A NOVEL REPETITION NORMALIZED ADVERSARIAL REWARD FOR HEADLINE GENERATION

Peng Xu and *Pascale Fung*

Department of Electronic and Computer Engineering
Center for Artificial Intelligence Research (CAiRE)
The Hong Kong University of Science and Technology, Clear Water Bay
pxuab@ust.hk, pascale@ece.ust.hk

ABSTRACT

While reinforcement learning can effectively improve language generation models, it often suffers from generating incoherent and repetitive phrases [1]. In this paper, we propose a novel repetition normalized adversarial reward to mitigate these problems. Our repetition penalized reward can greatly reduce the repetition rate and adversarial training mitigates generating incoherent phrases. Our model significantly outperforms the baseline model on ROUGE-1 (+3.24), ROUGE-L (+2.25), and a decreased repetition-rate (-4.98%).

Index Terms— adversarial training, reinforcement learning, headline generation, summarization

1. INTRODUCTION

Summarization is the task of condensing a long document into a short summary without losing the main information. It has attracted lots of attention from the research community for its application to digest a large amount of information produced every day. Summarization can be generally categorized into two classes, extractive summary, and abstractive summary. Extractive summary [2] exclusively takes words from an input document and assembles them into a summary, while abstractive summary [3] needs to understand the document first and learns to paraphrase and generate new phrases. We focus on abstractive summary in this paper.

Summarization models usually adopt maximum likelihood training. This training method suffers from two major disparities between training and testing, 1) exposure bias [4], that is during the training phases, the words from ground truth sentence are fed to the model while in the testing phase, the input to decoder comes from the prediction of generator. 2) different measurement between training and testing. In the training phase, the cost is measured by cross-entropy while in the testing phase, the model is evaluated with non-differentiable metrics, like ROUGE [5] score. Reinforcement learning, as it can directly optimize non-differentiable metrics, has been proposed and successfully improved the performance of generation quality, using ROUGE as reward [1]. Nevertheless, reinforcement learning also suffers from generating incoherent and repeated phrases. Thus, a mixed training objective function and an intra-decoder is proposed in [1]. From the perspective of reinforcement learning, we believe those two problems all come from a sub-optimal reward, ROUGE. As ROUGE has no awareness of the quality of generated samples but just counts n-gram matches, it enforces no penalty on repetition and incoherence. For example, Table 1 shows that a bad headline can achieve a high ROUGE score as it includes " benson " , " for ", " to " and " citigroup " that overlap with real headlines.

Article: (...) He will miss the first # minutes of the opening practice session for the NAPA California # on Friday as a penalty for being late to the pre-race inspection last Sunday at Alabama 's Talladega Superspeedway for the DieHard # . It will mark the second straight race in which Benson has missed practice time because of a rules infraction . (...)
RL: benson to the for ford
Ground truth: benson penalized for his bad timing
ROUGE-L score: 0.358
Article: (...) Vikram S. Pandit is doing some serious spring cleaning at Citigroup . Since becoming chief executive in December , Pandit has been clearing out the corporate attic of weak businesses and unloading worrisome assets at bargain-basement prices . (...)
RL: sports column : citigroup to citigroup at citigroup
Ground truth: citigroup embarks on plan to shed weak assets
ROUGE-L score: 0.25

Table 1. Examples show that a bad headline with repetition and incoherence phrases can have high ROUGE score.

In this paper, we propose a novel repetition normalized adversarial reward for reinforcement learning to mitigate the problems of incoherent and repeated headlines. We empirically show that our repetition penalized reward significantly reduces the repetition rate and adversarial training helps to generate more coherent phrases.

Thanks to ITS/319/16FP of Innovation Technology Commission, HKUST 16248016 of Hong Kong Research Grants Council for funding.

2. RELATED WORK

Deep learning methods are first applied to two sentence-level abstractive summarization task on DUC-2004 and Gigaword datasets [3] with an encoder-decoder model. This model is further extended by hierarchical network [6], variational autoencoders [7], a coarse to fine approach [8] and minimum risk training [9]. As long summaries becomes more important, CNN/Daily Mail dataset was released in [6]. Pointer-generator with coverage loss [10] is proposed to approach the task by enabling the model to copy unknown words from article and penalizing the repetition with coverage mechanism. [11] proposes deep communicating agents for representing a long document for abstractive summarization. There are more papers focusing on extractive summarizations[2, 12].

