Controlling Length in Abstractive Summarization Using a Convolutional Neural Network

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

Abstract

Convolutional neural networks (CNNs) have met great success in abstractive summarization, but they cannot effectively generate summaries of desired lengths. Because generated summaries are used in difference scenarios which may have space or length constraints, the ability to control the summary length in abstractive summarization is an important problem. In this paper, we propose an approach to constrain the summary length by extending a convolutional sequence to sequence model. The results show that this approach generates high-quality summaries with user defined length, and outperforms the baselines consistently in terms of ROUGE score, length variations and semantic similarity.

1 Introduction

Great progress (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017) has been made recently on abstractive summarization. Many use sequence-to-sequence model based on RNN and attention mechanism (Rush et al., 2015), which was originally used for machine translation (Sutskever et al., 2014; Bahdanau et al., 2014). Recently, Gehring et al. (2017) proposed a convolutional sequence to sequence model equipped with Gated Linear Units (Dauphin et al., 2017), residual connections (He et al., 2016) and attention mechanism. Such a convolutional model achieves state-of-the-art accuracies in abstractive summarization on single sentence summarization, and it is much faster than the previous recurrent models as it can be easily parallelized. Furthermore, unlike recurrent models, the convoluational model has more stable gradients because of its backpropagation path.

Constraining summary length, while largely neglected in the past, is actually an important aspect of abstractive summarization. For example, given the same input document, if the summary is to be displayed on mobile devices, or within a fixed area of advertisement slot on a website, we may want to produce a much shorter summary. Unfortunately, most existing abstractive summarization models are not trained to react to summary length constraints. When the constraint is given at test time, the current practice is **i)** to truncate the generated summary after N tokens are generated when you want the summaries of length no more than N, and **ii)** ignore EOS (end of summary) token until the first M tokens are generated when you want the summaries of length at least M. Such a crude way of controlling summary length makes the output summary incomplete or incoherent.

Previous research on controlling length of abstractive summary has been scarce. Fan et al. (2017), who applies convolutional sequence to sequence model on multi-sentence summarization, converts length range as some special markers which are predefined and fixed. These markers are included in the training vocabulary. At training time, the model prepends the input of the summarizer with marker indicating the length of input sequence. At test time, it controls the length of the generated summary also by prepending length marker indicating the desired length. Unfortunately, this approach can not generate summaries of arbitrary lengths. It only generates summaries in predefined ranges of length, thus only meets the length constraints approximately. This is shown in Table 1. The above truncation practice can be used in conjunction with any of the length control methods but the excessive parts (red) will be truncated leaving incomplete sentences.

In our work, we extend the convolutional sequence to sequence model (Gehring et al., 2017) by controlling the length of summarization. Our approach seeks to generate summaries of any de-

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4110–4119
Brussels, Belgium, October 31 - November 4, 2018.

Table 1: Example summaries generated by different models with a desired length of 10 (red parts exceed the 10 token limit).

Reference summary (53 tokens)
david de gea and victor valdes enjoyed an afternoon off at a theme park . spanish duo donned shades as they made the most of the rare sunshine . it has certainly been a rollercoaster season for manchester united . united are third in the premier league after an impressive recent run .
Basic CNN summary (35 tokens)
david de gea and victor valdes made the most of the rare english sun with a trip to a theme park . david de gea and victor valdes enjoyed some fun in the sun .
(Fan et al., 2017) summary (30 tokens)
david de gea and victor valdes enjoyed a trip to a theme park . the pair enjoyed a relaxing time just days after united 's win against manchester city .
Our Length Control summary (LC) (10 tokens)
david de gea and victor valdes enjoy some fun .

sired number of tokens (also shown in Table 1). To do this, a length constraint is added to each convolutional block of the initial layer of the model. This information is propagated layer by layer during training. Our contributions are as follows:

1. We propose a simple but effective method to generate summaries with arbitrary desired length (Section 2.2).

2. Our approach outperforms the state-of-art baseline methods substantially by all evaluation metrics, i.e., ROUGE scores, length variation and semantic similarity (Section 3).

3. The generated summaries from our model are natural and complete, especially when the desired length is short (Section 3).

Next, we present the basic convolutional sequence to sequence model and our extension, followed by the evaluation of our approach and a discussion of related work.

2 Methodology

In this section, we will describe the model architecture used for our experiments and propose our length control method which is implemented by extending the basic model.

