RETRIEVAL BIAS AWARE ENSEMBLE MODEL FOR CONDITIONAL SENTENCE GENERATION

Yiping Song[1], *Zheng Xie*[*1], *Jianping Li*[1], *Luchen Liu*[2], *Ming Zhang*[2], *Zhiliang Tian*[3]

[1] National University of Defense Technology, Changsha, China
[2] The School of Computer Science, Peking University, Beijing, China
[3] The Hong Kong University of Science and Technology, Hong Kong SAR, China
{songyiping,xiezheng81, jianpingli65}@nudt.edu.cn
{liuluchen,mzhang_cs}@pku.edu.cn, ztianac@cse.ust.hk

ABSTRACT

Conditional sentence generation aims to generate proper target sentences with the given condition, and has shown great promise in many text generation applications such as dialogue systems and poetry generation. The ensemble of retrieval and generation-based models retrieve texts according to the input condition to assist the generation-based model. Those approaches obtain great performance tasks as they can absorb both merits to generate informative and coherent sentences. However, the input condition and its retrieved results are usually not highly consistent due to the quality of retrieval. It leads to a retrieval bias between the condition and its retrieved result, and then text generation augmented by such results becomes unreliable. To fix this issue, we propose RBAEM, a Retrieval Bias Aware Ensemble Model. RBAEM employs two CVAEs (Conditional variational Auto-encoder) to represent the retrieved target and the ground truth with latent vectors, and then diminishes the bias by decreasing the distance of two corresponding distributions. The extensive experiments on two tasks show that the proposed methods excel the existing state-of-the-art generation models.

Index Terms— Generation, Ensemble, Retrieval Bias

1. INTRODUCTION

Conditional sentence generation is one of the basic tasks in the field of natural language processing. It aims to generate a proper target sentence under the constraints of the given condition. Many applications such as machine translation [1, 2], dialogue systems [3, 4, 5, 6] and poem generation [7, 8], are of substantial value both in industry and academia. Previous studies on this task can be roughly categorized into three groups: the retrieval-based methods, the generation-based methods, and the ensemble of these two.

Given the condition, the retrieval-based methods [9, 10, 11] retrieve the most relevant condition along with the corresponding target sentence. Since the corpus in the repository is from human beings, the retrieved sentences are usually vivid and informative, where the lack of informativeness is a major issue in dialog systems [12, 13, 14]. The generation-based methods [15, 16] tailor a specific target sentence for the given condition, so the generated sentences are highly consistent with the given condition. Recently, the ensemble of retrieval-based model and generation-based methods have become the mainstream for this task as it can absorb both merits [17, 18].

Some ensemble methods uses a re-ranker [18, 19] to combine the generation and retrieval models. Song et al. [18] use a GBDT-based classifier to score all the candidate sentences from both retrieval and generation modules. Tanaka et al. [19] sort these candidates using several hand-crafted features. The majority of works integrate the results of two methods more deeply [20, 17, 21, 22]. Tian et al. [20] propose to abstract the sentences of different clusters in the repository, and use the abstracted sentences to assist the target generation process. Zhang et al. [17] propose to successively feed the outputs of the condition encoder and the prototype (retrieved sentence) encoder to the decoder of the target sentence generation. These methods directly fuse the retrieved sentence into the generation model and ignore the fact that there remains an inconsistency between the given condition and the retrieved condition, which we call retrieval bias in this paper.

To consider the retrieval bias between the original inputs and their retrieved results, researchers proposed to edit the retrieved sentences by replacing or deleting the tokens. Wu et al. [23] proposed to construct an edit vector by explicitly encoding the lexical differences between the input text and its retrieved text. Cai et al. [24] extract the sentence skeleton according to the gap between the input and the retrieved results, and then revise the retrieved results by removing the tokens that mismatch the skeleton. Those methods focus on the token-level modifications but may be hard to model the non-lexical difference between the input and its retrieved text.

