LEARNING COMMENT GENERATION BY LEVERAGING USER-GENERATED DATA

Zhaojiang Lin, Genta Indra Winata, Pascale Fung

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

ABSTRACT

Existing models on open-domain comment generation are difficult to train, and they produce repetitive and uninteresting responses. The problem is due to multiple and contradictory responses from a single article, and by the rigidity of retrieval methods. To solve this problem, we propose a combined approach to retrieval and generation methods. We propose an attentive scorer to retrieve informative and relevant comments by leveraging user-generated data. Then, we use such comments, together with the article, as input for a sequence-to-sequence model with copy mechanism. We show the robustness of our model and how it can alleviate the aforementioned issue by using a large scale comment generation dataset. The result shows that the proposed generative model significantly outperforms strong baseline such as Seq2Seq with attention and Information Retrieval models by around 27 and 30 BLEU-1 points respectively.

Index Terms— comment generation, natural language generation, copy attention, pointer-generator, user-generated data

1. INTRODUCTION

Commenting on online articles has been a popular method to collect the opinions and interests from the crowd. The quality of the comments often represents the level of users' engagement [1]. However, not every comment is relevant, and often they contain inappropriate content such as abusive language [2]. Normally, upvote count is used by online forums to rank the quality of a comment according to users' preferences and bury the irrelevant comments. The persuasiveness of the comments positively correlates by the number of votes given to them [3]. Usually, popular comments receive many upvotes from users. The upvotes count in a comment indicates the level of attention from readers to the comment. The count is also useful to distinguish between relevant user opinions
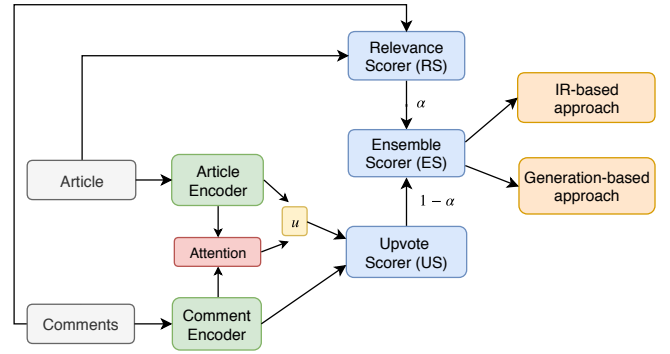
Fig. 1. Comment Generation Framework

and undesired content such as spams including advertisement, double posting, or any offensive comments.

In recent years, automatic comment generation has become a prominent topic. The ability to comment on an article requires natural language understanding to conceptualize the idea of the article and provide a relevant response. Previous generative models suffer from the mode collapse issue where the models produce samples with extremely low variety [4]. In comment generation task, we also suffer the same issue when we train our models on articles together with all comments that have a huge variance. As a result, the model is hard to converge, and generated comments are not meaningful. On the other hand, an Information Retrieval (IR) approach can pick comments from real users, but this approach is not scalable.

To build a scalable generator that can generate informative and relevant comments, we propose a framework to learn comment generation by leveraging user-generated data such as upvote count. Combining the user-generated data such as upvote count helps the model to decide which sequences are to be decoded. However, not every article has upvote information. To solve the problem, we propose to build a neural classifier to score the comments and combine with a generative model to effectively mitigate mode collapse issue. For the generation-based approach, we use a pointer-generator net-

work [5] to learn and copy essential words from the articles. Our proposed method significantly improves the performance by around 25 BLEU-1 points, 3 CIDEr points, 5 ROUGE_L points, and 6 METEOR points compared to standard pointer-generator network and it outperforms the IR models in most of the evaluation metrics.

2. RELATED WORK

Prior work in response generation focused on two major approaches: Information retrieval and Generative model.

Information retrieval: [6] proposed a neural-based method to clarify questions by calculating the expected value of the information in the technical discussion forum. In question answering, TF-IDF scores were employed to remove irrelevant answer candidates to reduce the search space [7]. A CNN encoder had been explored in automatic comment ranker to replace a standard IR-based similarity scorer [1].

Generative model: [8] introduced a gated attention neural-based generation model to address the problem of contextual relevance by choosing news context. [3] studied the impact of different sets of features to measure the comments' persuasiveness in the online forum. [9] showed that argumentative comment representations are useful to identify constructive comments using dataset provided by [10]. In summarization, the pointer-generator networks with copy mechanism were proposed by [5] to copy words from the source article to the generate a summary. Similarly to our work, the mechanism triggers the model to take some important keywords in the article and use it to generate relevant comments. [11][12][13] integrated copy mechanism with multi-hop memory network to efficiently utilize knowledge base in generating the response in task-oriented dialog system.