For summarization problems based on seq2seq model, given a sequence of tokens $\mathbf{x} = (x_1, x_2, ..., x_m)$ in the source document and a sequence of tokens $\mathbf{y} = (y_1, y_2, ..., y_n)$ in the target summary (i.e. $m > n$), the goal is to estimate the conditional probability $p(\mathbf{y}|\mathbf{x})$:

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

We aim at getting the above conditional probaability which can generate summaries with arbitrary desired length.

2.1 Basic CNN seq2seq Model

Our basic model consists of a multi-layer convolutional sequence to sequence model (CNN seq2seq)[1] (Gehring et al., 2017; LeCun et al., 1989) and an attention mechanism. Figure 1 illustrates the model.

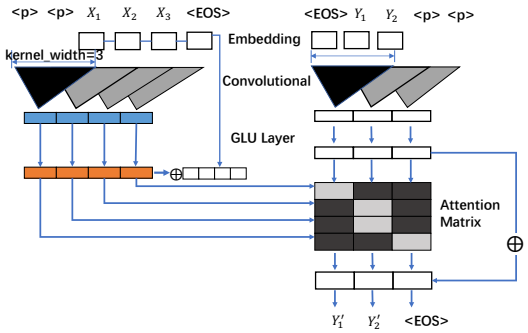


Figure 1: CNN seq2seq model

In the CNN seq2seq model, we obtain the input sequence $\mathbf{X} = (X_1, ..., X_m)$ and output sequence $\mathbf{Y} = (Y_1, ..., Y_n)$ after combining word vectors with their absolute positions in the document. We use $\mathbf{z} = (z_1^l, z_2^l, ..., z_m^l)$ and $\mathbf{h} = (h_1^l, h_2^l, ..., h_n^l)$ to denote the convolutional output of the encoder and decoder in l-th layer. Each element of the output sequence generated by the decoder network is fed back into the next layer of decoder network. Next, we add GLU (Dauphin et al., 2017) and residual connections (He et al., 2016) in each layer:

$$h_i^l = GLU(W^l[h_s^{l-1}, ..., h_t^{l-1}] + b^l) + h_i^{l-1} \quad (2)$$

where $[h_s^{l-1}, ..., h_t^{l-1}]$ corresponds to the h_i^l in the convolutional layers. The choice of s and t is based on kernel width and the padding method used to match the output of convolutional layers to the input length. We compute the probability distribution of generating the next elements y_{i+1} based on the current state and transform the top decoder output h_i^L via softmax:

$$p(y_{i+1}|y_1, ..., y_i, \mathbf{x}) = softmax(W_o h_i^L + b_o) \quad (3)$$

[1] https://github.com/facebookresearch/fairseq-py.

In addition, a multi-step attention mechanism that connects the encoder and decoder is used in each decoder layer. We define the decoder state d_i^l for attention as following:

$$d_i^l = W_d^l h_i^l + b_d^l + Y_i \quad (4)$$

The attention c_i^l is a weighted sum of the encoder outputs. The weights a_{ij}^l are based on the decoder states.

$$a_{ij}^l = \frac{\exp(d_i^l \cdot z_j^u)}{\sum_{t=1}^{m} \exp(d_i^l \cdot z_t^u)} \quad (5)$$

$$c_i^l = \sum_{j=1}^{m} a_{ij}^l (z_j^u + X_j) \quad (6)$$

At last, we add c_i^l to the current decoder elements h_i^l, which forms the final output or the input of the next layer in the decoder.

2.2 Modified Model with Length Control (LC)

We propose an approach which can control the summary length in CNN seq2seq model. The model can generate different summaries by setting desired length. It has the ability to generate the EOS tag at the appropriate time point in a natural manner.

To produce a summary of a given desired length, we modify the basic model by feeding the desired length as a parameter into the decoder of the CNN seq2seq model. At training time, we use the true length of the gold summary as the desired length. At test time, we can give any desired length len to the model and obtain a summary with length approximate to len. The modified decoder is shown in Figure 2.

