Adversarial Reinforcement Learning for Chinese Text Summarization

Hao Xu[1,2], Yanan Cao[2], Yanmin Shang[2(✉)], Yanbing Liu[2], Jianlong Tan[2], and Li Guo[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{xuhao2, caoyanan, shangyanmin, liuyanbing, tanjianlong, guoli}@iie.ac.cn

Abstract. This paper proposes a novel *Adversarial Reinforcement Learning* architecture for Chinese text summarization. Previous abstractive methods commonly use Maximum Likelihood Estimation (MLE) to optimize the generative models, which makes auto-generated summary less incoherent and inaccuracy. To address this problem, we innovatively apply the *Adversarial Reinforcement Learning* strategy to narrow the gap between the generated summary and the human summary. In our model, we use a generator to generate summaries, a discriminator to distinguish between generated summaries and real ones, and reinforcement learning (RL) strategy to iteratively evolve the generator. Besides, in order to better tackle Chinese text summarization, we use a character-level model rather than a word-level one and append Text-Attention in the generator. Experiments were run on two Chinese corpora, respectively consisting of long documents and short texts. Experimental Results showed that our model significantly outperforms previous deep learning models on rouge score.

Keywords: Chinese text summarization · Generative adversarial network
Deep learning · Reinforcement learning

1 Introduction

With the rapid growth of the online information services, more and more data is available and accessible online. This explosion of information has resulted in a well-recognized information overload problem [1]. However, the time-cost is expensive if you want to get key information from a mass of data in an artificial way. So, it is very meaningful to build an effective automatic text summarization system which aims to automatically produce short and well-organized summaries of documents [2]. While *extractive approaches* focus on selecting representative segments directly from original text [3, 4], we aim to capture its salient idea by understanding the source text entirely, i.e. using an *abstractive approach*.

Y. Shi et al. (Eds.): ICCS 2018, LNCS 10862, pp. 519–532, 2018.
https://doi.org/10.1007/978-3-319-93713-7_47

Most recent abstractive approaches apply a sequence-to-sequence (seq2seq) framework to generate summaries and use Maximum Likelihood Estimation (MLE) to optimize the models [8, 9]. The typical seq2seq model consists of two neural networks: one for encoding the input sequence into a fixed length vector C, and another for decoding C and outputting the predicted sequence. The state-of-the-art seq2seq method uses attention mechanism to make the decoder focus on a part of the vector C selectively for connecting the target sequence with each token in the source one.

Despite the remarkable progress of previous research, Chinese text summarization still faces several challenges: (i) As mentioned above, the standard seq2seq models use MLE, i.e. maximizing the probability of the next word in summary, to optimize the objective function. Such an objective does not guarantee the generated summaries to be as natural and accurate as ground-truth ones. (ii) Different from English, the error-rate of word segmentation and the larger vocabulary in Chinese call for character-level models. Character-level summarization depends on the global contextual information of the original text. However, the decoder with attention mechanism which performed well in other natural language processing (NLP) tasks [5] just pay attention to the key parts of text.

To address these problems, we propose a novel *Adversarial Reinforcement Learning* architecture for Chinese text summarization, aiming to minimize the gap between the generated summary and the human summary. This framework consists of two models: a summary generator and an adversarial discriminator. The summary generator based on a seq2seq model is treated as an agent of reinforcement learning (RL); the state is the generated tokens so far and the action is the next token to be generated; the discriminator evaluates the generated summary and feedback the evaluation as reward to guide the learning of the generative model. In this learning process, the generated summary is evaluated by its ability to cheat the discriminator into believing that it is a human summary. Beyond the basic ARL model, in order to well capture the global contextual information of the source Chinese text, the generator introduces the text attention mechanism based on the standard seq2seq framework.

We conduct the experiments on two standard Chinese corpora, namely LCSTS (a long text corpus) and NLPCC (a short text corpus). Experiments show that our proposed model achieves better performance than the state-of-the-art systems on two corpora.

The main contributions of this paper are as follows:

- We propose a novel deep learning architecture with *Adversarial Reinforcement Learning* framework for Chinese text summarization. In this architecture, we employ a discriminator as an evaluator to teach the summary generator to generate more realistic summary.
- We introduce the attention mechanism in the source text on the intuition that the given text provides a valid context for the summary, which makes character-level summarization more accurate.

