Adversarial Ranking for Language Generation

Xiaodong He, Zhengyou Zhang, Ming-Ting Sun

31st Conference on Neural Information Processing Systems (NIPS 2017)

KWANGJE BAEG

2018/06/20

Introduction

- Language generation plays an important role in NLP: RNN, LSTM
  - machine translation, image captioning, dialogue systems
- Generating sequential synthetic data using GAN for sequence
  - Generator

    → Trained to confuse the discriminator by generating high quality synthetic data
  - Discriminator

    → Aims to distinguish the synthetic from the real data
  - Primarily, the GANs have difficulties in dealing with discrete data (e.g., text sequence)

    → Binary predication is too restrictive
- RankGAN: Generator + Ranker with Policy gradient
  - Generator: synthesize sentences which confuse the ranker so that machine-written sentences are ranked higher than human-written sentences
  - Ranker: rank the machine-written sentences lower than human-written sentences

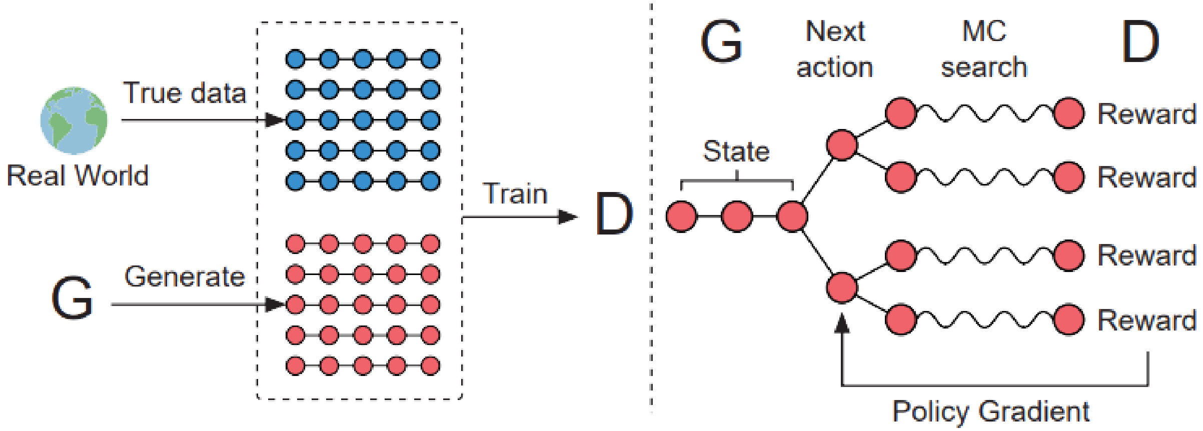
Background

- Policy gradient method
  - REINFORCE: Policy iteration algorithm using action-value function

    (Policy Gradient Methods for Reinforcement Learning with Function Approximation)
  - Maximize following expected end reward:

$$J(\theta) = \mathbb{E}[R_T|s_0,\theta] = \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1|s_0) \cdot Q_{D_\phi}^{G_\theta}(s_0, y_1)$$

- State = $\mathrm{s_t}(Y_{1:t-1})$: sequence of previously generated tokens
- Action = $\mathrm{a_t}(y_t)$: next token to generate
- Policy = $G_\theta(y_t|Y_{1:t-1})$: stochastic policy, form of RNN with weight θ
- State-Action value = $Q_{D_\emptyset}^{G\emptyset}(s,a)$: reward from discriminator $D_\emptyset$



- The major difference between SeqGAN and RankGAN
  - Regression based discriminator → A novel ranker, new learning objective function in the adversarial learning framework

Proposed Method (cont.)