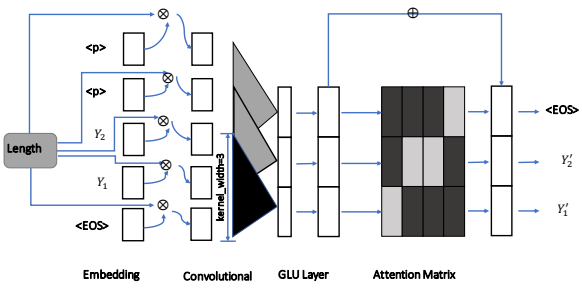


Figure 2: Modified Decoder

The CNN seq2seq model creates hierarchical structure over the input sequence. It is capable of capturing the correlation between elements over short distances at lower layers and between elements over long distances at higher layers. The useful information among the elements is aggregated after GLU. Therefore, we set the desired length as an input to the initial state of the decoder:

$$h_i^1 = v(W^1[h_s^0, ..., h_t^0] + b^1) + h_i^0 * len \quad (7)$$

where W is a trainable parameter, len is the desired length, v is GLU funciton and h_i^0 is the i-th element in the initial layer.

In the above function, we add length information at first layer in CNN model. GLU is like a gate. It can filter some information from a particular unit in each layer. The information attenuation occurs in GLU layer by layer. Different desired lengths have different degrees of information attenuation. Therefore the model is able to learn the probability of generating EOS with its own length information attenuation. This operation enables the model to produce a natural and complete summary for a given length constraint naturally.

3 Evaluation

In this section, our benchmark is the CNN/Daily Mail DMQA dataset (Hermann et al., 2015; Nallapati et al., 2016; See et al., 2017) [2], consisting 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. We follow the same pre-processing step used by See et al. (2017), and fill in the blanks with answer named entities. We show an example of such pairs in Table 4(a).

We compare our length constrained summarization model with the basic CNN seq2seq model and the state-of-the-art length controllable summarization model (Fan et al., 2017) [3]. Following Fan et al., we distribute the dataset into a set of disjoint buckets that correspond to summaries of different lengths. Each bucket contains roughly equal number of documents. The distribution is shown in Figure 3.

All competing methods have three flavors: *free*, *truncated* and *exact*. In the free version(Free), given the desired length N, each method generates summaries naturally until an EOS is generated. In the truncated version(Trunc), each method

[2] https://cs.nyu.edu/kcho/DMQA/

[3] All datasets, source code and generated summaries can be downloaded from http://202.120.38.146/sumlen.

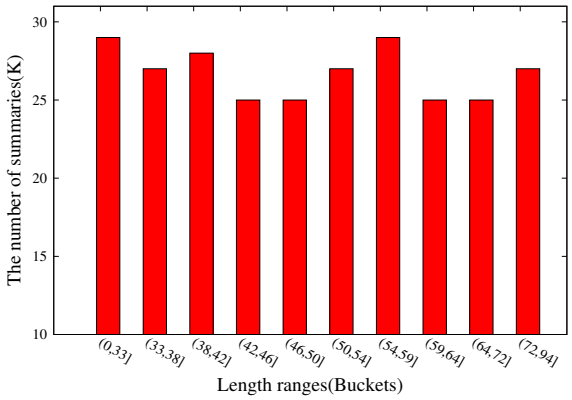


Figure 3: The buckets distribution of the dataset

artificially inserts an EOS if EOS has not been generated in the first N tokens. In the exact version(Exact), each method generates N non-EOS tokens by assigning a score of -∞ to the EOS and inserts an EOS after the N-th token. The purpose of Free version is to evaluate the method's ability to generate summaries with desired length; the purpose of the other two versions is to enable fair comparison of the summaries in terms of their content given that the summaries are of equal length.

3.1 Experimental Setup

In the following experiments, all the competing models have 8 convolutional layers in both encoder and decoder parts with kernel width as 3. For each convolutional layer, we set the hidden vector size as 512 and the embedding size as 256. To alleviate the overfitting problem, we add the dropout ($p = 0.2$) layer for all convolutional layers and fully connected layers.

To optimize the proposed model, we use Nesterov's accelerated gradient method (Sutskever et al., 2013) with gradient clipping 0.1 (Pascanu et al., 2013), momentum 0.99, and learning rate 0.2. We terminate the training process when the learning rate drops below $10e$-5. We set beam size as 5 for the beam search algorithm in the testing step. Next, we introduce the evaluation metrics in the following experiments:

1. *ROUGE* scores (F1 score) of the produced summaries, including ROUGE-1(R-1), ROUGE-2(R-2) and ROUGE-L(R-L) (Lin, 2004). ROUGE-2 is the most popular metric for summarization.