2 Related Work

Traditional abstractive works include unsupervised topic detection method, phrase-table based machine translation approaches [6], and Generative Adversarial Network approaches [7]. In recent years, more and more works employ deep neural network framework to tackle abstractive summarization problem. [8] were the first to apply seq2seq to English text summarization, achieving state-of-the-art performance on two sentence-level summarization datasets DUC-2004 and Gigaword. [13] improved this system by using encoder-decoder LSTM with attention and bidirectional neural net. Attention mechanism append to the decoder allows it to look back at parts of the encoded input sequence while the output is generated and gain better performance. [14] constructs a large-scale Chinese short text summarization dataset from the microblogging website Sina Weibo. And as far as we know, they made the first attempt to perform the seq2seq approach on a large-scale Chinese corpus, which is based on GRU encoder and decoder. In above works, the most commonly used training objective is Maximum Likelihood Estimation (MLE). However, maximizing the probability of generated summary conditioned on the source text is far from minimizing the gap between generated and human summary. This discrepancy between training and inference makes generated summaries less coherent and accuracy.

Different from MLE, reinforcement learning (RL) is a computational approach to learning whereby an agent tries to maximize the total amount of reward it receives when interacting with a complex, uncertain environment [15]. [16] proved RL methods can be adapted to text summarization problems naturally and simply on the premise of effectively selecting features and the score function.

Meanwhile, the idea of generative adversarial network (GAN) has got a huge success in computer vision [11, 12]. The adversarial training is formalized as a game between two networks: a generator network (G) to generate data, a discriminator network (D) to distinguish whether a given summary is a real one. However, discrete words are nondifferentiable and cannot provide a gradient to feed the discriminator reward back to the generator. To address this problem, Sequence Adversarial Nets with Policy Gradient (SeqGAN) [17] used the policy network as a generator, which enables the use of the adversarial network in NLP. [18] proposes to adversarial in hidden vectors of the generator rather than the output sequence.

Inspired by the successful application of RL and GAN in related tasks, we propose adversarial reinforcement learning framework for text summarization. And a discriminator is introduced as the adaptive score function. We use the discriminator as the environment or human, and output from discriminator as a reward. The updating direction of generator parameters can be obtained by using the policy gradient.

3 Adversarial Reinforcement Learning

The overall framework of our model is shown in Fig. 1. A given text sequence is denoted as $X = \{x_1, x_2, \ldots, x_n\}$ consisting of n words, where x_i is the i-th word. Summary generated by human (shown in the yellow box) is denoted as $Y = \{y_1, y_2, \ldots, y_m\}$, where y_j is the j-th word and $m < n$. The goal of this model is to

generate a summary $Y' = \{y'_1, y'_2, \ldots, y'_{m'}\}$ consisting of m' words, where $m' < n$ and m maybe not equal to m'.

The *adversarial reinforcement learning* framework consists of two models: a generative model G and a discriminative model D. We use G (shown in the green box) to transform the original text X into summary Y' based on a seq2seq framework. Here, we want to make the distribution of Y' and Y overlap as much as possible. To achieve this goal, we use D (shown in the red box) based on recursive neural networks (RNN). We take the same amount of positive samples $(X, Y) \sim \mathrm{P}_r$ and navigate samples $(X, Y') \sim \mathrm{P}_g$ randomly to train the D, where P_r means the joint distribution of source text and real summary, and P_g means that of source text and generated summary. Meanwhile, we use strategy gradient to train G according to the reward by D.

3.1 Summary Generator

Seq2seq Model. Most recent models for text summarization and text simplification are based on the seq2seq model. In the previous work [8, 9], the encoder is a four layer Long Short-term Memory Network (LSTM) [19], which maps source texts into the hidden vector. The decoder is another LSTM, mapping from i–1 words of Y' and X to y'_i, which is formalized as $y_i \sim G(Y'_{1:i-1}|X_{1:n})$, where $Y'_{1:i}$ means the generated summary at the i-th step.