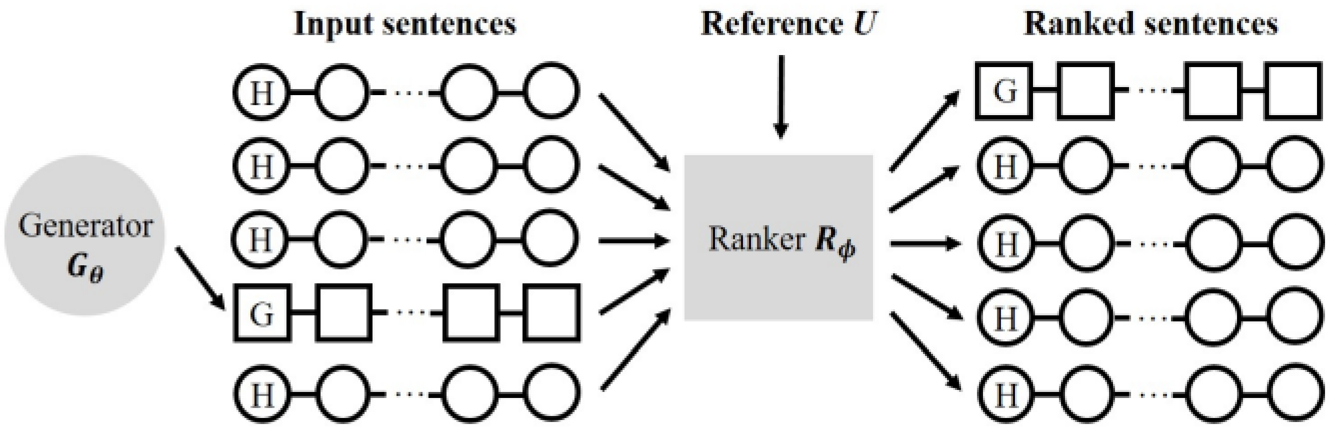


Figure 1: An illustration of the proposed RankGAN. **H** denotes the sentence sampled from the human-written sentences. **G** is the sentence generated by the generator G_θ. The inputs of the ranker R_ϕ consist of one synthetic sequence and multiple human-written sentences. Given the reference sentence U which is written by human, we rank the input sentences according to the relative scores. In this figure, it is illustrated that the generator tries to fool the ranker and let the synthetic sentence to be ranked at the top with respect to the reference sentence.

- RankGAN takes the ranking scores as the rewards to learn the language generator
- One of the first generative adversarial network which learns by relative ranking information

Proposed Method (cont.)

- Goal
  - G: Produce a synthetic sentence that receives higher ranking score than those drawn from real data
  - R : Rank the synthetic sentence lower than human-written sentences
  - G and R play a minimax game with the objective function L

$$\min_{\theta} \max_{\phi} \mathfrak{L}(G_\theta, R_\phi) = \mathop{\mathbb{E}}_{s \sim \mathcal{P}_h} \left[\log R_\phi(s|U, \mathcal{C}^-)\right] + \mathop{\mathbb{E}}_{s \sim G_\theta} \left[\log(1 - R_\phi(s|U, \mathcal{C}^+))\right]$$

- Argument
  - P_h: real data from human-written sentences
  - G_θ : synthesized sentences
  - U: reference set used for estimating relative ranks
  - C^-, C^+: comparison set with regard to different input sentences
    - C^-: When the input sentence s is the real data, pre-sampled from generated data
    - C^+ : If the input sentence s is the synthetic data, pre-sampled from human-written data
- Ranker R is consists of convolutional architecture

Proposed Method (cont.)

- Rank score
  - s: input sequence matrices, s = (w_0, w_1, w_2, ..., w_t)
  - u: reference from human-written sentences, $\mathcal{F}$: series of nonlinear functions
  - y_s: embedded feature vectors, $\mathcal{F}(s)$, y_u: embedded feature vectors, $\mathcal{F}$(u)

$$\alpha(s|u) = cosine(y_s, y_u) = \frac{y_s \cdot y_u}{||y_s||\,||y_u||}$$

  - Softmax-like formula is used to compute the ranking score

$$P(s|u,\mathcal{C}) = \frac{exp(\gamma\alpha(s|u))}{\sum_{s' \in \mathcal{C}'} exp(\gamma\alpha(s'|u))}$$

    - Lower γ : All sentences to be nearly equiprobable
    - Higher γ : Increases the biases toward the sentence with the greater score

  - Compute the rank scores indicating the relative ranks for the complete sentences

$$\log R_\phi(s|U,\mathcal{C}) = \mathop{\mathbb{E}}_{u \in U} \log\left[P(s|u,\mathcal{C})\right]$$

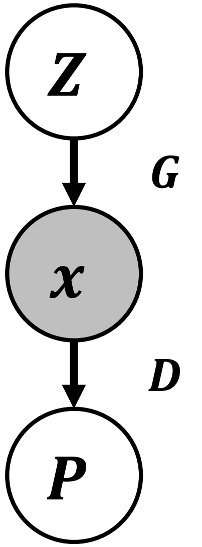
Generated by machine → C^+
Generated by human → C^-

Proposed Method (cont.)