2. *Variance*(Var) of the summary lengths against the desired length len:

$$var = 0.001 * \frac{1}{n}\sum_{i=0}^{n}|l_i - len|^2, \quad (8)$$

where n is the number of pairs in the dataset, and l_i is the length of the generated summary i. We introduce the variance to evaluate the ability of exact control of the output length.

3. *Similarity*(Sim) between generated summaries and their corresponding reference summaries:

$$sim = \frac{1}{n}\sum_{i=0}^{n}\frac{y_i \cdot y_i'}{||y_i||||y_i'||} \quad (9)$$

where n is the number of pairs. y_i is the vector representation of the reference summary i and y_i' is vector of the corresponding generated summary i. Both y_i and y_i' are the sum of GloVe [4] word vectors of the words in these summaries.

We introduce the similarity metric here to complement the ROUGE scores because Yao et al. (2017a) showed that the standard ROUGE scores cannot capture semantic similarity beyond n-grams. Given the same source document, abstractive summarization may create summaries that don't share many words but mean the same. To show the effectiveness of this Sim metric, we design a dataset from the summarization tasks of TAC 2010$\sim$2011[5]. The TAC dataset consists of 90 topics in total, each with 2 subset. Each subset has 4 reference summaries by different humans. We assume reference summaries about the same topic to be semantically similar to each other, while summaries across topics are unrelated. Thus we created 2,160 pairs of similar summaries as postive data and 2,160 pairs of unrelated summaries as negative data. We then compute the Pearson correlation between the ROUGE score and the ground truth as well as between Sim and the ground truth and show the results in Table 2. Sim metric certainly resembles semantic similarity better than ROUGE by this experiment.

In this paper, we don't use manual evaluation as the major metric. The reason is that Lin (2004) showed that the manual evaluation is unstable and

[4]https://nlp.stanford.edu/projects/glove/.
[5]https://tac.nist.gov/

Table 2: Pearson correlation with the true semantic relatedness

	TAC 2010	TAC 2011
R-2	0.6107	0.6034
Sim	**0.6653**	**0.7165**

the inter-human agreement is low due to the variety in abstractive summaries. The ROUGE scores and Similarity scores can respectively measure the syntactic similarity and semantic similarity. They are complementary to each other and give better quantitative assessment of the summarization quality.

3.2 Experiment 1: Gold Summary Lengths

In the first experiment, for each test document-summary pair, we set the desired length as the length of the gold summary and ask the competing methods to generate a summary with the desired length. As shown in Table 3, the proposed model (LC) outperforms the other models on all of the evaluation metrics. The ROUGE score shows the accuracy of these models. Lower variance reflects better length control of the model. Higher similarity reflects better quality of generated summaries from the semantic point of view.

Table 3: Desired Length: Gold Summary Lengths

		R-1	R-2	R-L	Var	Sim
Free	CNN	34.49	14.38	25.78	0.3465	0.9220
	Fan	34.53	14.40	25.78	0.3446	0.9216
	LC	**35.45**	**14.50**	**26.02**	**0.0005**	**0.9272**
Trunc	CNN	34.76	**14.53**	26.00	0.3045	0.9201
	Fan	34.74	14.52	25.97	0.3031	0.9197
	LC	**35.44**	14.48	**26.02**	**0.0002**	**0.9268**
Exact	CNN	35.39	14.43	**26.07**	0.0	0.9249
	Fan	35.37	14.42	26.03	0.0	0.9246
	LC	**35.44**	**14.50**	26.02	0.0	**0.9268**

The LC model achieves the highest ROUGE and similarity scores as well as the lowest variance in both Free and Exact version, which shows the effectiveness of LC for generating high quality summaries under length constraint. In the Trunc version, the LC model outperforms the other comparable models on all evaluation metrics except for the ROUGE score. Note that, the ROUGE scores of LC model are very stable, indicating its effective length control. As for the other two models, they have better ROUGE score on Trunc version. However, as the example shown in Table 4[6], higher ROUGE scores do not necessarily mean high quality abstractive summaries.