Attention mechanism is introduced to help the decoder to "attend" to different parts of the source sentence at each step of the output generation [8]. We redefine a conditional probability for seq2seq in the following:

$$G(y'_i|Y'_{1\ldots i-1}, X) = g(y'_{i-1}, s_i, c_i) \tag{1}$$

Where s_i is the hidden status unit in the decoder, and c_i is the context vector at step i. For standard LSTM decoder, at each step i, the hidden status s_i is a function of the previous step status s_{i-1}, the previous step output y'_{i-1}, and the i-th context vector:

$$s_i = f(s_{i-1}, y'_{i-1}, c_i) \tag{2}$$

$$c_i = \sum\nolimits_{j=1}^{n} \alpha_{ij} h_j \tag{3}$$

The weight α is defined as follows:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \tag{4}$$

α_{ij} is called the alignment model, which evaluates the matching degree of the j-th word of text and the i-th word of summary.

Text-Attention. Different from the sequence transformation problem, text summarization is a mapping from original space to subspace. So, summarization models should pay attention to potential key information in the source text. From another

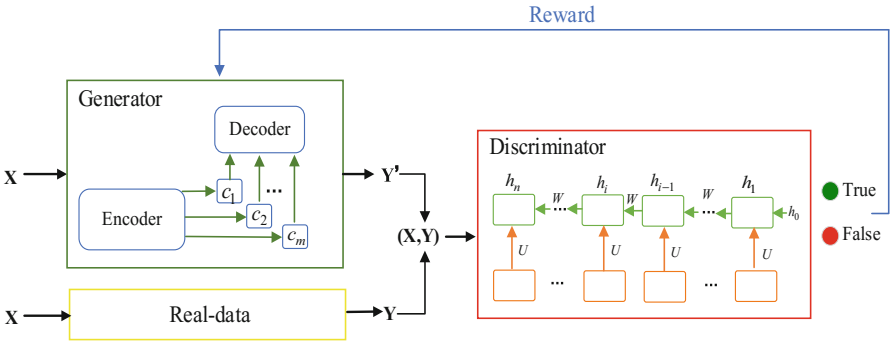

Fig. 1. Architecture of adversarial reinforcement learning for text summarization (Color figure online)

perspective, information needed by a partial summary, may be located anywhere in the source text. So, the attention should be anywhere around the text if needed. However, decoder with attention merely focuses on the latest context of the next decoded word.

As shown in Fig. 2, we introduce the Text-Attention based on IARNN-WORD [20]. In such a framework, we use attention mechanism on X, because the contextual information of X is very effective for the generated summary Y'. In order to well utilize the relevant contexts, we use attention before feeding X into the RNN model, which is formalized as follows:

$$\beta_i = \sigma(r_t m_{ti} x_i) \tag{5}$$

$$\tilde{x}_i = \beta_i * x_i \tag{6}$$

where m_{ti} is an attention matrix which transforms a text representation r_t into a word embedding representation, and β_i is a scaler between 0 and 1.

3.2 Adversarial Discriminator

The *discriminator*, called D for short, is used to distinguish generated summary from real as much as possible. This is a typical problem of binary classification. We use RNN model to capture text contextual information which is very effective for text classification, and the final layer is a 2-class softmax layer which gives the label 'Generated' or 'Real'. The framework of D is shown in Fig. 1. In order to prevent collapse mode, we use mini-batch method to train D. We sampled the same number of text-summary pairs (X, Y) and (X, Y') respectively from human and generator, where $Y' \sim G(\cdot|X)$ and the mini-batch size is k. For each text-summary pair (X_i, Y_i) sent to D, the optimization target is to minimize the cross-entropy loss for binary classification, using human summary as positive instance and generated summary as negative one.