Reinforcement learning is also gaining popularity as it can directly optimize non-differentiable metrics [13, 1]. [1] proposes an intra-decoder model and combines reinforcement learning and maximum likelihood training to deal with summaries with bad qualities. We instead propose to improve headlines with a novel reward. Reinforcement learning is also explored with generative adversarial network (GAN) [14]. [15] applies the generative adversarial network for summarization to achieve a better performance. However, they directly take the score from the discriminator as the reward while we combine it with repetition penalized ROUGE score.

3. METHODOLOGY

Our model consists of two parts, an attentional sequence to sequence model with pointer-generator and a Convolutional Neural Network (CNN) discriminator. The attentional sequence to sequence model takes a news article as input and generates a headline. The discriminator takes either the generated headline or real headline as input and outputs a probability of how likely the generated headline is real. This probability is then combined with the ROUGE score between the generated sample and the real one, as the reward for our sequence to sequence model. The model is shown in Figure 1.

3.1. Attentional sequence to sequence model with pointer-generator

We abbreviate this attentional sequence to sequence model with pointer-generator [10] as Pointer-Gen-Coverage. The model is described below. Firstly, the tokens of each article w_i are fed into the encoder one-by-one and the encoder generates a sequence of hidden states h_i. For each decoding step t, the decoder receives the embedding for each token of a headline as input, and updates its hidden states s_t. The attention

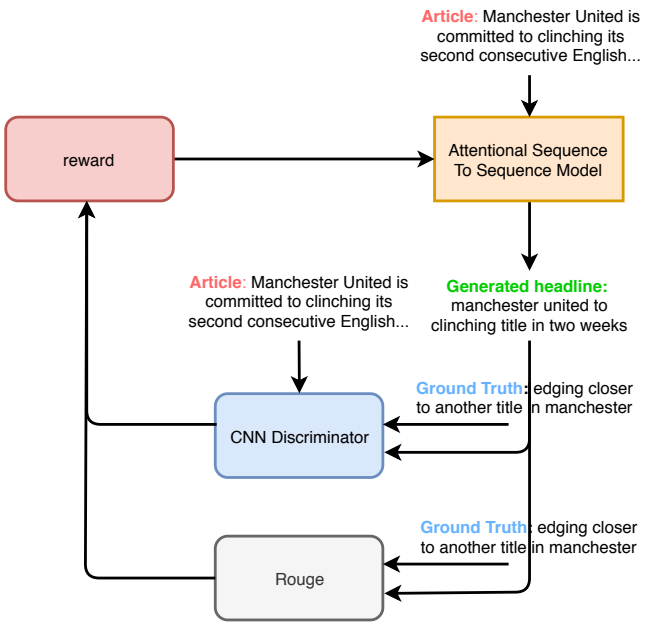

Fig. 1. Diagram of our learning framework

mechanism is calculated as in [16]

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{attn}) \quad (1)$$

$$a^t = \text{softmax}(e^t) \quad (2)$$

$$h_t^* = \sum_i a_i^t h_i \quad (3)$$

where W_h, W_s, w_c, b_{attn}, v are the trainable parameters, h_t^* is the context vector, c_i^t is the coverage vector defined below. s_t, h_t^* are then combined to predict the next word.

$$c_i^t = \sum_{t'=0}^{t-1} a_i^{t'} \quad (4)$$

$$o_t = V([s_t, h_t^*]) + b \quad (5)$$

$$P_{vocab} = \text{softmax}(V' o_t + b') \quad (6)$$

where V, b, V', b' are parameters to train. c_i^t is defined as the accumulated attention over specific positions. We also include pointer generator network to enable our model to copy rare/unknown words from input article.