In this paper, we propose to exclude the retrieval bias between the given condition and retrieved condition, so it

*Corresponding author

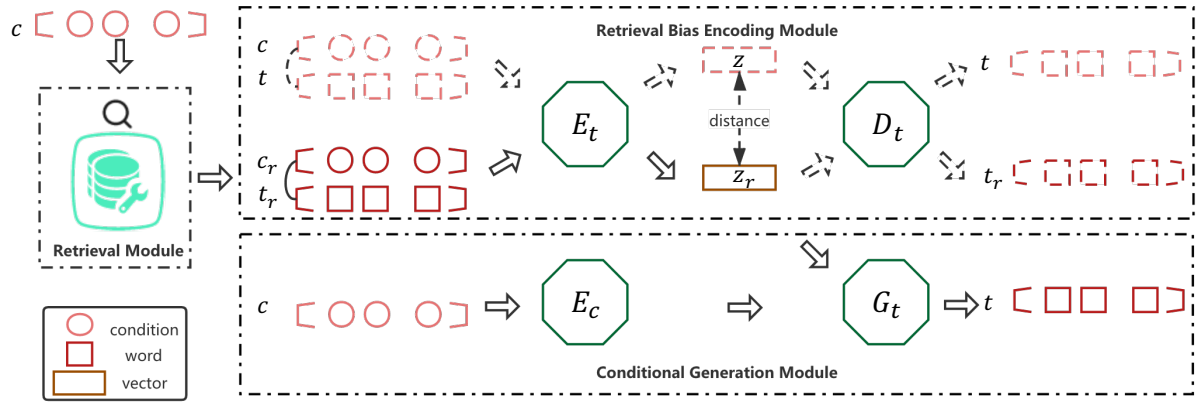

Fig. 1. The overall architecture of RBAEM, comprising of three modules. Given the condition c, the Retrieval Module retrieves sentence t_r, and the Retrieval Encoding Module maps the retrieved sentence into a latent vector z_r and diminishes the retrieval bias, then the Conditional Generation Module takes over both c and z_r for target sentence t generation. Dashed arrows indicate the parts involved in the training process, while the full arrows point out the parts involved both in training and inference.

can inspire the target generation in deep. This idea coincides with children's language acquisition. Research shows that understanding the similarities and differences between the sentences in different language backgrounds is far more important in the process of language acquisition [25]. Hence, we propose the Retrieval Bias Aware Ensemble Model (RBAEM). RBAEM first uses the given condition to retrieve the most relevant condition and the corresponding sentence in the repository, and then removes the retrieval bias by narrow down the distribution distance between the target sentence and retrieved sentence under the Conditional Variational Autoencoder (CVAE) framework [26, 11, 27], then generates the target sentence with the edited representation of the retrieved sentence. In this way, the value of the condition and retrieved sentence will be fully explored during the generation process.

2. METHODOLOGY

2.1. Overview

The main idea of Retrieval Bias Aware Ensemble Model (RBAEM) is to retrieve a sentence t_r under condition c, and generate target t by diminishing the retrieval bias in t_r. As shown in Figure 1, RBAEM contains three modules:

• *Retrieval Module.* Given the condition c, we retrieve the corresponding sentence t_r from repository. Our repository contains masses of condition-sentence pairs $\langle c_r, t_r \rangle$

• *Retrieval Bias Encoding Module.* We obtain the latent representation of t_r from the encoder E_t, and call the retrieval bias-aware latent representation as z_r.

• *Conditional Generation Module.* We then apply the generator G_t to generate the target sentence t using latent representation z_r and condition c.

2.2. Retrieval Module

This module uses the given condition c to retrieve the target sentence t_r from the repository, and t_r will further contributes to generating target sentence t.