3. METHODOLOGY

In this section, we describe a standard information retrieval (IR)-based method and generation-based method for comment generation. We propose a user-centered method for scoring comments by leveraging upvotes collected from users to improve the relevance of the generated comments. We denote $a \in \{w_1^a, ..., w_n^a\}$ as the concatenation of the article and title with n tokens, and $c \in \{w_1^c, ..., w_m^c\}$ as the comment consists of m tokens. As our proposed method, we apply pointer-generator network to encode and copy words from the article. Figure 1 depicts the pipeline of our comment generation architecture.

3.1. Comment Scorer

We propose three methods to score comments: a relevance scorer (RS), an upvote scorer (US), and an ensemble of both.

3.1.1. Relevance Scorer (RS)

Similarly to IR-based methods, RS computes the dot product of TF-IDF weighted vector between an article with title a and comment c. We normalize the scores with the maximum to the article.

$$S_r = \text{Normalize}((v^a)^T v^c) \qquad (1)$$

3.1.2. Upvote Scorer (US)

Here, we introduce a semi-supervised method to score comments in the articles without any upvotes. The model is trained on articles that have at least a comment with ten upvotes. Comments with ten upvotes or above are used as positive samples, and others as negative samples. First, we represent each word as an embedding vector and pass it to a bidirectional LSTM (BiLSTM). The model is shared for both articles with title and comments.

$$\{h_1^a, ..., h_n^a\} = \text{BiLSTM}(\{v_1^a, ..., v_n^a\}) \qquad (2)$$

$$\{h_1^c, ..., h_m^c\} = \text{BiLSTM}(\{v_1^c, ..., v_m^c\}) \qquad (3)$$

We compute attention scores [14] to capture the dependency between articles and comment.

$$u = \sum_{i=1}^{n} \frac{exp(e_i)}{\sum_{k=1}^{n} exp(e_k)} h_i^a \qquad (4)$$

where

$$e_i = (h_i^a)^T \mathbf{W}_a h_m^c \qquad (5)$$

e_i is the attention score and $\mathbf{W}_a$ is a trainable parameter. We concatenate the context vector u and comment feature vectors h_m^c and pass it to a fully connected layer F. Next, the resulting value is normalized with a sigmoid function.

$$S_u = \sigma(\text{F}([u, h_m^c])) \qquad (6)$$

3.1.3. Ensemble Scorer (ES)

We combine the relevance score S_r and upvote score S_u by linear interpolation as an ensemble score.

$$S = \alpha S_r + (1 - \alpha) S_u \qquad (7)$$

where $\alpha \in [0, 1]$ is a hyper-parameter to weight S_r and S_u, We use the ensemble score to rank the comments by considering the relevance and user-generated upvote count.

3.2. Information Retrieval-based Commenting

This method is an unsupervised method to find the most relevant comment by retrieving comments from a large pool of candidates. In the standard IR-based method the articles and comments are represented as TF-IDF weighted bag-of-word

Table 1. Results on information retrieval and generation approaches. Higher scores are better.

	Model	BLEU-1	CIDEr	ROUGE_L	METEOR
Retrieval	TF-IDF + CNN [1]	**35.55**	0.25	21.92	14.25
	TF-IDF + RS	34.67	**1.37**	**23.67**	14.80
	TF-IDF + ES	34.53	1.19	23.59	**14.85**
Generation	Seq2Seq-Attn [1]	38.80	1.41	23.53	6.08
	pointer-generator + coverage	40.84	1.29	25.44	6.49
	pointer-generator + coverage + upvote	62.03	3.55	28.24	11.00
	pointer-generator + coverage + RS	56.39	3.95	26.47	11.88
	pointer-generator + coverage + US	64.22	4.17	**30.87**	12.11
	pointer-generator + coverage + ES	**65.70**	**4.35**	30.53	**12.62**

vectors [1]. The tokens are bigrams, and we use the *murmur3 hash function* to map bigrams to 2^{24} space similar to [7]. We apply a two-step retrieval to narrow down the search space and retrieve more relevant comment to the article. Given an article, we retrieve the top-5 most similar articles by calculating the dot product to their TF-IDF weighted vectors and build a candidate pool of comments from the retrieved articles. Then, we apply three different methods to score the comments:

- As a baseline, we apply a convolutional neural network (CNN) [15] to encode articles and comments similar to [1]. The inputs are tokens from an article and comments, and the model outputs the relevance score.

- We use RS by taking the dot product of the TF-IDF weighted vector of articles and comments.

- As our proposed method, we use ES and use them to rank the comments.

Next, we rank them in descending order to their scores, and we choose a comment with the highest score.

