Automated Thematic and Emotional Modern Chinese Poetry Composition

Xiaoyu Guo, Meng Chen, Yang Song, Xiaodong He, and Bowen Zhou

JD AI guoxiaoyu404@163.com
chenmeng20,songyang23,xiaodong.he,bowen.zhou@jd.com

Abstract. Topic and emotion are two essential elements in poetry creation, and also have critical impact on the quality of poetry. Inspired by this motivation, we propose a novel model to inject rich topics and emotions into modern Chinese poetry generation simultaneously in this paper. For this purpose, our model leverages three novel mechanisms including 1) learning specific emotion embeddings and incorporate them into decoding process; 2) mining latent topics and encode them via a joint attention mechanism; and 3) enhancing content diversity by encouraging coverage scores in beam search process. Experimental results show that our proposed model can not only generate poems with rich topics and emotions, but can also improve the poeticness of generated poems significantly.

Keywords: Poetry generation · Deep learning · Sequence to sequence model.

1 Introduction

As a fascinating writing art, poetry is an important cultural heritage in human history since its unique elegance and conciseness. Poetry generation has become a hot and challenging task in recent years which attracts lots of researchers in artificial intelligence field, partially because it's a typical study case for constrained Natural Language Generation (NLG) research. The past few years have witnessed lots of interesting approaches focusing on classical Chinese poetry generation including quatrain and lüshi [19, 23, 25], which have specific phonological or structural patterns to follow. Different from that, modern poetry is more flexible in style and length, and easier to understand and transmit. There are also a few works on modern Chinese poetry generation, such as XiaoIce [3]. XiaoIce generates poetry based on keywords extracted from pictures and uses a hierarchical model with two levels of LSTM [6] to maintain both the fluency of sentences and the coherence between sentences.

Although these methods can generate fluent Chinese poems with semantic consistency and tonal patterns, they do not give enough attention to topic and emotion, which are very essential and important aspects of human-created poems. Fig. 1 shows a poem written by a well-known poet Dai Wangshu[1]. We can see the poem has a specific topic on "lane in the rain" and reflects obvious "sadness" emotion. The emotion is expressed by words like "alone" and "solitary" that can deepen the topic expression. Correspondingly, there is a consistent emotion enhancement given by topic words like "oil-paper

[1] https://en.wikipedia.org/wiki/Dai_Wangshu

《雨巷》	/ *A Lane in the Rain*
撑着**油纸伞**，**独自**	/ **Alone** holding an **oil-paper umbrella**,
彷徨在悠长、悠长	/ I **wander** along a long
又**寂寥**的**雨巷**，	/ **solitary lane in the rain**,
我希望逢着	/ Hoping to encounter
一个**丁香**一样地	/ A **girl** like a bouquet of **lilacs**
结着**愁怨**的**姑娘**。	/ Gnawed by **anxiety** and **resentment**.

Fig. 1. An example of poems written by human.

unbrella" and "lilacs". Indeed, topic and emotion are complementary in poems and make poems impressive and imaginary together.

Motivated by this, we propose a novel model to generate modern Chinese poems with rich topic and emotion simultaneously. Inspired by methods dealing with conversation generation [7, 21], we modify the basic sequence to sequence [17] (seq2seq for short) framework. To incorporate distinct emotions and latent topics into poetry, we leverage embeddings for different emotions and a joint attention mechanism for various topics. Experimental results show an obvious increase in perception of emotion and topic in our generated poems. And by injecting both, the poeticness of generated poems is improved significantly. Our contributions are as follows:

- We achieve emotional poetry generation by adding embedding of distinct emotion categories to the decoder in basic seq2seq model.
- By mining latent topics from poems and adding a joint attention mechanism, we generate poems with distinct topics.
- We leverage a coverage decoder to enhance the effect of topic and emotion and generate long and diverse poems.