Concretely, our pre-built repository is composed of condition-sentence pairs $\mathcal{P} = \{\langle c_1, t_1 \rangle, \langle c_2, t_2 \rangle, \cdots, \langle c_m, t_m \rangle\}$ (See details of dataset construction in Section 3.1.) To retrieve t_r, we follow popular retrieval approaches for dialogue systems [18] to leverage the textual similarities between input condition c and all candidate condition $\mathcal{C} = \{c_1, c_2, \cdots, c_m\}$, then the pair $\langle c_i, t_i \rangle$ is the retrieval result when $Sim(c_i, c)$ achieves the maximum value over all candidates, which is,

$$\mathrm{Sim}(c, c_i) = \cos(\{w_1, \cdots, w_p\}, \{w_1^i, \cdots, w_q^i\}) \quad (1)$$

where w is the words in sentence c. Here, the similarity operation $\mathrm{Sim}(\cdot)$ first average the word embeddings in c and c_r, then uses the cosine $\cos(\cdot)$ to calculate the cosine of two averaged vectors as the similarity score.

2.3. Retrieval Bias Encoding Module

This module encodes the retrieved target sentence t_r into z_r, and eliminates the inconsistency between the given condition c and retrieved condition c_r.

Specifically, this module is based on the CVAE (Conditional vVriational Auto-encoder), comprising of an encoder E_t and a decoder D_t. E_t takes two pairs $\langle c, t \rangle$ and $\langle c_r, t_r \rangle$, and encodes them into z and z_r respectively. Here, two training pairs go through the same encoder and the same decoder. To get a better latent representation z_r without retrieval bias, we then narrow down the distance between these two vectors, so z_r could not only retain the useful condition information of t_r but also eliminate the unfit part in c. We assume that z

and z_r subject to two different distributions: $z \sim P(z)$ and $z_r \sim P(z_r)$, and employ Kullback-Leibler divergence D_{KL} to measure the distance between them. So the loss here is,

$$J(KL) = \mathbf{D}_{\mathrm{KL}}(P(z) \parallel P(z_r))) \tag{2}$$

Next, the decoder D_t takes over two vectors z and z_r, and reproduce t and t_r to ensure the latent vectors retains enough information of original sentences, where standard cross-entropy loss over all words in the original sentences is applied as the training objective, which is

$$\begin{aligned} J(t) &= E_{z\sim P_\phi(z|c,t)}[\log Q_\theta(t|z)] \\ J(t_r) &= E_{z\sim P_\phi(z_r|c,t_r)}[\log Q_\theta(t_r|z_r)] \end{aligned} \tag{3}$$

where ϕ and θ are the parameters in G_t and G_t.

Here, E_t takes condition c and target sentence t as the input and a latent vector as the output. We have two types of conditions that need to be modeled in this paper, the discrete features and the sentences. For example, to generate a verse, we take the author name, dynasty, and keywords as the discrete features, and takes the former verse as the conditional sentence, and the goal is to generate the next verse based on these conditions. For all discrete features, we create a vocabulary for them and use a layer multilayer perceptron (MLP) to map the embeddings into a vector. We employ RNN with LSTM cells to model the sentences. We concatenate the last hidden state of RNN and the representing vector of features as the final representation for all the input. Notice that two pairs $\langle c, t\rangle$ and $\langle c_r, t_r\rangle$ use the same E_t and go through E_t twice for efficiency consideration. The decoder D_t aims to reproduce the original input sentence of E_t, we also use RNN with LSTM to generate the words one by one.

2.4. Conditional Generation Module

This module takes over the latent representation z_r of sentence t_r from the last module and generates the target sentence t. Here, we also use an embedding with a multi-layer perceptron or RNN with LSTM in E_c to encode the given condition c into z_c. The retrieval bias-aware representation z_r as well as encoded condition vector z_c are catenated together to feed to the generator G_t. G_t uses the same architecture of D_t but different parameters with the loss function,

$$J'(t) = E_{z_r\sim P_\phi(z_r|c)}[\log Q_{\theta'}(t|z_r, c)] \tag{4}$$

where z_r is the retrieval bias aware representation from E_r, and θ' is the parameters in G_t.