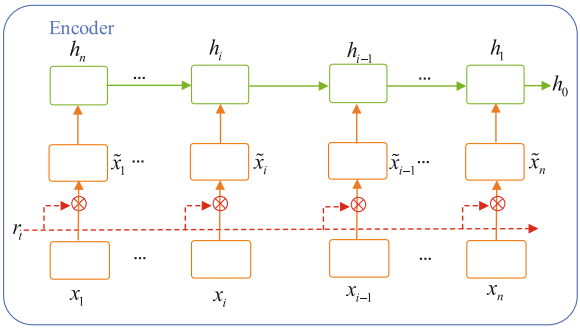

Fig. 2. Text-attention

3.3 Strategy Gradient for Training

Our goal is to encourage the generator to generate summaries that make the discriminator difficult to distinguish them from real ones. G is trained by policy gradient and reward signal is passed from D via Monte Carlo search. To be more precise, there is generally a markov decision process, performing an action y_i based on the state s_i with $Reward(s_i, y_i)$, where s_i denotes the decoding result of the previous i–1 words Y_{i-1}. A series of performed actions are called a "strategy" or "strategy path" θ^{π}. The target of RL is to find out the optimal strategy which can earn the biggest prize:

$$\theta^{\pi}_{best} = arg\max_{\theta^{\pi}} \sum_{A_i \in \theta^{\pi}_{best}}^{i} \text{Re}ward(s_i, y_i) \tag{7}$$

RL can evaluate each possible action in any state through the environment feedback of reward and find out one action to maximize the expected reward $E(\sum_{y_i \in \theta^{\pi}}^{i} Reward(s_i, y_i), \theta^{\pi})$. Based on this, we assume that the generated summary is rewarded from the real summary by D, denoted as $R(X, Y')$. We denote parameters in the framework of encoder-decoder as θ, then our objective function is expressed as maximizing the expected reward of generated summary based on RL:

$$\begin{aligned} \theta^{\pi}_{best} &= arg \max_{\theta} \mathbb{E}(R(X, Y')) \\ &= arg \max_{\theta} \sum_{X} \sum_{Y'} P_{\theta}(X, Y') R(X, Y') \\ &= arg \max_{\theta} \sum_{X} P(X) \sum_{Y'} P_{\theta}(Y'|X) R(X, Y') \end{aligned} \tag{8}$$

Where $P_{\theta}(X, Y')$ denotes the joint probability of a text-summary pair (X, Y') under the parameter θ. We redefine he right-hand side of Eq. (8) as J_{θ}, which is the expectation of reward when G gets the optimal parameter. The probability distribution of each text-summary pair (X_i, Y'_i) can be regarded as a uniform distribution:

$$J_\theta = \sum_X P(X) \sum_{Y'} P_\theta(Y'|X) R(X, Y') \approx \frac{1}{n} \sum_{i=1}^{n} R(X_i, Y_i') \tag{9}$$

Whose gradient w.r.t. is:

$$\begin{aligned} \nabla J_\theta &= \sum\nolimits_X P(X) \sum\nolimits_{Y'} R(X, Y') \nabla P_\theta(Y'|X) \\ &= \sum\nolimits_X P(X) \sum\nolimits_{Y'} P_\theta(Y'|X) \frac{\nabla P_\theta(Y'|X)}{P_\theta(Y'|X)} \\ &= \sum\nolimits_X P(X) \sum\nolimits_{Y'} R(X, Y') P_\theta(Y'|X) \nabla \log P_\theta(Y'|X) \\ &\approx \frac{1}{n} \sum_{i=1}^{n} R(X_i, Y_i') \nabla \log P_\theta(Y_i'|X_i) \end{aligned} \tag{10}$$

So, in this case, our optimization goal is to maximize the probability of generating a summary. That is, by updating the parameters θ, the reward will make the model improve the probability of the occurrence of the high-quality summary, while the punishment will make the model reduce the probability of the occurrence of the inferior summary. Therefore, we can use the reinforcement learning to solve the problem of GAN cannot differentiable in discrete space.

However, in all cases, mode-collapse will appear during the game. So, we adopted monte carol search to solve this problem. To be specific, when $t \neq n$, the decoding result is just a partial one whose reward is $D(X_i, Y_{i:t}')$. We use monte carol search to supplement its subsequent sequence, calculating the mean of all possible rewards.

