>.

Figure 2 shows summary length distributions on CNN/Daily and Mainichi. Compared to the length distribution of summaries in CNN/Daily, the one of long summaries in Mainichi has a low variance. Almost all long summaries are 50-54 characters in Mainichi while lengths of almost all summaries are 10^2-10^3 in CNN/Daily.

7.2 Summarization Models to be Compared

We compared a state-of-the-art model that is not capable of controlling summary length, and two length controllable models and simple baselines that extract the first part of source text.

LEAD extracts the first part of source text. For CNN/Daily, we used reported scores of LEAD-3sent that extracts first three sentences of a source text (See et al., 2017). For long summaries and short summaries in Mainichi, we used the first 54 characters (LEAD-54chars) and 17 characters of a source text (LEAD-17chars).

PG is an LSTM-based standard pointer-generator that does not have capability of summary length control. PG showed state-of-the-art performances on CNN/Daily. Therefore, we used PG in our evaluation. We trained two models of PG for short summaries and long summaries on Mainichi because this model cannot control summary length.

PG w/ LE is an extension of the PG with length embeddings (LenEmb) proposed by Kikuchi et al. (2016). We set the dimension of remaining summary length embeddings to 100 and the number of length types to 401 (i.e., 0 to 400). If the remaining length of a summary is larger than 400, we kept using 400 as the input of LenEmb until the actual remaining length is less than 401.

LC (Liu et al., 2018) is a convolutional encoder-decoder-based summarizing model for controlling the summary length. In contrast to PG w/ LE, we use the number of the remaining words to be outputted instead of the number of characters by following the original settings.

7.3 Optimization Methods to be Compared

MLE is the optimization method based on the maximum log-likelihood estimation of Equation (1).

MRT optimizes models with respect to a ROUGE score of Equation (2).

GOLC is our method for globally optimizing length controllable models under a length constraint of Equation (3).

Before applying MRT and GOLC to summarization models, they are trained with MLE. We used Adam (Kingma and Ba, 2014) ($\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) for updating the LenEmbs, and Nesterovs Accelerated Gradient (Bengio et al., 2013) for updating the LCs. Other hyperparameters of models and optimization methods used in our experiments are summarized in Table 1. We halve the word embedding size, hidden state size, and the number of layers of LC from the original setting of Liu et al. (2018). This is because avoiding out-of-memory error on our GPU when applying MRT, and GOLC, and our objective of the experiments with LC is the evaluation of length control ability of each optimization method.

7.4 Evaluation Metrics

ROUGE We used ROUGE F-score on CNN/Daily. When calculating ROUGE F-score, full-length summaries are used. We also used ROUGE recall on Mainichi with a length constraint, which is the length of a reference summary. Overlength summaries are truncated to the length constraint for evaluating ROUGE recall scores.

We used `pyrouge`[3], which is the same evaluation script used by See et al. (2017), scores on CNN/Daily. This is because the `pyrouge` does not support Japanese. Therefore, we used `sumeval`[4] with the MeCab on the ROUGE evaluation of the Mainichi dataset.

Length controllability For evaluating the capability of summary length control, we use two metrics. The first one is the variance of a summary length $c_*(\mathbf{y}_i)$ against the desired length l_i (Liu et al., 2018):

$$Var_* = 0.001 * \frac{1}{n}\sum_{i=0}^{n}|l_i - c_*(\mathbf{y}_i)|^2. \quad (7)$$

The other is *%over* that is calculated by dividing the number of overlength summaries with the number of test data. Because of the difference of the length unit between LenEmb and LC, *Var* and *%over* of LenEmb and those of LC are not comparable. Since GOLC optimizes length controllable models, we compare models optimized by GOLC with models trained with other optimization methods.

Average time of generation (avg. time) We evaluated average time of generation of summaries on CPU per new article.