2.5. Training and Inference

The Retrieval Bias Encoding and Conditional Generation Module are trained jointly. The training loss is as follows:

$$\mathcal{L}_{\mathrm{RBAEM}} = J(t) + J(t_r) + J(KL) + J'(t) \tag{5}$$

During training, if we can not obtain a sentence t_r from the repository, we pre-train the model via sampling the vector z_r from the standard Gaussian distribution. During inference, we first retrieve t_r from the repository using condition c . Then t_r goes through Encoder E_t to attain the representation z_r, and c goes through Encoder E_c to attain the representation z_c, Finally, the conditional generation module G_t takes over z_r and z_c to generate the target sentence. The solid arrows in Figure 1 indicate the inference process.

3. EXPERIMENTS

3.1. Dataset

We use two datasets and Statistical details are showed in Table 2. One is Chinese poem corpus from a Chinese poetry website [1]. Each poem contains 4 or 8 verses with a fixed length, and we use the adjacent former verse as the condition (namely c) to generate later verse (namely t) in a poem. Besides, we also add the tags annotated by experts into the condition c for each target verse t, so the former verse and tags of the poem are both conditions for verse generation. Tags include the author name, poem dynasty, and other properties of the poem, such as "describing the emotion about homesick","describing a view", "describing a person". Another one is the real-world chatting corpus crawled from Weibo [2], a state-of-the-practice open-domain online chatting platform in Chinese. We form the training samples via using the query from the user as the condition c and using the response from the target user as the target sentence t. Except for the query, we also plug the keywords extracted from the chatting history that representing the wording habit of the target user into the condition c for response generation, so the condition comprises both query and keywords of the target user. We employ a popular Chinese sentence segmentation toolkit [3] for both queries and replies.

3.2. Hyperparameter Settings

Most of the hyperparameters are chosen by following [28] and [29] as they generally work well in sentence generation. The batch size was 50.Word embeddings, tags embeddings and hidden state in recurrent layers were 64. We optimized all the embeddings asynchronously by stochastic gradient descent.

3.3. Comparison Methods and Evaluation Metrics

• *Seq2Seq* is the basic Seq2Seq model [30] with an encoder-decoder framework. • *CVAE* [5] introduce conditional variational auto-encoder in text generation, which treats the input as the condition and reconstructs the target sentence. • *Ensemble* [18] uses an GBDT-based re-ranker to combine the

[1] www.gushiwen.org
[2] weibo.com
[3] github.com/FudanNLP/fnlp

Method	Poem Generation						Dialogue					
	Bleu1	Bleu2	Bleu3	Bleu4	Dist1	Dist2	Bleu1	Bleu2	Bleu3	Bleu4	Dist1	Dist2
Seq2Seq	10.56	2.27	0.75	0.12	0.035	0.071	15.61	5.39	1.04	0.31	0.064	0.091
CVAE	9.82	1.58	0.43	0.09	0.057	0.109	14.53	4.76	1.05	0.24	0.103	0.136
Ensemble	13.91	2.87	0.96	0.19	0.055	0.122	16.79	6.01	1.32	0.41	0.131	0.162
Edit-Merge	13.79	2.86	1.14	0.12	0.051	0.119	15.91	6.27	1.35	0.39	0.119	0.154
Skeleton	14.84	3.13	1.24	0.31	0.042	0.098	16.95	6.83	1.56	0.62	0.114	0.138
RBAEM−	15.91	3.41	1.35	0.46	0.055	0.102	**18.74**	7.10	**1.96**	0.73	0.152	0.187
RBAEM	**16.39**	**3.71**	**1.56**	**0.49**	**0.066**	**0.139**	18.63	**7.28**	1.89	**0.75**	**0.157**	**0.195**

Table 1. The performance of the two datasets on our model, RBAEM, with the baselines.