3.3. Generation-based Commenting

We use Seq2Seq [14] with pointer-generator [5] that copy words from articles to generate comments. During decoding time, we calculate $p_{gen} \in [0,1]$, the generation probability to weight vocabulary distribution for generation as following:

$$p_{gen} = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (8)$$

Next, the vocabulary distribution $P_{vocab}(w)$ is calculated by concatenating the decoder state s_t and the context vector h_t^*. The final distribution $P(w)$ is calculated from the weighted sum of the attention weights a and vocabulary distribution $P_{vocab}(w)$. To avoid repetitions, we added coverage [16] in the loss function [5].

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (9)$$

Similar to the IR-based approach, we use the proposed score functions to choose the best comments. We pair the best comment with the corresponding article to build the training set. Instead, for the standard seq2seq and pointer generator network with coverage baseline models, we create a set of training samples from all comments paired with the article. We also compare US with raw upvote as the score. We randomly sample one comment when there is an article without any upvote.

4. EXPERIMENT

4.1. Dataset

We collected a large-scale Chinese dataset from Tencent News, with four million real comments along with rich metadata using the script provided by [1]. The dataset reported by [1] has around 200K news articles and each of them has more than 20 comments. However, around 10% of news articles are already expired. The dataset has 177,368 articles with four million users comments. Each article has a title, text body and metadata including user upvote count and article categories. Upvote is given by users, and it represents comment's popularity. Overall, the dataset has four upvotes per comments on average with a long-tail distribution, where most of the comments have zero upvotes. For popular comments, they have more than thousands of votes. The dataset is split into training/validation/test sets which contains $171,440/4,512/1,417$ samples respectively.

4.2. Experimental Setup

For all experiments, we use a bidirectional LSTM with a 256-dimensional hidden state for the encoder and a unidirectional LSTM with a 512-dimensional hidden state for the decoder. We use a word embedding with a size of 128 with a vocabulary size of 50k following [5]. The word embedding is shared with the encoder and decoder. We use Adam for the optimizer, and our model is trained with an initial learning rate of 1e-3, and it decays at the rate of 0.5 every epoch when there is an increase in validation perplexity. At the inference time,

Table 2. The generated comments from different models.

Title: 流浪狗被人残忍踢断下巴无法进食，路人心疼不已送它去医院。 (The stray dog was brutally kicked off the chin and could not eat. A passerby sent it to the hospital.)	**Model**	**Generated Comment**
Body: 一条流浪狗在路边的垃圾堆翻找着吃的东西，嘴里发出叫声，似乎很痛苦，路过的男子见状将其抱回。他发现流浪狗狗的下巴断了，似乎是人为踢断的，非常可怜，可以想象它有多么的痛苦。狗狗很饿，在垃圾堆里找吃的，但是它的下巴变成这样是无法吃东西的，不知道是什么人这么狠毒。这名男子非常心疼，马上带狗狗去了宠物医院。经过医生的治疗，狗狗的下巴正在恢复中，现在能够吃东西了，不过只能吃一些流体的、软的食物,回到咬骨头的时期恐怕是需要一段时间了。 (A stray dog was searching for food in the rubbish heap on the roadside. The cry seemed to be very painful, and the passing man saw it and took it home. He found **the dog's chin was broken**. It seemed to be kicked by someone. Very **poor**. **We can imagine how painful it is**. The dog was hungry, looking for food in the rubbish heap, but **it was unable to eat** due to the broken chin. We don't know who is so vicious. The man was very distressed, immediately took the **dog** to the pet hospital. After the doctor's treatment, dog's jaw is recovering, and now it can eat, but it can only eat some fluid, soft food, it may take a while to go back to the time of biting bones.)	TF-IDF + CNN [1]	想起来我曾经养的一条小狗...后来, 因为吃死老鼠被药死了。我一直都记着它那双会说话的眼睛。 (I remembered a puppy I used to raise... Later, It was dead from food poisoning because of a dead mouse. I always remember its talking eyes.)
	TF-IDF + ES	听听都害怕, 踢断下巴 (I feel scared to hear that the chin was kicked off.)
	Seq2Seq-Attn	好人有好报 (A good person will be rewarded)
	pointer-generator + coverage + RS	狗狗的下巴断了，似乎是无法吃东西的，但是只能想象它有多么的痛苦。 (**The dog's chin was broken**. It seemed that **it was unable to eat**. But **we can imagine how painful it is**.)
	pointer-generator + coverage + ES	可怜的狗狗! (**Poor dog**!)

we replace all unknown tokens with the source tokens with the highest attention weight and block repeating tokens. The upvote scorer is trained with the same setting as above, and we add a pre-trained 300-dimensional skip-gram Word2Vec [17] trained on 374M words Sogou News[1] with 270,830 words vocabulary. The embedding layer is fixed in the entire training. We take 71,319 of 171,440 articles with at least one comment received ten upvotes or above as our training set. We compute the score for comments from the remaining 100,121 articles using equation (6). The ensemble ratio α is set to 0.2. For the evaluation of models, we use the script from Coco Caption[2] [18].