- On continuous data, there is direct gradient

$$\nabla_\theta(G)\frac{1}{m}\sum_{i=1}^{m}\log(1-D\left(G\left(z^{(i)}\right)\right))$$

  - Guide the generator to (slightly) modify the output
    - G_θ consists of a series of differential functions with continuous parameters guided by the objective function from discriminator D_φ

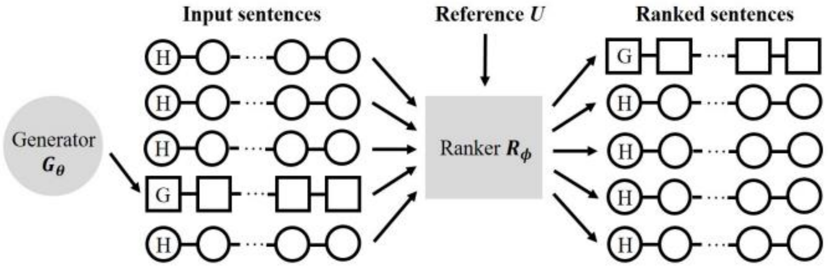


- No direct gradient on discrete data
  - the synthetic data in the text generation task is based on discrete symbols, which are hard to update through common back-propagation

    → To solve this issue : Policy Gradient method in reinforcement learning

Proposed Method (cont.)

- Training
  - V : Suppose the vocabulary set
  - t : time step
  - $(w_0, w_1, \ldots, w_{t-1})$: previous tokens generated in the sequence ($w_i \in$ V)
  - Compared to Reinforcement learning algorithm
    - Current state : $s_{1:t-1}$, $(w_0, w_1, \ldots, w_{t-1})$
    - Action : w_t (selected next step from the policy $\pi_\theta(w_t \mid s_{1:t-1})$
    - Reward : ranking reward R(s | U, C), (i.e., s = $s_{1:t}$)
      - Utilized the Monte Carlo rollouts methods to simulate intermediate rewards when a sequence is incomplete.
      - Expected future reward V for partial sequences can be computed by:



$$V_{\theta,\phi}(s_{1:t-1}, U) = \mathop{\mathbb{E}}_{s_r \sim G_\theta}\left[\log R_\phi(s_r | U, \mathcal{C}^+, s_{1:t-1})\right]$$

      - s_r : complete sentence sampled by rollout methods with the given starter sequence $s_{1:t-1}$

Proposed Method (cont.)

- Objective function
  - Gradient of the generator objective function
    - s_0 : first generated token w_0

$$\nabla_\theta \mathfrak{L}_\theta(s_0) = \mathop{\mathbb{E}}_{s_{1:T}\sim G_\theta}\left[\sum_{t=1}^{T}\sum_{w_t\in V}\nabla_\theta \pi_\theta(w_t|s_{1:t-1})V_{\theta,\phi}(s_{1:t},U)\right]$$

    - Note that we only compute the partial derivatives for θ, as the R_φ is fixed during the training of generator
    - Difference from SeqGAN
      - Replaced the simple binary outputs with a ranking system based on multiple sentences, which can better reflect the quality of the imitate sentences and facilitate effective training of the generator G

  - Ranking objective function

$$\min_\theta \max_\phi \mathfrak{L}(G_\theta, R_\phi) = \mathop{\mathbb{E}}_{s\sim\mathcal{P}_h}\left[\log R_\phi(s|U,\mathcal{C}^-)\right] + \mathop{\mathbb{E}}_{s\sim G_\theta}\left[\log(1 - R_\phi(s|U,\mathcal{C}^+))\right]$$

→ y = log(1 - x) : The gradient is relatively small at D(G(z))=0

$$\mathfrak{L}_\phi = \mathop{\mathbb{E}}_{s\sim\mathcal{P}_h}\left[\log R_\phi(s|U,\mathcal{C}^-)\right] - \mathop{\mathbb{E}}_{s\sim G_\theta}\left[\log R_\phi(s|U,\mathcal{C}^+)\right]$$

→ y = log(x) : The gradient is very large at D(G(z))=0

Experimental results (cont.)