We use the D as the reward for RL and assume the length of generated summary is m'. Then the calculation of the reward value J_θ of the generated summary is as follows:

$$J_\theta = \frac{1}{n} \sum_{i=1}^{n} D(X_i, Y_{1:i-1}' + y_i') \tag{11}$$

Where $Y_{1:i-1}'$ denotes the previously generated summary. Then we can have n path to get n sentences by Monte Carlo search. The *discriminator* D can give a reward for the generated summary in the whole sentence.

When updating model parameters θ, if the reward is always positive, the samples cannot cover all situations. So, we need use the baseline setting to reward. The gradient after joining the baseline is:

$$\nabla J_\theta \approx \frac{1}{n} \sum_{i=1}^{n} D(X_i, Y_i') \nabla \log P_\theta(Y_i'|X_i) \tag{12}$$

Equation (12) is a reward of the probability of generated summary. Unfortunately, the probability value is non-negative, which means that the *discriminator* doesn't give negative penalty term, no matter how bad the generated summary is. This will cause the generator to be unable to train effectively. Therefore, we introduced the basic value *baseline*. When we calculate the reward, we minus this *baseline* from the feedback of

reward. The basic value of the reward and punishment is b, and the calculation formula of the optimization gradient in Eq. (12) is modified as follows:

$$\nabla J_\theta \approx \frac{1}{n}\sum_{i=1}^{n}(D(X_i, Y_i') - b)\nabla \log P_\theta(Y_i'|X_i) \tag{13}$$

G and *D* are interactive training. When we train the generator, the *G* continuously optimizes itself by the feedback of *D*. The gradient approximation is used to update θ, where α denotes the learning-rate:

$$\theta^{i+1} = \theta^i + \alpha \nabla J_{\theta^i} \tag{14}$$

It's time to update the new *D* until the generated summary is indistinguishable.

As a result, the key to gradient optimization is to calculate probability of generated summary. So, as the model parameter updates, our model will gradually improve the summary and reduce the loss. The expectation of the reward is an approximation of a sample.

To sum up, our target is to approximate the distribution of generated summary to the that of real ones in a high-dimensional space. Our model works like a teacher, and the *Discriminator* directs the *Generator* to generate natural summaries. In the perfect case, the distribution of generated summaries and that of real ones will overlap completely.

4 Experiments and Results

4.1 Datasets and Evaluation Metric

We train and evaluate our framework on two datasets, one consists of short texts (on average 320 characters) and the other is long (840 characters). The short text corpus is Large Scale Chinese Short Text Summarization Dataset (LCSTS) [14], which consists of more than 2.4 million text-summary pairs, constructed from the Chinese microblogging website Sina Weibo[1]. It is split into three parts, with 2,400,591 pairs in the training set, 10,666 pairs as the development set and 1,106 pairs in the test set. The long one is NLPCC Evaluation Task 4[2], which contains text-summary pair (50k totally) for the training and the development set respectively and the test set contains 2500 text-summary pairs.

Preprocessing for Chinese Corpus. As we know, word segmentation is the first step in Chinese text processing, which is very different from English one. The accuracy of word segmentation is about 96% and more [10]. However, this tiny error-rate results in more high-frequency but wrong words and unregistered word with the growth of

[1] weibo.sina.com.

[2] http://tcci.ccf.org.cn/conference/2015/pages/page05_evadata.html.

corpus size, which makes the vocabulary larger. This problem will bring about more time-cost and accuracy-loss in Chinese text summarization.

Previous works generally use a 150k word vocabulary on 280k Long English corpus (CNN/Daily Mail). This vocabulary can be further reduced to 30k or lower by means of morphological reduction [21], stem reduction, and wrong word checking [22]. However, the long Chinese corpus NLPCC has a 500k word vocabulary with word frequency higher than 10. Unfortunately, in Chinese we usually directly truncate the word vocabulary, which leads to more unregistered words. Therefore, in experiments we reduce word vocabulary by representing text using characters rather than words. In previous study, the character level methods have achieved good results in English summarization [23, 24]. From intuition, character-level models for Chinese summarization are more effective because a Chinese character is meaningful, while an English letter is meaningless. This strategy also bypasses the cascade errors reduced by word segmentation.

Evaluation Metric. For evaluation, we adopt the popular evaluation metrics F1 of Rouge proposed by [25]. Rouge-1 (unigrams), Rouge-2 (bigrams) and Rouge-L (longest-common substring LCS) are all used.