2 Background

Most of poetry generation works [4, 19, 25] are constructed from an attention based seq2seq model with LSTM. The encoder receives input sequence $X = \{x_t\}_{t=1}^{T}$ and converts it into hidden state sequence $H = \{h_t\}_{t=1}^{T}$, where h_t is:

$$h_t = LSTM(h_{t-1}, x_t). \tag{1}$$

According to the work [19], when generating the current line $l_c = \{w_{c,j}\}_{j=1}^{n}$, where $w_{c,j}$ is the j^{th} word of this line, they take former m lines as context and keywords of current line as inputs. Formally, the context is tokenized lines $L = \{l_{c-i}\}_{i=1}^{m}$, where $c-i > 0$, and keywords $K = \{k_{c,q}\}_{q=1}^{p}$ are extracted base on simple keyword extraction method (TFIDF or TextRank [14]), where $k_{c,q}$ is the q^{th} keywords of this line. These two parts form the total inputs, i.e. $X = L \cup K$. And during the decoding procedure, the decoder also uses LSTM and decoder states h'_s are updated as:

$$h'_s = LSTM(h'_{s-1}, [y_{s-1}; c_s]), \tag{2}$$

where h'_{s-1} and y_{s-1} are the previous LSTM hidden state and decoded word respectively and $[y_{s-1}; c_s]$ is the concatenation of the word embedding y_{s-1} and context vector c_s. c_s is computed as a weighted sum of h_t according to the work [1]:

$$c_s = \sum_{t}^{T} a_{st} h_t, \tag{3}$$

$$a_{st} = \frac{\exp(e_{st})}{\sum_{i=1}^{T} \exp(e_{si})}, \tag{4}$$

$$e_{st} = v_a^T \tanh(W_a h'_{s-1} + U_a h_t). \tag{5}$$

where a_{st} is the weight of h_t and e_{st} is an alignment model that scores the matching degree between h'_{s-1} and h_t. The probability of the next generated word will be given by:

$$p(y_s | y_1, y_2, ..., y_{s-1}, c_s) = g(h'_s), \tag{6}$$

where g is a nonlinear function. During inference, beam search is used to choose more reasonable words.

3 Proposed Model

3.1 Model Overview

Different from previous approaches, our model takes as encoder inputs not only tokenized history lines L and keywords K, but also emotion embeddings e of lines and topic representations t of poems. Fig. 2 shows our model structure of modern Chinese poetry generation. The lower bidirectional LSTM represents the encoder and the upper one denotes the decoder. To incorporate emotion information, emotion category embeddings e (oval regions with dotted line) are concatenated with each decoder cell input. To generate poems with topic, we mine latent topics and represent them as explicit topic words, then leverage a joint attention mechanism (adding topic attention) to better integrate topic information.

3.2 Generation with Emotion

To generate poems with emotion, we modify the decoder in Equation (2) to:

$$h'_s = LSTM(h'_{s-1}, [y_{s-1}; e; c_s]), \tag{7}$$

where $[y_{s-1}; e; c_s]$ is the concatenation of the word embedding y_{s-1}, emotion embedding e and context vector c_s. In this way, emotion information is incorporated (oval regions in Fig. 2). Note that the emotion embedding is a random initialized vector and corresponds to the emotion category of the current line, so all word embeddings concatenate with the same emotion embedding e.

In order to obtain emotion labels for poem lines, we employ a simple lexicon-based emotion classifier [12], which achieves 88% accuracy. Suppose that each line of poems

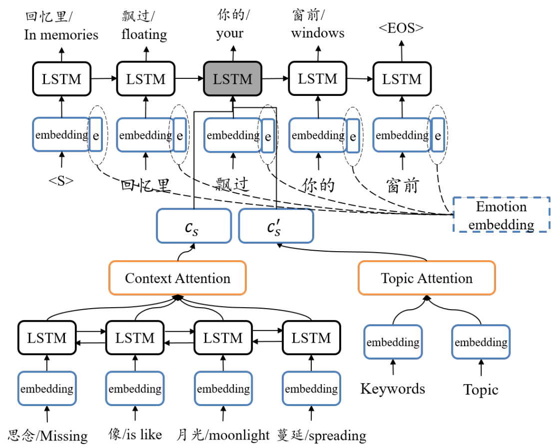

Fig. 2. The overall structure of our proposed poetry generation model.

can be classified into one of the following seven emotion categories: *Happiness*, *Anger*, *Sadness*, *Fear*, *Disgust*, *Surprise* and *Neutral*. We employ emotion vocabulary lists [22] that contain separate categories of emotion words. For emotion labeling, a line containing words from a specific emotion category will be assigned to the corresponding emotion category.