5. RESULTS

We show our empirical results in Table 1. Overall, the generation-based approach outperforms IR in most of the metrics except METEOR. Adding a scorer in the IR-based model doesn't improve the retrieval performance. In contrast, the generation models such as pointer-generator networks receive advantages from using our proposed scorers. Our standard pointer-generator network achieves better performance regarding BLEU-1 and Rouge_L by around 2 points. We achieve the state-of-the-art by combining ES to the pointer-generator model.

Effectiveness of scorer: Table 2 is depicted to further analyze the results. We find that when we apply RS to the pointer-generator network, it starts to copy phrases from the articles and uses them to form a new comment. It shows that the model can preserve the relevance of the generated comments by utilizing RS. Notably, using upvote scorer (US) is

more effective than using raw upvote counts as the score, and it mitigates the issue when upvote information is not available in the unpopular articles. Moreover, after applying ES, the generated comments are more expressive and solve the mode collapse issue in Seq2Seq with attention model that often generates common phrases such as "我也是 (me too)" and "我也是醉了 (I am speechless)".

Retrieval vs. Generation: According to our observation, there are some trade-offs of choosing either IR or generation-based model. The comments from IR are always real human comments, and often they have very similar words to the articles. However, the candidates are very limited, and most of the time, they are irrelevant to the context. On the other hand, the sequences generated from the generation-based model learns how to decode important keywords and copies the words from the article text, and may produce the ungrammatical sequences. The proposed scorer helps the model to preserve the relevance and generate comments that are likely to be favored by humans. After applying the comment scorer, the performance increases significantly, and it can generate more relevant and attractive comments.

6. CONCLUSION

In this work, we present a novel framework for comment generation to leverage user-generated data and generate relevant comments according to the user preference. Our results show that user-generated information helps the generative model. We efficiently alleviate the mode collapse issue by incorporating upvote scorer and relevance score to our model and produce more meaningful comments. For future work, we plan to apply reinforcement learning to the comment generation by taking scores as the reward.

---

[1]news.sogou.com

[2]https://github.com/tylin/coco-caption

7. REFERENCES

[1] Lianhui Qin, Lemao Liu, Wei Bi, Yan Wang, Xiaojiang Liu, Zhiting Hu, Hai Zhao, and Shuming Shi, "Automatic article commenting: the task and dataset," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 151–156, Association for Computational Linguistics.

[2] Ji Ho Park and Pascale Fung, "One-step and two-step classification for abusive language detection on twitter," in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 41–45.

[3] Zhongyu Wei, Yang Liu, and Yi Li, "Is this post persuasive? ranking argumentative comments in online forum," in *ACL*, 2016.

[4] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein, "Unrolled generative adversarial networks," *ICLR*, 2017.

[5] 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.

[6] Sudha Rao and Hal Daumé III, "Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information," *arXiv preprint arXiv:1805.04655*, 2018.

[7] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes, "Reading wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 1870–1879, Association for Computational Linguistics.

[8] H. Zheng, W. Wang, W. Chen, and A. K. Sangaiah, "Automatic generation of news comments based on gated attention neural networks," *IEEE Access*, vol. 6, pp. 702–710, 2018.

[9] Varada Kolhatkar and Maite Taboada, "Constructive language in news comments," in *Proceedings of the First Workshop on Abusive Language Online*. 2017, pp. 11–17, Association for Computational Linguistics.

[10] Courtney Napoles, Joel Tetreault, Enrica Rosata, Brian Provenzale, and Aasish Pappu, "Finding good conversations online: The yahoo news annotated comments corpus," in *Proceedings of The 11th Linguistic Annotation Workshop*, Valencia, Spain, April 2017, pp. 13–23, Association for Computational Linguistics.

[11] Andrea Madotto, Chien-Sheng Wu, and Pascale Fung, "Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 1468–1478, Association for Computational Linguistics.

[12] Chien-Sheng Wu, Andrea Madotto, Genta Indra Winata, and Pascale Fung, "End-to-end dynamic query memory network for entity-value independent task-oriented dialog," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6154–6158.

[13] Chien-Sheng Wu, Richard Socher, and Caiming Xiong, "Global-to-local memory pointer networks for task-oriented dialogue," in *International Conference on Learning Representations*, 2019.

[14] Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 76–85, Association for Computational Linguistics.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.