4.2 Comparative Methods

To evaluate the performance of *Adversarial Reinforcement Learning*, we compare our model with some baselines and state-of-the-art methods: (i) Abs: [26] is the basic seq2seq model, which is widely used for generating texts, so it is an important baseline. (ii) Abs+: [13] is the baseline attention-based seq2seq model which relies on LSTM network encoder and decoder. It achieves 42.57 ROUGE-1 and 23.13 ROUGE-2 on English corpus Gigaword, using Google's textsum[3]. The experiment setting is 120-words text length, 4-layers bidirectional encoding and 200k vocabulary. (iii) Abs +TA: We extend Abs+ by introducing Text-Attention, referring to [20]. We compare this model to Abs+, in order to verify the effectiveness of Text-Attention. (iv) DeepRL: [27] is a new training method that combines standard supervised word prediction and reinforcement learning (RL). It uses two 200 dimensional LSTMs for the bidirectional encoder and one 400-dimensional LSTM. The input vocabulary size is limited to 150k tokens.

4.3 Model Setting

We compare our model with above baseline systems, including Abs, Abs+, Abs+TA and DeepRL. We refer to our proposed model as ARL. Experiments were conducted at word-level and character-level respectively.

In ARL model, the structure of G is based on Abs+TA. The encoder and decoder both use GRUs. In a series of experiments, we set the dimensions of GRU hidden state as 512. We start with a learning-rate of 0.5 which is an empirical value and use the Adam optimization algorithm. For D, the RNNs use LSTM units and learning rate is set

[3] https://github.com/tensorflow/models/tree/master/research/textsum.

as 0.2. The settings of hidden state layer and optimization algorithm in *D* and *G* are consistent. In order to successfully train the ARL model, we sampled the generated summary and the real summary randomly before training *D*. Due to the finiteness of generated summary, we use mini-batch strategy to feed text-summary pairs into *D*, in case of collapse mode. The minibatch is usually set as 64.

In LCSTS word-level experiments, to limit the vocabulary size, we prune the vocabulary to top 150k frequent words, and replace the rest words with the 'UNK' symbols. We used a random initialized 256-dimensional word2vec embeddings as input. In char-level ones, we use Chinese character sequences as both source inputs and target outputs. We limit the model vocabulary size to 12k, which covers most of the common characters. Each character is represented by a random initialized 128-dimensional word embedding.

In NLPCC word-level experiments, we set vocabulary size to 75k, and the encoder and decoder shared vocabularies. 256-dimensional word2vec embeddings are used. In char-level ones, the vocabulary size is limited to 4k and the dimensional of word2vec embeddings to 128.

All the models are trained on the GPUs Tesla V100 for about 500,000 iterations. The training process took about 3 days for our character-level model on NLPCC and LCSTS, 4 days for the NLPCC word-level model, and 6 days for the LCSTS character-level model. The training cost of comparative models varies between 6–8 days.

4.4 Training Details

As we all know, training GAN is very difficult. Therefore, in the process of implementing the model, we have applied some small tricks.

At the beginning of training, the generator's ability is still poor, even after pre-training. *G* are almost impossible to produce smooth and high-quality summary. And when *G* send these bad summaries to the *D*, *D* can only back a low reward. As previously mentioned, the training of the *G* can only be optimized by the feedback of the *D*. So, the *G* cannot know what a good result. Under the circumstances, the iteration training between *G* and *D* is obviously defective.

To alleviate this issue and give the generator more direct access to the gold-standard targets, we introduce the professor-forcing algorithm of [28]. We update model by human-generated responses. The most straightforward strategy is to automatically assign 1 (or other positive) rewards to the human generated response and let the *generator* use the reward to update the human generated example.

We first pre-train the generator by predicting target sequences given the text history. We followed protocols recommended by [26], such as gradient clipping, mini-batch and learning rate decay. We also pre-train the discriminator. To generate negative examples, we decode part of the training data. Half of the negative examples are generated using beam-search with mutual information and the other half is generated from sampling.