We also attempt to build a classifier on an annotated emotion dataset, consisting of NLPCC 2013[2]&2014[3] Chinese Weibo emotion recognition datasets, via fastText [8], but find it is unsuitable for our task (whose accuracy is only 78%). It may result from not only the severe asymmetry of NLPCC dataset (*Fear* (1.5%) and *Surprise* (4.4%) account for small proportions), but also the inconsistency of domains. Considering the emotions expressed in poems are often simple, we take the lexicon-based method in labeling.

3.3 Generation with Topic

To express specific topics in generation, we need to solve following three basic problems: how to mine latent topics from poetry corpus, how to represent latent topics, and how to incorporate topics into generation process.

Firstly, suppose that each poem can be assigned to a latent topic, and all topics are distinguished by ids. To realize generation with topic, we firstly use a framework combining LightLDA [24] and K-Means algorithm [2] to mine latent topics. LightLDA is an open source distributed system for large scale topic modeling and computes the topic distribution for each poem. By transforming the outputs of LightLDA into features, we adopt K-Means algorithm for poetry clustering and obtain latent topic ids for all poems.

[2] http://tcci.ccf.org.cn/conference/2013/dldoc/evsam02.zip
[3] http://tcci.ccf.org.cn/conference/2014/dldoc/evtestdata1.zip

Secondly, summarizing high frequent topic word list for each latent topic, we use explicit topic words via random or deliberate selection as latent topic representations. Random selection strategy assigns the same probability to topic words. In contrast, for deliberate selection, we firstly keep words of nouns, adjectives and adverbs, then sample topic words based on their frequencies.

Thirdly, we add another attention for topic to form a joint attention mechanism. Topic attention leverages hidden states of latent topic representations as topic embeddings together with keywords (topic attention in Fig. 2). As Equation (8) shows, both context and topic vectors can jointly impact the poetry generation:

$$h'_s = LSTM(h'_{s-1}, [y_{s-1}; e; c_s; c'_s]), \tag{8}$$

where c'_s represents the topic context vector and is calculated by

$$c'_s = \sum\nolimits_{w=1}^{W} a_{sw} e_w, \tag{9}$$

where e_w represents the embedding of keyword or topic word and W is the number of these words. Attention weights a_{sw} is computed by

$$a_{sw} = \frac{\exp{(e_{sw})}}{\sum_{i=1}^{W} \exp{(e_{si})}}. \tag{10}$$

The calculation of e_{sw} is similar to Equation (5).

3.4 Coverage Decoder

To generate poems with diverse content and enhance the expression of topic and emotion, we incorporate the coverage score into beam search procedure. Compared to the coverage model proposed by the work [18], coverage score includes no extra models. Moreover, it is applied to each decoding step instead of involving in complex reranking procedures [20]. In detail, we define the coverage score cs_t of t^{th} word in source sequence as the total sum of the past attention values [11]. Similarly, the coverage score cs_w of w^{th} word in keyword and topic word sequence is the sum of past topic attention values. Taking $Y = \{y_s\}_{s=1}^{S}$ as target sequence, cs_t and cs_w can be computed by

$$cs_t = \sum\nolimits_{s=1}^{S} a_{st} \tag{11}$$

$$cs_w = \sum\nolimits_{s=1}^{S} a_{sw}, \tag{12}$$

where a_{st} and a_{sw} are the context attention weight and the topic attention weight. And the coverage score of this sentence pair (X, Y) is defined by

$$c(X, Y) = \sum\nolimits_{t=1}^{T} \log \max(cs_t, \beta) + \sum\nolimits_{w=1}^{W} \log \max(cs_w, \beta), \tag{13}$$

where β is a hyper-parameter for model warm-up, which makes the model easy to run in the first few decoding steps [11]. For decoding, the total probability is modified as:

$$s(X, Y) = (1 - \alpha) \cdot \log P(Y|X) + \alpha \cdot c(X, Y), \tag{14}$$

where α is the linear interpolation coefficient and $\log P(Y|X)$ is the decoded word score from Equation (6).