Size	**Poem dataset**	**Dialogue dataset**
Tag/Word Num	1125/2505	3015/10005
Train/Valid/Test	40000/5000/1000	420000/10000/5000

Table 2. Data statistics, including two datasets.

retrieval-based and generation-based models. • *Edit-Merge* [23] combines the token-level editing in prototypes with the re-ranking mechanism. • *Skeleton* [24] obtains the sentence skeleton according to the token difference between inputs and the retrieved results, and feeds the skeleton into the generation-based model with joint integration strategy. • *RBAEM−* is a simplified version of our full model, which does not diminish the distance between the target sentence t and retrieved target sentence t_r during the training. • *RBAEM* is the full model of this paper.

• *Bleu-N* measures the n-gram matching between the generated results with the ground-truth [6, 18, 7, 31]. • *Dist-N* (Distinct-1 and Distinct-2) [12] evaluate the diversity of generated responses by counting the percentage of unique unigrams and bi-grams among them.

3.4. Performance and Analysis

• *Poem Generation.*As shown in the left side of Table 1, Seq2Seq acts as a fundamental baseline for text generation. CVAE achieves a higher diversity than Seq2Seq but is weak in matching with ground truth. The reason is that the latent distribution introduces uncertainty in condition representation. Ensemble model further prompt the performance on both Bleu and Dist since it combines the retrieval-based model and generation-based model and employs a re-ranker. Edit-Merge and Skeleton also absorb the merits of retrieval-based and generation-based model and achieve further promotion by treating the retrieved sentence as a template then editing the words in the sentence with the generation model. They incorporate two models at the token level, which results in information fusion of two models. When considering the retrieval bias from the latent vector z of the target sentence and the retrieved target sentence, the Bleu scores increase sharply on our variant RBAEM−. This phenomenon verifies the effectiveness of considering the retrieval bias mechanism in poem generation. Our full model, RBAEM, outperforms all the baselines at most metrics. The improvements from RBAEM− to RBAEM also verifies that diminishing the retrieval bias contribute the text generation. It also indicates that our proposed strategy works effectively, which reduces the retrieval bias by decreasing the KL divergence of their corresponding latent vectors.

• *Dialogue Generation.* As shown in the right side of the Table 1, the performance on dialogue is quite similar to that on poem generation. CVAE is good at diversity and bad at overall quality, compared to Seq2Seq. Ensemble, Edit-Merge, and Skeleton perform much better than Seq2seq and CVAE. RBAEM still surpasses our other variant, RBAEM−, which shows diminishing the retrieval bias also helps on the dialogue. The gap between RBAEM and RBAEM− becomes smaller and RBAEM− sometimes works better than RBAEM. We guess the reason is that the semantic coherence between the query and response in dialogue is much lower than the coherence between the former verse and later verse in the poem, so the retrieved response may not contribute to the generation in some cases.

4. CONCLUSION

In this paper, we propose a retrieval bias aware ensemble model (RBAEM) for conditional sentence generation tasks. We notice the retrieval bias between the condition and the retrieved results. We first employ two CVAEs to represent the retrieved sentence and ground-truth sentence, and then diminish the retrieval bias by reducing their KL divergence. Such a strategy works more effectively than the existing ensemble models that only conduct the modification on lexical information (e.g. tokens). We verify our model's effectiveness on poem and dialogue generation tasks. The experimental results show that our approach outperforms other comparison methods in terms of quality and diversity evaluation metrics.

5. ACKNOWLEDGMENTS

This paper is supported by National Natural Science Foundation of China (NSFC Grant No. 62106275 and No.62106008).

6. REFERENCES

[1] Zhen Yang, Wei Chen, Feng Wang, and Bo Xu, "Improving neural machine translation with conditional sequence generative adversarial nets," *arXiv preprint arXiv:1703.04887*, 2017.

[2] Rakshith Shetty, Bernt Schiele, and Mario Fritz, "A4nt: Author attribute anonymity by adversarial training of neural machine translation," *arXiv preprint arXiv:1711.01921*, 2017.

[3] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan, "A neural network approach to context-sensitive generation of conversational responses," *ACL*, pp. 196–205, 2015.

[4] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," *AAAI*, pp. 3776–3783, 2016.