The ROUGE score consists of Recall(R), Precision(P) and F1-measure(F). The summary tends to achieve a better ROUGE score when the length of generated summary is slightly shorter than the desired length. In Table 4(b), the CNN model has the same R score as LC model and a higher P score than LC model because of its slightly shorter length. We can see that the CNN model achieve a higher F score even its generated summary is not good. Moreover, for the basic model, the generated summary always repeats the sentences when the length of generated summary is longer than the desired length. In Table 4(c), the P score of its Trunc version would be improved by a large margin. Thus, the ROUGE score for the Trunc version biases toward the models with weak length control. The generated summaries of the LC model in Table 4(d), which capture the semantic of the reference summary and satisfy the constraint length very well, are better than the other two models even with a slightly lower ROUGE score. The topic of this example is that Louis Jordan, who is the son of Frank Jordan, got lost during sailing and was finally rescued from his boat. Our model generates the summary with correct information, but other two models get the Louis Jordan and Frank Jordan mixed up. This is correctly measured by the similarity scores.

3.3 Experiment 2: Arbitrary Lengths

In the second experiment, we ask the methods to generate summaries with arbitrary lengths. We report the results of all three methods with five arbitrary lengths: 10, 30, 50, 70 and 90. We show the performance of each model with different length constraints in Table 5, Table 6, Figure 4 and Figure 5. The basic CNN model has the same ROUGE scores in the Free version since it cannot control the length of generated summaries on its own. For Fan et al. (2017), the desired length is mapped to the model's predefined fixed length range(s) that *contains* the desired length before it produces its summaries. For example, the desired length 10 is mapped to the first bucket $(0, 33]$.

To demonstrate the effectiveness of LC model and further illustrate the results, we show an example of generated summaries by LC(Free) model with different lengths. As shown in Table 7, when the desired length (e.g., 10) is very different from the length of the reference summary, the ROUGE

[6]The entities in different color indicate two important roles in the text. The words in bold type mean correct content.

Table 4: Example summaries generated in Experiment 1.

(a) Source document and reference summary (36 tokens)

Source document
the last time frank jordan spoke with his son, louis jordan was fishing on **a sailboat a few miles off the south carolina coast**. the next time ... more than two months had passed and the younger jordan was on a contrainer ship 200 miles from north carolina, just **rescued from his disabled boat** . "**i thought i lost you**,"the relieved father said. louis jordan, 37, took his sailboat out in **late january** and **hadn't been heard from in 66 days** ... the younger jordan said **he took his sailboat out to the gulf stream to find some better fishing** ... the boat capsized two more times before he was **rescued**, according to jordan.
Reference summary
louis jordan says his sailboat capsized three times . he **survived** by collecting rainwater and eating raw fish . frank jordan told cnn **his son** is n't an experienced sailor but has a strong will .

(b) Free summary(29 tokens), Trunc summary(29 tokens) and Exact summary of CNN

		Summary	R	P	F	Var	Sim
CNN	Free	frank jordan **took his sailboat out to the gulf stream to find some better fishing** , jordan says . " it took so long , " jordan says .	6.06	9.09	7.27	0.049	0.9217
	Trunc	frank jordan **took his sailboat out to the gulf stream to find some better fishing** , jordan says . " it took so long , " jordan says .	6.06	**9.09**	7.27	0.049	0.9217
	Exact	frank jordan **took his sailboat out to the gulf stream to find some better fishing** , jordan says . jordan says he took his sailboat out to the gulf stream to find some better fishing .	6.06	**6.25**	6.15	-	0.9254

(c) Free summary(50 tokens), Trunc summary(36 tokens) and Exact summary of Fan

		Summary	R	P	F	Var	Sim
Fan	Free	frank jordan **took his sailboat out to the gulf stream to find some better fishing .** jordan says he took his sailboat out to the gulf stream to find some better fishing . jordan says he took his sailboat out to the gulf stream to find some better fishing .	6.06	**4.35**	5.06	0.196	0.9215
	Trunc	frank jordan took his sailboat out to the gulf stream to find some better fishing . **his son** , louis jordan **took his sailboat out to the gulf stream to find some better fishing .**	12.12	**12.90**	12.50	0.0	0.9194
	Exact	frank jordan took his sailboat out to the gulf stream to find some better fishing . **his son** , louis jordan **took his sailboat out to the** gulf stream to find some **better fishing .**	12.12	12.90	12.50	-	0.9194

(d) Free summary(36 tokens), Trunc summary(36 tokens) and Exact summary of LC(ours)