[3]https://github.com/andersjo/pyrouge
[4]https://github.com/chakki-works/sumeval

pointer-generator (PG)

Sum. Model (Opt. Method)	R-1 F	R-2 F	R-L F	Var_b	$\%over$	avg. time (sec.)
LEAD-3sents (See et al., 2017)	40.34	17.70	36.57	-	-	-
PG (MLE)	37.74	15.78	34.35	19.35	58.35	15.25
PG w/ LE (MLE)	37.45	15.31	34.28	**4.50**	19.11	12.83
PG w/ LE (MRT)	**38.47**	**16.30**	**35.30**	18.74	43.32	24.13
PG w/ LE (GOLC)	38.27	16.22	34.99	5.13	**6.70**	**10.31**

length control CNN (LC)

Sum. Model (Opt. Method)	R-1 F	R-2 F	R-L F	Var_w	$\%over$	avg. time (sec.)
LC (MLE)	30.67	11.00	**28.97**	**0.17**	44.67	16.93
LC (MRT)	**31.02**	**11.29**	28.54	0.21	61.67	17.19
LC (GOLC)	29.38	10.38	27.18	0.22	**21.55**	**16.41**

Table 2: Experimental results of three summarization models (Sum. Model), PG and LC on CNN/Daily, with three optimization methods (Opt. Method), MLE, MRT and GOLC. The best score in each column is shown in bold. The length of a reference summary was used as a desired length for length controllable models. LC originally has capability of summary length control. Therefore, we only compare the differences obtained with optimization methods. The avg. time indicates a number for the average summary generation time (seconds).

Human Evaluation We also evaluate post-editing time of automatically generated summaries for human post-editing.

8 Experimental Results

8.1 ROUGE

Table 2 shows ROUGE scores (F-scores of full length summaries), *Var*, and *%over* on CNN/Daily. PG w/ LE trained with GOLC shows better ROUGE scores and better *%over* than those of MLE. Although ROUGE scores of PG w/ LE trained with MRT showed better ROUGE scores than GOLC, *%over*s are higher than those of GOLC. From these results, we see that GOLC improves ability to generate summaries under length constraints while maintaining ROUGE scores. ROUGE scores of LCs are lower than those of pointer-generator (See et al., 2017) and PG of our implementation. This is because LC could not copy words of a source text into its target text. The difference between ROUGE scores and Var_w of LC and reported scores in Liu et al. (2018) is due to differences of hyperparameters between ours and the original paper.

Table 3 shows ROUGE scores (recall of truncated summaries), *Var*, and *%over* on Mainichi. ROUGE scores of PG w/ LE are higher than those of PG. This is because PG w/ LE was able to trained with two times larger training data compared to PG. Since PG cannot control summary length, two models were trained for short summaries and for long summaries separately. Although ROUGE scores of neural summarization models are lower than those of LEAD-3sents on CNN/Daily, ROUGE scores of neural summarization models are higher than those of LEAD-54chars and LEAD-17chars. These results come from the difference between the writing rules of summaries and ones of news articles in Mainichi. For example, *yomigana* that indicates phonetic symbols of Japanese kanji characters sometimes follow person names and location names of kanji characters in a news article but not in a summary. Furthermore, noun phrases are often rewritten to shorter paraphrases.

8.2 Length Controllability

We evaluated the length controllability of each optimization method. On CNN/Daily, we used the length of each summary as the length constraint. On Mainichi, for PG w/ LE, we used 17 for short summaries and 54 for long summaries as their length constraints. For LC, we used the number of words in a reference summary as the length constraint because no length constraints with respect to the number of words are given.

We see that ROUGE scores of PG w/ LE trained with GOLC are higher than those of MLE on CNN/Daily and Mainichi. Furthermore, GOLC contributes to reduced generation of over-length summaries compared to other optimization methods on CNN/Daily and long summary on

pointer-generator (PG)

Sum. Model (Opt. Method)	R-1 R	R-2 R	R-L R	Var_b	$\%over$	avg. time (sec.)
LEAD-54chars	48.71	24.33	31.84	-	-	-
PG (MLE)	56.11	36.95	48.66	0.05	29.3	4.65
PG w/ LE (MLE)	56.22	36.58	48.49	0.03	27.1	5.06
PG w/ LE (MRT)	**57.10**	36.93	**49.28**	1.102	18.0	9.47
PG w/ LE (GOLC)	56.44	**36.94**	49.14	**0.0007**	**7.8**	**4.64**
length control CNN (LC)						
Sum. Model (Opt. Method)	R-1 R	R-2 R	R-L R	Var_w	$\%over$	avg. time (sec.)
LC (MLE)	48.40	28.87	41.53	0.0063	16.0	14.57
LC (MRT)	**49.82**	**30.69**	**43.02**	**0.007**	11.7	14.69
LC (GOLC)	42.69	24.83	36.61	0.048	**0.5**	**12.79**