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x^t + b_{ptr}) \quad (7)$$

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (8)$$

where x^t is the embedding of input word of decoder, $w_{h^*}^T$, w_s^T, w_x^T, b_{ptr} are trainable parameters. Our final loss function for Maximum Likelihood (ML) training thus becomes:

$$L_{\text{ml}} = -\frac{1}{T} \sum_{t=1}^{T} (\log P(w_t) + \lambda \sum_i \min(a_i^t, c_i^t)) \quad (9)$$

3.2. Adversarial training

To measure the quality of generated sample, we train a CNN discriminator to classify whether the sample is generated or a real one. Our CNN model takes an article A and real headline or generated headline H as input. We employ two CNNs with same structures [17] to encode H and A respectively. The features are then concatenated and then projected to one single scalar $D(A, H)$ with a sigmoid activation, as the score of the headline being a real one. The loss function is constructed to maximize the log likelihood of real samples and minimize that of generated samples.

$$L_d = -E_{H\sim p_{data}}\log D(A,H) - E_{H\sim P(w)}\log(1-D(A,H)) \quad (10)$$

3.3. Repetition normalized adversarial reward

Our Repetition normalized adversarial reward (ROUGE-RP-ADV) is the harmonic mean of repetition penalized ROUGE score (ROUGE-RP) and CNN discriminator score $D(A, H)$ by further introducing β to balance ROUGE-RP and $D(A, H)$. Larger β encourages our model to emphasize ROUGE-RP. Repetition-rate, ROUGE-RP and ROUGE-RP-ADV are defined below.

$$\text{repetition-rate} = 1. - \frac{N(\text{unique tokens of H})}{N(\text{total tokens of H})} \quad (11)$$

$$\text{ROUGE-RP} = (1 - \text{repetition-rate}) * \text{ROUGE} \quad (12)$$

$$\text{ROUGE-RP-ADV} = \frac{(1+\beta^2)\,\text{ROUGE-RP} * D(A,H)}{\text{ROUGE-RP} + \beta^2 D(A,H)} \quad (13)$$

where N counts the number of tokens.

3.4. Reinforcement Learning

We use REINFORCE algorithm [18] and baseline model proposed in [4] to train our generator (Pointer-Gen-Coverage). In each training step, a sentence is first sampled based on the $P(w)$ from our generator. A reward R is then calculated between generated sample and real headline. For each time t, a linear regression model is utilized to estimate the reward of step t based on t-th state o_t. The linear regression model and loss function is shown below:

$$\hat{R}_t = W_r o_t + b_r \quad (14)$$

$$L_b = \frac{1}{T}\sum\nolimits_{t=1}^{T} ||R - \hat{R}_t||^2 \quad (15)$$

where W_r and b_r are trainable parameters, R is the reward for whole sentence. Our final loss function for reinforcement learning (RL) becomes:

$$L_{\text{RL}} = -\frac{1}{T}\sum\nolimits_{t=1}^{T}(R - \hat{R}_t)\log P(w_t) \quad (16)$$

4. EXPERIMENTS

4.1. Dataset

Recent neural headline generation models focus on generating headlines from selected recapitulative sentences. However, these selected sentences may not have enough information for the generation, as for example, in New York Times, the overlap between headlines and the sentences is very low [8]. Thus, in this paper, we focus on generating headlines from full document. We use the dataset of New York Times part in Gigaword [19]. Different from [3, 6], we use the whole document as our input. We follow the preprocessing steps [1] in [3] and we then use the NLTK [20] to tokenize our dataset. The average length of headlines and articles are 863.4 and 8.6, respectively. We empirically choose 400 words as the maximum article length as it covers 67% tokens of headlines while full documents achieve 73% overlap. Following [8], we randomly split our train, dev, and test set as 1106824, 2000 and 2000.