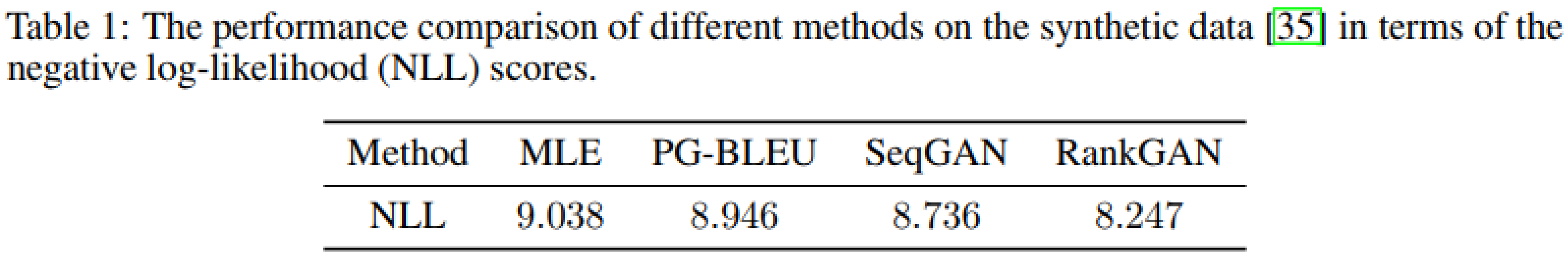
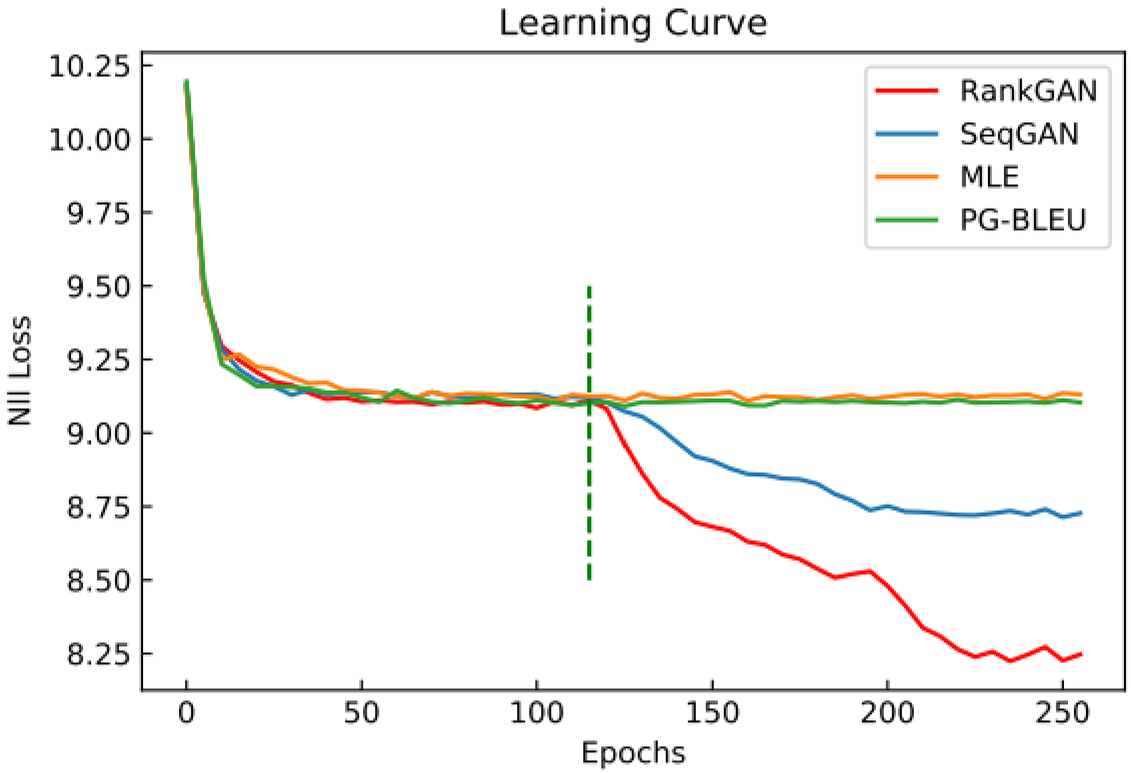