(a) long summary

pointer-generator (PG)

Sum. Model (Opt. Method)	R-1 R	R-2 R	R-L R	Var_b	$\%over$	avg. time (sec.)
LEAD-17chars	51.94	33.21	49.46	-	-	-
PG (MLE)	54.75	40.92	53.70	0.017	**1.9**	**1.20**
PG w/ LE (MLE)	61.31	46.43	59.32	**0.007**	8.2	1.54
PG w/ LE (MRT)	**64.60**	**48.52**	**62.14**	2.53	30.4	10.11
PG w/ LE (GOLC)	62.71	46.88	60.23	0.01	12.2	1.51
length control CNN (LC)						
Sum. Model (Opt. Method)	R-1 R	R-2 R	R-L R	Var_w	$\%over$	avg. time (sec.)
LC (MLE)	46.96	31.43	45.72	0.004	0.	3.36
LC (MRT)	**51.27**	**35.81**	**49.85**	**0.003**	0.	3.33
LC (GOLC)	44.72	29.99	43.53	0.006	0.	**3.23**

(b) short summary

Table 3: Experimental results of (a) long summary and (b) short summary on Mainichi with three optimization methods. The meaning of each item in the first column is the same as Table 2. Summaries generated by models were truncated to the length constraints for calculating ROUGE scores. Length constraints are 17 for short summaries and 54 for long summaries for PG and the number of words in a reference summary in LC.

Mainichi. *%over* of PG w/ LE (GOLC) is larger than that of PG (MLE) and PG w/ LE (MLE). Since short summary lengths distribute approximately 10 to 17, lengths of summaries generated by PG (MLE), which does not has capability of controlling summary length, are less than the length constraint 17. In contrast, PG w/ LE (MLE) and PG w/ LE (GOLC) tend to generate as the same length of summary as the length constraint. As a result, some summaries were overlengthed.

By training LC with GOLC, ROUGE scores degraded while *%over* was improved on Table 2 and Table 3. LC trained with GOLC sometimes generated much shorter summaries against length constraints. Thus, recall scores were lower and hence F-scores were also lower than those of other methods.

8.3 Summarization Speed

We evaluated generation time of models trained with different optimization methods on CNN/Daily and Mainichi. The rightmost columns of Table 2 and Table 3 show average time of summary generation with beamsearch of the beam width 5. We see that GOLC-based summarization is faster than the other methods. One of the reasons is models trained with GOLC usually generate summaries within length constraints. In contrast, the avg. times of the MRT-based models is slower than other methods because the models trained with MRT often generate longer summaries than those of other methods.

We evaluated generation time of models trained with different optimization methods on CNN/Daily and Mainichi. Table 2 and Table 3

Over or Not \ LC	17 chars.	54 chars.
Overlength	21.3 sec.	78.6 sec.
In-length	12.90 sec.	55.7 sec.

Table 4: Human post-editing time on Mainichi Shimbun. LC indicates the number of maximum characters and each time is the average time of post-editing.

also show average time of generation with beam-search of the beam width 5. Since models trained with GOLC usually generate summaries within length constraints, generation time of GOLC is faster than those of MRT.

8.4 Post-Edit Evaluation

In order to demonstrate the importance of the generation of in-length summaries, we evaluate the post-editing time of generated summaries. We recruited 7 Japanese native speakers for this evaluation as editors. The editors are required to edit summaries generated by summarization models if they are overlength and have grammatical errors and factual errors.

We randomly collected 10 overlength summaries and 10 in-length summaries from summaries generated by PG, PG w/LE (MLE), PG w/LE (MRT) and PG w/LE (GOLC) because our objective is to evaluate the importance of the generation of in-length summaries, not comparison of optimization methods.

Table 4 shows the average time of post-editing. The experimental results show that overlength summaries require longer editing time. The reduction is approximately 39.4% for 17 chars and 29.1% for 54 chars. This result indicates that the generation of in-length summaries is important when we use generated summaries for assisting workers. Combined with the Table 3 and Table 4, we estimate use of GOLC-based summarizer contributed to approximately 10% reduction of post-editing time compared with MRT-based one.