		Summary	R	P	F	Var	Sim
LC	Free	louis jordan was on **a sailboat a few miles off the south carolina coast** . he **had n't been heard from in 66 days** when he was **rescued** . he was **rescued from his boat** .	6.06	6.06	6.06	0.0	**0.9293**
	Trunc	louis jordan was on **a sailboat a few miles off the south carolina coast** . he **had n't been heard from in 66 days** when he was **rescued** . he was **rescued from his boat** .	6.06	6.06	6.06	0.0	**0.9293**
	Exact	louis jordan was on **a sailboat a few miles off the south carolina coast** . he **had n't been heard from in 66 days** when he was **rescued** . he was **rescued from his boat** .	6.06	6.06	6.06	-	**0.9293**

score may not be good even though the generated summary matches the reference quite well semantically. The generated summaries from LC model are *natural* and *complete*. The summaries with short desired length on Trunc and Exact version would be more vulnerable to the incomplete problem. We randomly sample 100 summaries generated by each model under Trunc and Exact with desired length of 10 and 30, and manually inspect their readibility. This is a simplified human-evaluation of summarization, which just determines whether the sentences in summaries under length control are complete or not. If complete, the score is 1; if not, it is 0. It is easier to accomplish and more reliable than other sophisticated human-evaluation. Table 8 shows that the LC model has a clear advantage over the other two models in terms of summary fluency.

In this experiment, the desired length is fixed for all the documents which is independent from the corresponding lengths of reference summaries such that the generated summaries may include more versatile words and phrases different from the reference summaries. Thus, the similar-

Table 5: Desired Length: 10, 30, 50, 70, 90

(a) **Free** version

Free	10			30			50			70			90		
	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC
R-1	**34.49**	34.28	19.03	**34.49**	34.28	32.26	34.49	34.60	**34.71**	34.49	**34.65**	33.83	**34.49**	30.56	32.17
R-2	**14.38**	14.18	8.45	**14.38**	14.18	13.60	14.38	**14.41**	14.24	14.38	**14.50**	13.67	**14.38**	12.20	13.00
R-L	**25.78**	25.60	16.47	**25.78**	25.60	24.64	25.78	**25.79**	25.62	25.78	**25.82**	24.67	**25.78**	22.08	23.28

(b) **Trunc** version

Trunc	10			30			50			70			90		
	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC
R-1	**20.14**	20.12	18.77	32.96	**32.99**	32.25	35.14	35.07	**35.60**	34.49	**34.67**	33.83	31.27	**34.70**	32.16
R-2	**9.27**	9.22	8.31	**14.06**	14.04	13.60	**14.46**	14.40	14.30	14.38	**14.50**	13.67	12.40	**14.55**	13.00
R-L	**17.35**	17.34	16.28	**25.11**	25.06	24.62	**26.09**	26.05	25.90	25.78	**25.82**	24.67	22.69	**25.86**	23.29

(c) **Exact** version

Exact	10			30			50			70			90		
	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC	CNN	Fan	LC
R-1	**20.14**	20.14	20.06	**33.05**	32.83	32.94	34.71	34.72	**34.81**	32.24	33.35	**33.82**	31.27	31.37	**32.04**
R-2	**9.27**	9.23	9.23	**14.08**	13.79	14.00	**14.78**	14.17	14.23	13.33	13.39	**13.59**	12.41	12.47	**12.86**
R-L	**17.36**	17.36	17.30	**25.15**	24.87	25.02	**25.65**	25.63	25.60	24.31	24.35	**24.56**	22.69	22.76	**23.14**

Table 6: Similarity of different length

(a) **Free** version

	CNN	Fan	LC
10	**0.9220**	0.9205	0.8124
30	**0.9220**	0.9214	0.9092
50	0.9220	0.9216	**0.9263**
70	0.9220	0.9222	**0.9323**
90	0.9220	0.9234	**0.9256**

(b) **Trunc** version

	CNN	Fan	LC
10	0.7966	0.7968	**0.8003**
30	0.9079	0.9080	**0.9085**
50	0.9236	0.9231	**0.9286**
70	0.9219	0.9222	**0.9323**
90	0.9325	0.9329	**0.9353**

(c) **Exact** version

	CNN	Fan	LC
10	0.7968	0.7961	**0.7975**
30	0.9083	0.9073	**0.9090**
50	0.9248	0.9245	**0.9251**
70	0.9299	0.9230	**0.9320**
90	0.9325	0.9327	**0.9347**