4 Experiments

4.1 Dataset and Setup

We collect 263,870 modern Chinese poems and lyrics of songs that contain 9,211,510 lines in total. Then we use TFIDF to extract keywords. For all poems, we first tokenize each line into words, then take the 54,500 most frequently used words as our vocabulary. To construct our dataset, we hold out 10% for validation and 10% for automatic evaluation, and the rest 80% for training. For human evaluation, we sample 25 poems from evaluation set, and manually check their keywords. To train our model, we use Adam [9] optimizer with batch size set to 512 and learning rate set to 3e-4. Hyper-parameters are listed in Table 1. We tuned these hyper-parameters based on our validation set.

Table 1. Hyper-parameters.

Symbol	Meaning	Value
$\|x\|$	Word embedding dimension	128
$\|h\|$	Encoder hidden size	128
$\|h'\|$	Decoder hidden size	128
l	Number of LSTM layers	4
m	Number of context lines	2
$\|e\|$	Emotion embedding dimension	5
α	Coefficient parameter in coverage decoder	0.6
β	Warm-up parameter in coverage decoder	0.4
K	Number of poetry clusters in K-Means	25
r	Dropout rate	0.3

Note that word embeddings are pretrained by word2vec [15] with poetry corpus. And we choose the seq2seq+attention model mentioned in Section 2 as our baseline.

4.2 Evaluation Metrics

Automatic Evaluation We choose five automatic evaluation metrics:

Average sentence length (ASL): it reflects the average sentence length of poems. A higher ASL means poems are longer and contain more content.

Distinct-1/2: it measures whether poems are rich in content. A higher distinct-1/2 indicates a higher number of distinct unigrams/bigrams, which represents the information and diversity of poems.

Perplexity (PPL): it measures whether the generated poems are fluent and coherent or not. We train a 5-gram character based language model with poetry corpus to calculate PPL. A lower PPL indicates the generated poem is more fluent.

Emotion word hit-rate: it is the proportion of one specific emotion words to all emotion words in generated poems. A higher emotion word hit-rate indicates poems have stronger emotions.

Topic word hit-rate: it is the proportion of one specific topic words to all topic words in generated poems. A higher topic word hit-rate indicates poems are more thematic in this specific topic.

Human Evaluation We design five standards for human evaluation:

Fluency: it measures whether a single line is fluent. With grammar and syntax errors, a poem that cannot be smoothly read gains a lower score.

Coherence: it reflects the relevance among lines in a poem. If a poem expresses consistent content, it gains a higher coherence score.

Perception of Emotion: it represents the emotion intensity. The stronger emotion a poem owns, the higher perception of emotion score it obtains.

Perception of Topic: it denotes the topic intensity. The stronger topic a poem holds, the higher perception of topic score the poem gains.

Poeticness: it reflects the creativity of a poem in poetic aspect. A higher score means that it leaves a more striking impression on readers.

4.3 Results for Generation with Emotion

In inference time, we assign one emotion category each time and compare generated poems with baseline. Results are listed in Table 2: first row records emotion word hit-rate of poems generated by baseline, and all other rows record word hit-rate of poems with different emotions generated by our model. And each column denotes word hit-rate of a specific emotion. Taking the last row as an example, we generate poems in *Surprise* emotion and count emotion words separately based on emotion categories: these poems contain 0% *Anger* emotion words but 76.06% *Surprise* emotion words, so they express the most strongest *Surprise* emotion as expected.

Table 2. Emotion word hit-rate results with emotion embeddings.