Table 1: The performance comparison of different methods on the synthetic data [35] in terms of the negative log-likelihood (NLL) scores.

Method	MLE	PG-BLEU	SeqGAN	RankGAN
NLL	9.038	8.946	8.736	8.247



Experimental results (cont.)

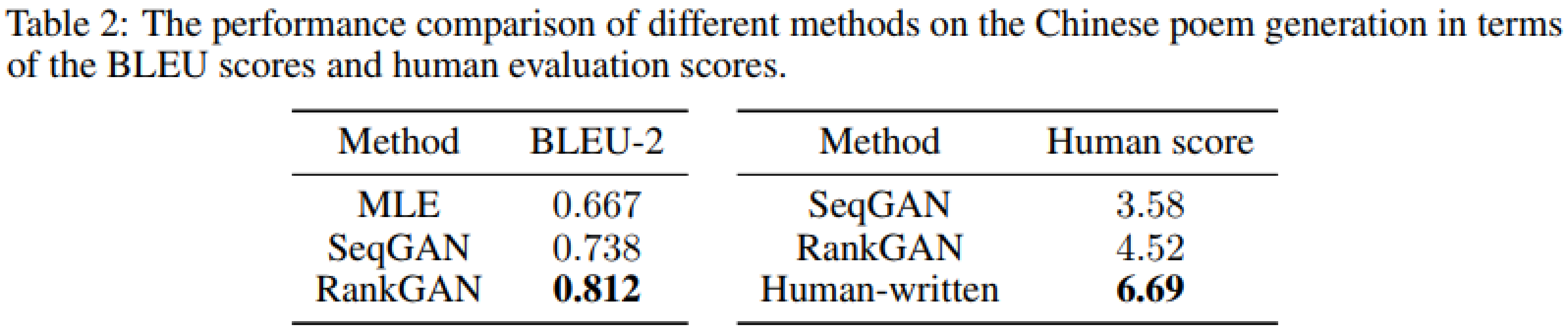


Table 2: The performance comparison of different methods on the Chinese poem generation in terms of the BLEU scores and human evaluation scores.