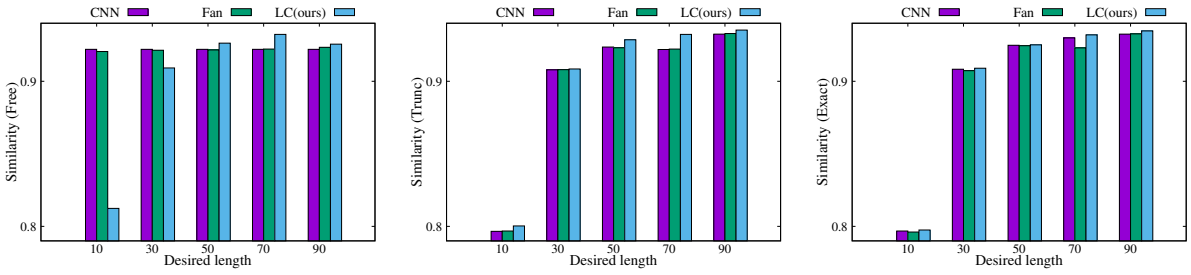
Table 7: Generated summaries of LC (Free) model

10	the younger jordan was **rescued from his disabled boat** .
30	louis jordan was **rescued from his disabled** . he **boat had n't been heard from in 66 days in late january** . he was rescued from his disabled boat .
50	" **i thought i lost you** , " jordan says . the younger jordan was on **a sailboat a few miles off the south carolina coast** . " i thought i lost you , " jordan tells **his son** . jordan says he was grateful to the people .

Table 8: The proportion of summaries that are natural and complete with desired length 10 and 30

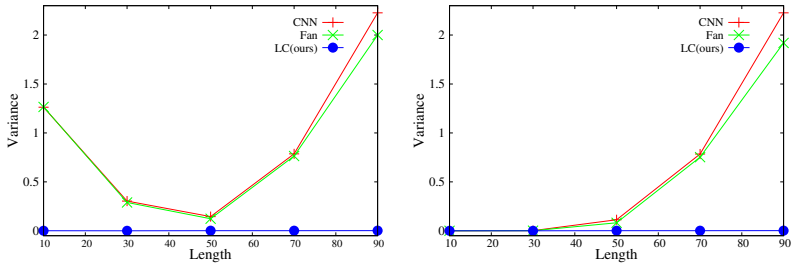
	Trunc		Exact	
	10	30	10	30
CNN	0.41	0.37	0.48	0.47
Fan	0.50	0.42	0.53	0.57
LC	**0.62**	**0.59**	**0.88**	**0.86**

ity score is more reasonable for evaluation than ROUGE score. As shown in Table 6 and Figure 4, the LC model achieves the highest similarity score except for the length of 10 and 30 in the Free version. The reason is that there is only 5% of testing data with the length of reference summary shorter than 30. Due to the effective length control of LC model, the lengths of generated summaries from LC model are usually much shorter than those from the other models and the length of corresponding reference summaries when we set the desired length as 10 or 30. This leads to a relative lower similarity score shown in Figure 4(b) and Figure 4(c). As shown in Figure 5, the LC model achieves the lowest variance. In Figure 5(a), as the length of most summaries is around 50 and the number of summaries with a length of 10 or 90 is small, the CNN model and Fan model has lowest variance at 50 and highest variance at 90. In Figure 5(b), the length of generated summaries in Trunc version is no more than desired length. So the variances of CNN model and Fan model are incremental. Besides, we can find that the LC model is stable under all conditions because of its effective length control model.



(a) Similarity in Free version (b) Similarity in Trunc version (c) Similarity in Exact version

Figure 4: Similarity of different length



(a) Variance in Free version (b) Variance in Trunc version

Figure 5: Variance of different length

3.4 Significance Test on Similarity Result

We use significance test to prove that similarity metric is reliable even though the numerical difference of similarity scores in experiment is little. Because the similarity scores of generated summaries do not follow normal distribution, we take Kruskal-Wallis test (Loukina et al., 2014; Albert, 2017) as our significance test to measure that the difference of similarity results of three methods is significant or not. As shown in Table 9, all p-values are less than 0.05. The smaller p-value, the higher significant. Thus, the difference of the similarity results is significant.