4.2. Training Details

ML learning: Our reinforcement learning model is first pretrained by optimizing L_{ml}. Adam optimizer is used and learning rate is 0.0001. Batch size is set as 16 and one layer, bi-directional Long Short-Term Memory (bi-LSTM) [21] model with 512 hidden sizes and 100 embedding size is utilized. Gradients with l2 norm larger than 2.0 are clipped. We stop training when the ROUGE-1 f-score stops increasing. Beam-search with the beam size of 5 is adopted for decoding.
Adversarial Training: As our RL model is well pretrained, CNN discriminator also needs to be pretrained to avoid an imbalanced generator and discriminator. CNN discriminator is trained by optimizing L_d by one epoch. We use one layer CNN model with filter sizes of 1, 3, and 5. Each channel contains 512 filters. Adam optimizer with the learning rate of 0.001 is used. After training, our CNN discriminator achieves the accuracy of 0.6945 on real headlines and accuracy of 0.6975 on generated headlines.
RL training: We found that using reward of ROUGE-1 f-score will always reach the maximum decoding length, thus the f-score for ROUGE-L is utilized as the ROUGE reward. For Pointer-Gen-Coverage model, we use the Adam optimizer with a learning rate of 0.0001. When ROUGE-RP-ADV is used as the reward, our best model is achieved when β is 2000. When mixing ML with RL [1], a large weight α for RL is necessary to achieve good performance and the best model is acquired with α set as 0.97.

4.3. Results

We report f-score for ROUGE-1, ROUGE-2, ROUGE-L on the test set. Table 2 shows the results of different models.

[1] https://github.com/facebookarchive/NAMAS

models	ROUGE-1	ROUGE-2	ROUGE-L	repetition-rate
Pointer-Gen-Coverage	24.68	10.92	21.78	9.54 %
Pointer-Gen-Coverage + ROUGE	**28.65**	9.70	23.89	13.42%
Pointer-Gen-Coverage + ROUGE + MLE	26.64	**11.87**	23.54	10.28%
Pointer-Gen-Coverage + ROUGE-RP	27.51	9.39	23.54	**4.23%**
Pointer-Gen-Coverage + ROUGE-RP-ADV	27.92	10.27	**24.03**	4.56%

Table 2. Results of different models on f-score for different ROUGE measures and repetition rate. The larger ROUGE score and smaller repetition rate implies better results.

Pointer-Gen-Coverage (**26.43**): **manchester united to retain second english premier title**
Pointer-Gen-Coverage+ROUGE (**52.86**): **manchester united to second title in manchester united**
Pointer-Gen-Coverage+ROUGE-RP (**28.57**) : **manchester united to second english premier title**
Pointer-Gen-Coverage+ROUGE-RP-ADV (**39.65**) : **manchester united to clinching title in two weeks**
Pointer-Gen-Coverage+ROUGE+MLE (**26.43**) : **manchester united to keep # nd english title**
Ground Truth : **edging closer to another title in manchester**

Table 3. An example of headlines generated by different models. The ROUGE-L score is reported inside parenthesis. Pointer-Gen-Coverage+ROUGE+MLE generates a headline with 26.43 ROUGE-L score. Pointer-Gen-Coverage+ROUGE gives 52.86 ROUGE-L score with repetition of "manchester united", Pointer-Gen-Coverage+ROUGE-RP gives incoherent phrases with a score of 28.57, and Pointer-Gen-Coverage+ROUGE-RP-ADV generates a more coherent headline and achieves the high ROUGE-L score of 39.65.

The baseline Pointer-Gen-Coverage model achieves 24.68 ROUGE-1 score, 10.92 ROUGE-2 score, 21.78 ROUGE-L score, and repetition-rate of 9.54%. When applying ROUGE score alone as the reward, ROUGE-1 score increases to 28.65, and ROUGE-L increases to 23.89. However, the repetition-rate also increases to 13.42%, which shows that using ROUGE alone as reward improves the performance but also introduces more repetitions.

When using repetition penalized rouge reward, repetition rate decreases from 13.42% to 4.23%. It implies that by adding the penalty to reward, our model learns to generate headlines with fewer repetitions. However, we observe that both Pointer-Gen-Coverage+ROUGE and Pointer-Gen-Coverage+ROUGE-RP model produce incoherent headlines like " for opera singer , a tenor to the ", which ends the headline abruptly. By combining ROUGE score with CNN discriminator score, our Pointer-Gen-Coverage+ROUGE-RP-ADV model generates more natural headlines and achieves ROUGE-1 27.92, ROUGE-L 24.03, and a decreased repetition-rate of 4.56%. It outperforms the baseline model of Pointer-Gen-Coverage on ROUGE-1 (+3.24), ROUGE-L (+2.25), and a decreased repetition-rate (-4.98%). An example is shown in Table 3 to demonstrate the differences. We also compare our Pointer-Gen-Coverage+ROUGE-RP-ADV model to the model with a mixed training objective function (Pointer-Gen-Coverage+ROUGE+ MLE), which is introduced in [1] to deal with incoherent generations. Our model achieves better results on ROUGE-1 (+1.28), ROUGE-L (+0.49), and repetition-rate (-5.72%).