Method	BLEU-2
MLE	0.667
SeqGAN	0.738
RankGAN	**0.812**

Method	Human score
SeqGAN	3.58
RankGAN	4.52
Human-written	**6.69**

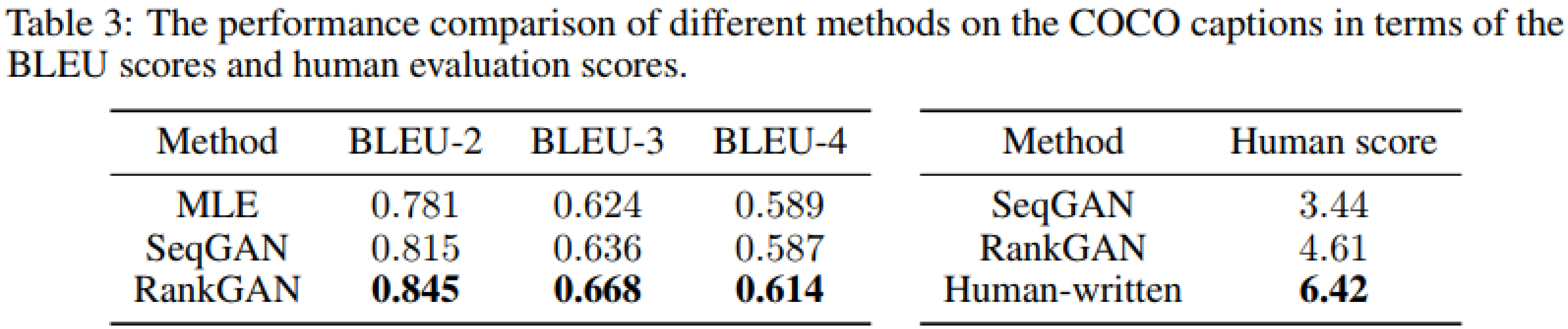


Table 3: The performance comparison of different methods on the COCO captions in terms of the BLEU scores and human evaluation scores.

Method	BLEU-2	BLEU-3	BLEU-4
MLE	0.781	0.624	0.589
SeqGAN	0.815	0.636	0.587
RankGAN	**0.845**	**0.668**	**0.614**

Method	Human score
SeqGAN	3.44
RankGAN	4.61
Human-written	**6.42**

Experimental results (cont.)