Table 9: p-value of significance test

	Free	Trunc	Exact
Exp.1	3.4e-32	2.12e-45	0.01
Exp.2	0.0	4.6e-39	1.0e-4

4 Related Work

In this section, we discuss some previous work on length control in abstractive summarization and explain why we choose CNN as our basic summarization model.

4.1 Length Control for Abstractive Summarization

When summarizing a document, it is desirable to be able to control the length of summary so as to cater to different users and scenarios. Most abstractive summarization systems are based on encoder-decoder models and generate summaries whose length depends on the training summaries. Due to the variability of the sequence generation models, such as the different structures and functions, it is hard to design a length constraint method on all summarization models.

Previous methods control summary length by generating EOS token at a particular time. Rush et al. (2015) used an ad-hoc method, in which the system is inhibited from generating the EOS tag by assigning a score of -∞ to the tag and generats a fixed number of words. Kikuchi et al. (2016) proposed two different methods for RNN seq2seq model which can control the summary length by taking length embedding as an additional input for the LSTM and adding desired length into initial memory cell for the LSTM. In this model, they use the Gigawords as dataset and focus on the abstractive summarization in sentence level which generates one sentence as the summary. For CNN seq2seq model, Fan et al. (2017) put some spe-

cial markers into the vocabulary which denote different length ranges. It prepends the input of the summarizer with the marker during training and testing. These special markers are predefined and fixed. In this paper, we aim at generating complete summaries with arbitrary desired length naturally for CNN seq2seq model. We use multi-layers CNN seq2seq model on both encoder and decoder. We set the length constraint at the first layer of decoder to implement the length control of the summarization. Compared with other methods, our approach can effectively control the length of generated summary in a natural manner. Meanwhile, it can generate summaries with length approximate to the desired length without semantic losing in less time.

4.2 Encoder-Decoder for Abstractive Summarization

Automatic document summarization generates short summaries for original documents. A summary should cover the key topics of the original document(s). A good summary should be coherent, non-redundant and readable(Yao et al., 2017b). The research in abstractive summarization with encoder-decoder model (Sutskever et al., 2014; Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017; Fan et al., 2017) has made some progress.

Most of them use RNN with different attention mechanisms (Nallapati et al., 2016; See et al., 2017; Paulus et al., 2017). Rush et al. (2015) used RNN with soft-attention, while Paulus et al. (2017) used the RNN with intra-attention. Recently, research on CNN based summarization has gained momentum. Gehring et al. (2017) proposed the CNN seq2seq model with multi-step attention, which was extended in (Fan et al., 2017). Bai et al. (2018) showed that CNN is more powerful than RNN for sequence modeling. What's more, CNN enables much faster training and more stable gradients than RNN. Therefore we select CNN seq2seq model as our basic model and do not compare our model with RNN seq2seq model.

5 Conclusion

We presented a simple approach to modify existing CNN seq2seq model with a summary length input and were able to train a model that produces summaries of desired length that are fluent and coherent. This is a better solution than the current practice of summary truncation. Compared with the existing summarization methods, we show that our model has the ability to control the output length on its own using its internal state without losing semantic information or sacrificing the ROUGE score.

Acknowledgment

Kenny Q. Zhu is the contact author and was supported by NSFC grants 91646205 and 61373031. Thanks to the anonymous reviewers for their valuable feedback.

References

Corbin Albert. 2017. Exploring teacher forcing techniques for sequence-to-sequence abstractive headline summarization.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 93–98.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 933–941.

Angela Fan, David Grangier, and Michael Auli. 2017. Controllable abstractive summarization. *arXiv preprint arXiv:1711.05217*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1243–1252.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

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, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

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, Austin, Texas, USA, November 1-4, 2016*, pages 1328–1338.

Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 396–404.

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

Anastassia Loukina, Klaus Zechner, and Lei Chen. 2014. Automatic evaluation of spoken summaries: the case of language assessment. In *Proceedings of the ninth workshop on innovative use of NLP for building educational applications*.

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, Berlin, Germany, August 11-12, 2016*, pages 280–290.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304.

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, Lisbon, Portugal, September 17-21, 2015*, pages 379–389.

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, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1139–1147.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Jin-Ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017a. Recent advances in document summarization. *Knowl. Inf. Syst.*

Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. 2017b. Recent advances in document summarization. *Knowl. Inf. Syst.*, 53(2):297–336.