In order to keep the *G* and the *D* optimize synchronously, experimentally, we train *G* once every 5 steps of the *D* until the model converges.

4.5 Results Analysis

Results on LCSTS Corpus. The ROUGE scores of the different summarization methods are presented in Table 1. As can be seen, the character-level models always perform better than their corresponding word-level ones. And it's notable that our proposed character-level ARL model enjoys a reasonable improvement over character-level DeepRL, indicating the effectiveness of the adversarial strategy. Besides, ARL model prominently outperforms two baselines (Abs and Abs+). With respect to other methods, we found that, the MLE's training objective is flawed in text summarization task. In addition, the performance of character-level Abs+TA proved the effectiveness of Text-Attention.

Table 1. Rouge-score on LCSTS corpus

System	Rouge-1	Rouge-2	Rouge-L
Abs(word)	17.7	8.5	15.8
Abs(char)	21.5	8.9	18.6
Abs+(word)	26.8	16.1	24.1
Abs+(char)	29.9	17.4	27.2
Abs+TA(char)	30.1	18.7	28.8
DeepRL(char)	37.9	21.4	29.0
ARL(word)	31.9	17.5	27.5
ARL(char)	***39.4**	***21.7**	***29.1**

Table 2 is an example to show the performance of our model. We can find that the results of ARL model in word-level and char-level are both closer to the main idea in semantics, while the results generated by Abs+ are incoherent. And there is a lot of "_UNK" in ABSw, even using a large vocabulary. Even more, on test set, the results of word-level models (ABSw and ARLw) have a lot of "_UNK", which are rare in the character-level models (ABSc and ARLc). This indicates that the character-level models can reduce the occurrence of rare words. To a certain extent, it improves the performance of all models referred in this section.

Results on NLPCC Corpus. Results on the long text dataset NLPCC are shown in Table 3. Our model ARL also achieves the best performance. It is worth noting that the methods character-level Abs+ is not better the word-level one. That's because attention will produce offset in long text, and our character-level ABS+TA has a good effect at the moment.

Table 2. An example of generated summaries on the test set of LCSTS corpus. **S:** the source texts, **R:** human summary, **ABSw:** Abs+ summary with word-level, **ABSc:** Abs+ summary with char-level (ABc), **ARLw:** AR summary with word-level and **ARLc:** AR summary with char-level and replacing Arabic numbers in "TAGNUM"

S:今天有传在北京某小区，一光头明星因吸毒被捕的消息。下午北京警方官方微博发布声明通报情况，证实该明星为李代沫。李代沫伙同另外6人，于17日晚在北京朝阳区三里屯某小区的暂住地内吸食毒品，6人全部被警方抓获，且当事人对犯案实施供认不讳。 Today, a bald star was arrested for drug abuse, in a Beijing neighborhood. In the afternoon, the Beijing police issued a statement, through the official microblog, confirming that the star was Daimo Li. Daimo Li, with six other people, took drugs in a temporary residence in a district of Sanlitun, Chaoyang district, Beijing, on the evening of the 17th. All six people were arrested by the police, and the parties confessed to the crime.
R: 北京警方确认李代沫吸毒被捕(图) Beijing police confirmed Daimo Li was arrested for drug abuse (photo)
ABSw: 北京警方李代沫吸毒被捕系歌手 _UNK _UNK _UNK _UNK _UNK Beijing police, Daimo Li arrested for drug as a singer _UNK*5.
ABSc:北京警方明星为李代沫吸毒被捕系谣言 Beijing police, the star Daimo Li arrested for drug was a rumor.
ARLw: 网传李代沫吸毒被抓 Internet Communication, Daimo Li was arrested for drug abuse
ARLc: 北京警方确认李代沫吸毒被捕,警方抓获TAGNUM 人 Beijing police confirmed Daimo Li was arrested for drug abuse. Police arrested TAGNUM people

Table 3. Rouge-score on NLPCC corpus

System	Rouge-1	Rouge-2	Rouge-L
Abs+(word)	11.0	6.7	14.0
Abs+(char)	14.3	5.7	13.2
ABS+TA(char)	24.5	8.7	21.8
DeepRL(char)	33.9	16.4	29.5
ARL(char)	***34.4**	***17.6**	***29.6**

5 Conclusion

In this work, we propose an *Adversarial Reinforcement Learning* architecture for Chinese text summarization. This model got promising results in experiments which generating more natural and continuous summaries. Meanwhile, we successfully solved the word segmentation error and distant dependence of text via character-level representation and Text-Attention mechanism. In such a framework, we teach the

generator to generate analogous human summary in the continuous space, which is achieved via introducing an adversarial discriminator which tries it best to distinguish the generated summarizations from the real ones.