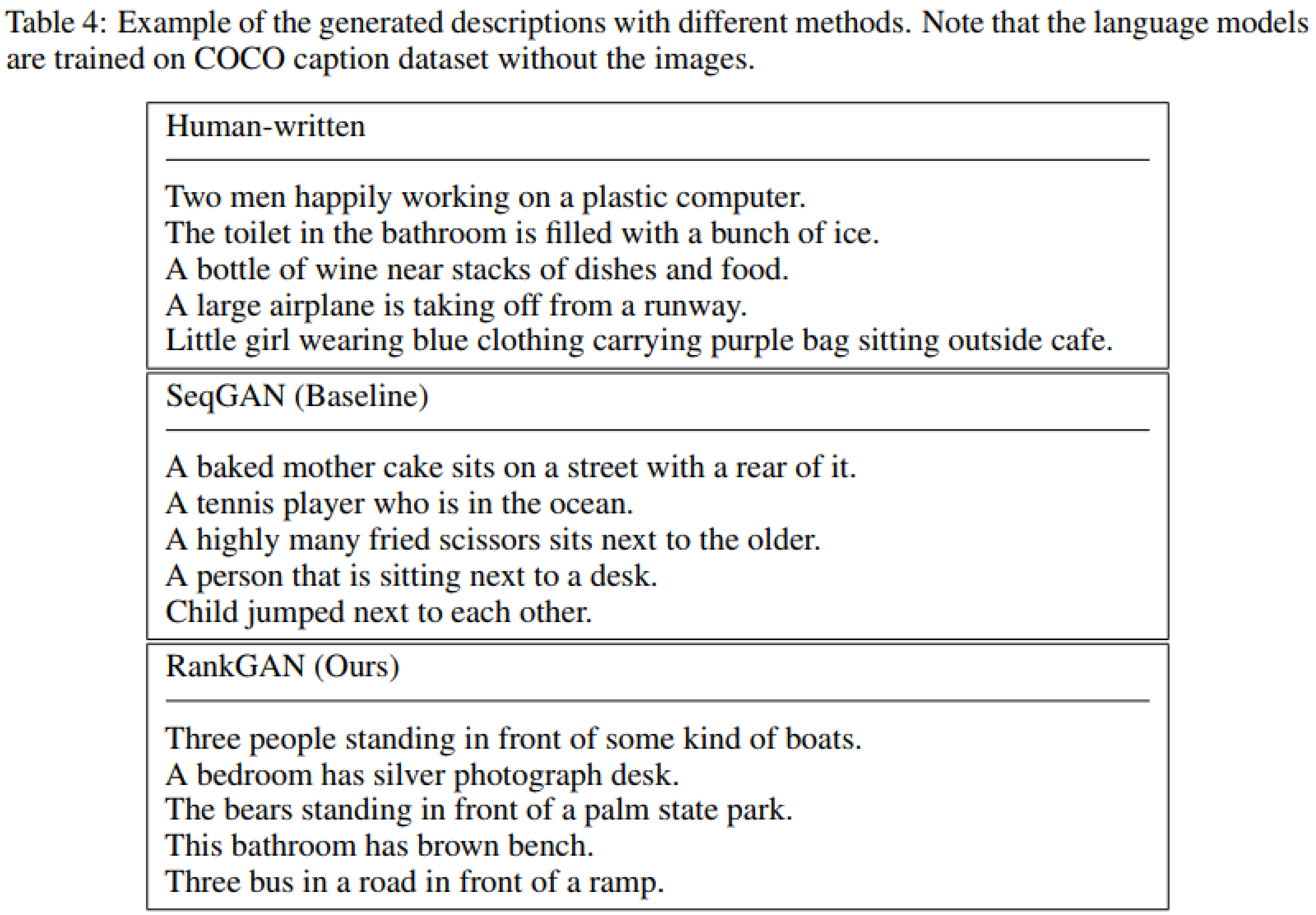


Table 4: Example of the generated descriptions with different methods. Note that the language models are trained on COCO caption dataset without the images.

Human-written
Two men happily working on a plastic computer.
The toilet in the bathroom is filled with a bunch of ice.
A bottle of wine near stacks of dishes and food.
A large airplane is taking off from a runway.
Little girl wearing blue clothing carrying purple bag sitting outside cafe.

SeqGAN (Baseline)
A baked mother cake sits on a street with a rear of it.
A tennis player who is in the ocean.
A highly many fried scissors sits next to the older.
A person that is sitting next to a desk.
Child jumped next to each other.

RankGAN (Ours)
Three people standing in front of some kind of boats.
A bedroom has silver photograph desk.
The bears standing in front of a palm state park.
This bathroom has brown bench.
Three bus in a road in front of a ramp.

Experimental results (cont.)

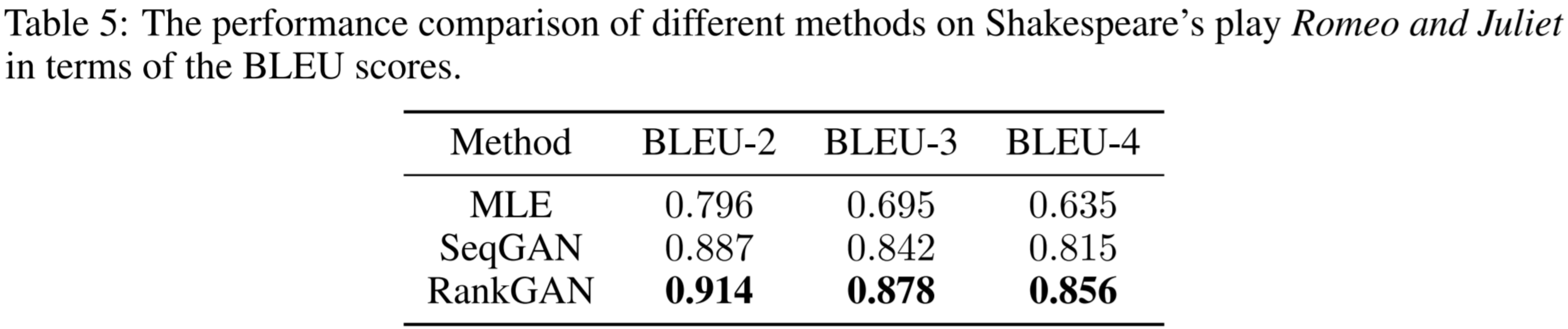


Table 5: The performance comparison of different methods on Shakespeare's play *Romeo and Juliet* in terms of the BLEU scores.

Method	BLEU-2	BLEU-3	BLEU-4
MLE	0.796	0.695	0.635
SeqGAN	0.887	0.842	0.815
RankGAN	**0.914**	**0.878**	**0.856**

Conclusion

- Presented a RankGAN, for generating high-quality natural language descriptions
- By relaxing the binary-classification restriction and conceiving a relative space with rich information for the discriminator in the adversarial learning framework, the proposed learning objective is favorable for synthesizing natural language sentences in high quality
- RankGAN showed better performance than previous work (SeqGAN)
- For future work, plan to explore and extend model in many other tasks, such as image synthesis and conditional GAN for image captioning