[5] Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, David Vandyke, and Steve Young, "Conditional generation and snapshot learning in neural dialogue systems," *EMNLP*, vol. 2153-2162, 2016.

[6] Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao, "How to make context more useful? an empirical study on context-aware neural conversational models," *ACL*, vol. 2, pp. 231–236, 2017.

[7] Marvin C Santillan and Arnulfo P Azcarraga, "Poem generation using transformers and doc2vec embeddings," in *IJCNN*. IEEE, 2020, pp. 1–7.

[8] Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, and Enhon Chen, "Chinese poetry generation with planning based neural network," *ICCL*, pp. 1051–1060, 2016.

[9] Lifeng Shang, Zhengdong Lu, and Hang Li, "Neural responding machine for short-text conversation," *ACL*, vol. 1, pp. 1577–1586, 2015.

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

[11] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[12] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan, "A diversity-promoting objective function for neural conversation models," in *NAACL*, 2016, pp. 110–119.

[13] Yiping Song, Zhiliang Tian, Dongyan Zhao, Ming Zhang, and Rui Yan, "Diversifying neural conversation model with maximal marginal relevance," in *IJCNLP*, 2017, pp. 169–174.

[14] Yu Cao, Liang Ding, Zhiliang Tian, and Meng Fang, "Towards efficiently diversifying dialogue generation via embedding augmentation," in *ICASSP*. IEEE, 2021, pp. 7443–7447.

[15] Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang, "Template-based question generation from retrieved sentences for improved unsupervised question answering," *arXiv preprint arXiv:2004.11892*, 2020.

[16] Rui Yan, Yiping Song, and Hua Wu, "Learning to respond with deep neural networks for retrieval-based human-computer conversation system," in *SIGIR*, 2016, pp. 55–64.

[17] Liang Zhang, Yan Yang, Jie Zhou, Chengcai Chen, and Liang He, "Retrieval-polished response generation for chatbot," *IEEE Access*, vol. 8, pp. 123882–123890, 2020.

[18] Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang, "An ensemble of retrieval-based and generation-based human-computer conversation systems.," *IJCAI*, 2018.

[19] Ryota Tanaka, Akihide Ozeki, Shugo Kato, and Akinobu Lee, "Context and knowledge aware conversational model and system combination for grounded response generation," *CSL*, vol. 62, pp. 101070, 2020.

[20] Zhiliang Tian, Wei Bi, Xiaopeng Li, and Nevin L Zhang, "Learning to abstract for memory-augmented conversational response generation," in *ACL*, 2019, pp. 3816–3825.

[21] Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu, "A hybrid retrieval-generation neural conversation model," in *CIKM*, 2019, pp. 1341–1350.

[22] Wei Wang, Hai-Tao Zheng, and Zibo Lin, "Self-attention and retrieval enhanced neural networks for essay generation," in *ICASSP*. IEEE, 2020, pp. 8199–8203.

[23] Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou, "Response generation by context-aware prototype editing," in *AAAI*, 2019, vol. 33, pp. 7281–7288.

[24] Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi, "Skeleton-to-response: Dialogue generation guided by retrieval memory," in *NAACL*, 2019, pp. 1219–1228.

[25] H Douglas Brown et al., *Principles of language learning and teaching*, Longman New York, 2000.

[26] Kihyuk Sohn, Honglak Lee, and Xinchen Yan, "Learning structured output representation using deep conditional generative models," *NIPS*, pp. 3483–3491, 2015.

[27] Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, Guodong Zhou, and Shuming Shi, "A discrete cvae for response generation on short-text conversation," in *EMNLP*, 2019, pp. 1898–1908.

[28] Andrej Karpathy, Justin Johnson, and Li Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv preprint arXiv:1506.02078*, 2015.

[29] Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin, "Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation," *ICCL*, pp. 3349–3358, 2016.

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[31] Jing He, Ming Zhou, and Long Jiang, "Generating chinese classical poems with statistical machine translation models.," *AAAI*, 2012.