We used readability and informativeness for subjective evaluation of the articles of post-editing: Readability (Read.) is evaluation of grammatical correctness of summaries. Informativeness (Info.) is evaluation of coverage of important parts of the original source text under the length constraint. We asked the editors to assign a five scale of 1 (bad) to 5 (good) to summaries of readability and informativeness. Table 5 shows readability and informativeness are improved by post-editing. Therefore, we see the post-editing results were reasonable.

	17 chars.		54 chars.	
Sum.	Read.	Info.	Read.	Info.
no-edit				
In-length	2.8	2.4	3.4	2.8
Overlength	2.6	3.2	3.6	3.8
edit				
In-length	4.2	4.2	4.0	4.2
Overlength	3.8	4.4	4.8	4.6

Table 5: Evaluation results of Readability (Read.) and Informativeness (Info.).

9 Conclusion

We proposed a global optimization method for neural text summarization under a length constraint. Our methods outperformed the conventional methods in terms of both ROUGE, while maintaining the ability to generate a summary within a length constraint. We also demonstrated the importance of the generation of summaries in a length constraint for real use. The post-edit evaluation with automatically generated summaries showed that in-length summaries contributed to approximately 30% to 40% improved post-editing time compared with use of the baselines.

References

Ayana, Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, and Mao-Song Sun. 2017. Recent Advances on Neural Headline Generation. *Journal of Computer Science and Technology*, pages 768–784.

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock. 2000. Headline Generation Based on Statistical Translation. In *Proceedings of ACL'00*, pages 318–325.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. 2013. Advances in Optimizing Recurrent Networks. In *Proceedings of ICASSP'13*.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *Proceedings of AAAI'18*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In *Proceedings of ACL'18*, pages 675–686.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of NAACL-HLT'16*, pages 93–98.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In *Proceedings of HLT-NAACL'03 on Text summarization workshop-Volume 5*, pages 1–8. Association for Computational Linguistics.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In *Proceedings of WNMT'18*, pages 45–54.

Thibault Fevry and Jason Phang. 2018. Unsupervised Sentence Compression using Denoising Auto-Encoders. In *Proceedings of the CoNLL'18*, pages 413–422. Association for Computational Linguistics.

Katja Filippova and Yasemin Altun. 2013. Overcoming the Lack of Parallel Data in Sentence Compression. In *Proceedings of EMNLP'13*, pages 1481–1491.

Katja Filippova and Michael Strube. 2008. Dependency Tree Based Sentence Compression. In *Proceedings of INLG'08*, INLG '08, pages 25–32.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. *CoRR*, abs/1705.03122.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of ACL'16*, pages 1631–1640.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. In *Proceedings of ACL'16*, pages 140–149.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of EMNLP'16*, pages 1328–1338.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Wenjie Li, Furu Wei, Sujian Li, and Ziqiang Cao. 2018. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. In *Proceedings of ACL'18*, pages 152–161.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries.

Yizhu Liu, Zhiyi Luo, and Kenny Zhu. 2018. Controlling Length in Abstractive Summarization Using a Convolutional Neural Network. In *Proceedings of EMNLP'18*, pages 4110–4119.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL'03*, pages 160–167.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *Proceedings of ICLR'18*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence Level Training with Recurrent Neural Networks. *CoRR*, abs/1511.06732.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of EMNLP'15*, pages 379–389.

Raphael Schumann. 2018. Unsupervised Abstractive Sentence Summarization using Length Controlled Variational Autoencoder. *CoRR*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of ACL'17*, pages 1073–1083.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *Proceedings of ACL'16*, pages 1683–1692.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a Next-Generation Open Source Framework for Deep Learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS'15*.

Oriol Vinyals and Quoc V. Le. 2015. A Neural Conversational Model. *CoRR*, abs/1506.05869.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title Generation with Quasi-synchronous Grammar. In *Proceedings of EMNLP'10*, pages 513–523.

Yuxiang Wu and Baotian Hu. 2018. Learning to Extract Coherent Summary via Deep Reinforcement Learning. *CoRR*.

David Zajic, Bonnie Dorr, and Richard Schwartz. 2004. BBN/UMD at DUC-2004: Topiary. *Proceedings of HLT-NAACL'04 Document Understanding Workshop*, pages 112 – 119.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. Efficient Summarization with Read-Again and Copy Mechanism. *CoRR*, abs/1611.03382.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective Encoding for Abstractive Sentence Summarization. In *Proceedings of ACL'17*, pages 1095–1104.