To further understand how the repetition penalized reward reduces generations of repeated phrase, we calculate the ROUGE-RP score for Pointer-Gen-Coverage+ROUGE and Pointer-Gen-Coverage+ROUGE-RP respectively and we get 20.91 and 22.58. Compared to Pointer-Gen-Coverage+ROUGE, Pointer-Gen-Coverage+ROUGE-RP achieves lower ROUGE-L score (23.54 vs 23.89) but higher ROUGE-RP score (22.58 vs 20.91) and lower repetition rate. This illustrates that, our model is encouraged to sacrifice ROUGE-L score for repetition avoidance. For the adversarial training, we take the trained CNN discriminator out and feed it with the model outputs of Pointer-Gen-Coverage+ROUGE-RP and Pointer-Gen-Coverage+ROUGE-RP-ADV in Table 3. The scores for Pointer-Gen-Coverage+ROUGE-RP, Pointer-Gen-Coverage+ROUGE-RP-ADV and ground truth are 0.9993, 0.9996, and 1.0. Our CNN discriminator believes Pointer-Gen-Coverage+ROUGE-RP-ADV generates a better headline than Pointer-Gen-Coverage+ROUGE-RP.

5. CONCLUSION

In this paper, we proposed a repetition normalized adversarial reward for reinforcement learning on headline generation. We empirically showed that our repetition penalized reward greatly decreased the repetition rate of the generated headlines and adversarial training further helped the model generate more natural headlines. Experiments showed that our model outperformed the baseline model on ROUGE-1 (+3.24), ROUGE-L (+2.25), and a decreased repetition-rate (-4.98%).

6. REFERENCES

[1] Romain Paulus, Caiming Xiong, and Richard Socher, "A deep reinforced model for abstractive summarization," *Sixth International Conference on Learning Representations*, 2018.

[2] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.," in *AAAI*, 2017, pp. 3075–3081.

[3] Alexander M Rush, Sumit Chopra, and Jason Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 379–389.

[4] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba, "Sequence level training with recurrent neural networks," *International Conference on Learning Representations*, 2016.

[5] Chin-Yew Lin, "Rouge: A package for automatic evaluation of summaries," *Text Summarization Branches Out*, 2004.

[6] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," *CoNLL 2016*, p. 280, 2016.

[7] Yishu Miao and Phil Blunsom, "Language as a latent variable: Discrete generative models for sentence compression," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 319–328.

[8] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao, "From neural sentence summarization to headline generation: a coarse-to-fine approach," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 4109–4115.

[9] Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, Mao-Song Sun, et al., "Recent advances on neural headline generation," *Journal of Computer Science and Technology*, vol. 32, no. 4, pp. 768–784, 2017.

[10] Abigail See, Peter J Liu, and Christopher D Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, vol. 1, pp. 1073–1083.

[11] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi, "Deep communicating agents for abstractive summarization," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, vol. 1, pp. 1662–1675.

[12] Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, vol. 1, pp. 654–663.

[13] Ramakanth Pasunuru and Mohit Bansal, "Multi-reward reinforced summarization with saliency and entailment," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, vol. 2, pp. 646–653.

[14] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu, "Seqgan: Sequence generative adversarial nets with policy gradient.," in *AAAI*, 2017, pp. 2852–2858.

[15] Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li, "Generative adversarial network for abstractive text summarization," *AAAI*, 2018.

[16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations*, 2015.

[17] Yoon Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[18] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[19] Courtney Napoles, Matthew Gormley, and Benjamin Van Durme, "Annotated gigaword," in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics, 2012, pp. 95–100.

[20] Steven Bird and Edward Loper, "Nltk: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.

[21] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.