There are several problems need to be resolved in the future work. One is that, due to the complex structure of Chinese sentences, we want to combine linguistic features (such as part-of-speech, syntax tree) with our ARL model. The other one is that, our model is still a supervised learning one relying on high-quality training datasets which is scarce. So, we will study an unsupervised or semi-supervised framework which can be applied to the text summarization task.

Acknowledgement. This work was supported by the National Key Research and Development program of China (No. 2016YFB0801300), the National Natural Science Foundation of China grants (NO. 61602466).

References

1. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization. MIT Press, Cambridge (1999)
2. Mani, I.: Automatic Summarization, vol. 3. John Benjamins Publishing, Amsterdam (2001)
3. Ruch, P., Boyer, C., et al.: Using argumentation to extract key sentences from biomedical abstracts. Int. J. Med. Inform. **76**(2–3), 195–200 (2007)
4. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
5. Wu, Y., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
6. Ma, S., Sun, X.: A semantic relevance based neural network for text summarization and text simplification. arXiv preprint arXiv:1710.02318 (2017)
7. Liu, L., et al.: Generative Adversarial Network for Abstractive Text Summarization. arXiv preprint arXiv:1711.09357 (2017)
8. Rush, A.M., et al.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
9. Nallapati, R., Zhou, B., et al.: Abstractive text summarization using sequence-to-sequence RNNS and beyond. arXiv preprint arXiv:1602.06023 (2016)
10. Peng, F., Feng, F., et al.: Chinese segmentation and new word detection using conditional random fields. In: Proceedings of the 20th International Conference on Computational Linguistics, p. 562 (2004)
11. Goodfellow, I., Pouget-Abadie, J., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
12. Radford, A., Metz, L., et al.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
13. Liu, P., Pan, X.: Sequence-to-Sequence with Attention Model for Text Summarization (2016)
14. Hu, B., Chen, Q., et al.: LCSTS: A Large Scale Chinese Short Text Summarization Dataset (2015)
15. Sutton, R.S., Barto, A.G.: Reinforcement Learning: an Introduction, vol. 1. MIT Press, Cambridge (1998). no. 1

16. Ryang, S., Abekawa, T.: Framework of automatic text summarization using reinforcement learning. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 256–265 (2012)
17. Yu, L., Zhang, W., et al.: SeqGAN: sequence generative adversarial nets with policy gradient. In: AAAI, pp. 2852–2858 (2017)
18. Makhzani, A., Shlens, J., et al.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Wang, B., Liu, K., et al.: Inner attention based recurrent neural networks for answer selection. In: ACL (1) (2016)
21. Tolin, B.G., et al.: Improved translation system utilizing a morphological stripping process to reduce words to their root configuration to produce reduction of database size (1996)
22. Perkins, J.: Python Text Processing with NLTK 2.0 Cookbook. Packt Publishing Ltd (2010)
23. Kim, Y., Jernite, Y., et al.: Character-aware neural language models. In: AAAI, pp. 2741–2749 (2016)
24. Zhang, X., Zhao, J., et al.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems, pp. 649–657 (2015)
25. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using N-gram co-occurrence statistics. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 71–78 (2003)
26. Sutskever, I., Vinyals, O., et al.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
27. Li, P., Lam, W., Bing, L., et al.: Deep Recurrent Generative Decoder for Abstractive Text Summarization. arXiv preprint arXiv:1708.00625 (2017)
28. Lamb, A.M., Goyal, A.G., et al.: Professor forcing: a new algorithm for training recurrent networks. In: Advances in Neural Information Processing Systems, pp. 4601–4609 (2016)