Approaches	*Happiness*	*Anger*	*Sadness*	*Fear*	*Disgust*	*Surprise*
Baseline	.2117	.0061	.4172	.0399	.3098	.0153
+Happiness	**.8209**	.0081	.0843	.0183	.0650	.0034
+Anger	.1792	**.4309**	.0555	.1223	.2110	.0011
+Sadness	.0422	.0004	**.8654**	.0110	.0748	.0062
+Fear	.0696	.0251	.0743	**.7060**	.1226	.0024
+Disgust	.0507	.0076	.0910	.0131	**.8300**	.0076
+Surprise	.0372	0	.1050	.0040	.0932	**.7606**

From Table 2, we can see that when assigning a specific emotion category, the emotion word hit-rate of this category (1) is much higher than baseline, as bold numbers are much larger than the correspondent numbers of baseline, and (2) obviously dominates and exceeds emotion word hit-rate of others, as the diagonal line of Table 2 shows. Hence, it proves that our model can not only learn different emotion representations correctly but can also generate poems with specific emotion type correctly.

However, we also find that generation with certain emotions seems to cause emotion word hit-rate of others high. For example, when we generate *Anger* poems, the emotion word hit-rate of *Disgust* is also high. To address this, we have excavated several explanations. Firstly, we notice that “Sadness”, “Disgust” and “Happiness” occupy the most part in baseline, which means there exist some biases in training corpus, and they will affect the generation with emotions. Secondly, human emotions are complicated and sometimes concurrent. Thirdly, our weak emotion labeling approach may also bring some negative influences into generation procedure. Finally, all these three reasons may jointly impact the generation procedure and cause the emotion interaction.

4.4 Results for Generation with Topic

To evaluate generation with topic, we compare three different approaches with baseline.

Topic Attention with Latent Topic Ids (TA-ID): leveraging a joint attention mechanism and taking keywords and latent topic ids as topic attention inputs.

Topic Attention with Topic Words Randomly Selected (TA-RSW): replacing latent topic ids with randomly selected topic words as input for the topic attention.

Topic Attention with Topic Words Deliberately Selected (TA-DSW): using deliberately selected topic words as input instead of random ones.

We use topic ids to briefly represent latent topics from *Topic 1* to *Topic 25*. By assigning specific topics to these models, we generate poems and summarize the results of topic word hit-rate in Table 3. Different rows denote different models and different columns denote the assigned topics to models. For example, when generating with *Topic 10*, the TA-ID approach obtains 2.57% topic word hit rate on *Topic 10*.

Table 3. Topic word hit-rate results with latent topics.

Approaches	*Topic 1*	*Topic 5*	*Topic 10*	*Topic 15*	*Topic 20*	*Topic 25*
Baseline	.0346	.0189	.0193	.0251	.0235	.0221
TA-ID	.0565	.0335	.0257	.0329	.0290	.0385
TA-RSW	*.0859*	*.0701*	*.0516*	*.0710*	*.0678*	*.0760*
TA-DSW	**.1661**	**.1656**	**.1643**	**.2159**	**.1780**	**.1694**

From Table 3, we observe following interesting results. Firstly, compared with baseline, all our proposed three approaches improve the topic word hit-rate in different degrees, which suggests incorporating topics via joint attention mechanism is sufficiently feasible. Secondly, TA-RSW and TA-DSW perform better than TA-ID, which indicates that representing latent topics with explicit words is better than implicit topic ids. Thirdly, TA-DSW achieves the best performance and exceeds TA-RSW significantly by 10.61%, which proves our deliberate selection strategy for topic words is also effective. In a word, our model can generate more thematic poems, and all results are also consistent with our initial motivation in Section 3.3.

4.5 Results for Coverage Decoder

We compare beam search and coverage decoder in Table 4. And we can see that when using coverage decoder, ASL increases by 12.6%, and distinct-1/2 rise by 4.7% and 8.9% respectively. Meanwhile, PPL changes subtly. The results confirm that, coverage decoder can effectively help increase ASL and diversity of poems without sacrificing fluency.

Table 4. Comparison between beam search decoder and coverage decoder.

decoder	ASL	distinct-1	distinct-2	PPL
beam search	5.62	.383	.821	49.98
+coverage decoder	**6.33**	**.401**(+4.7%)	**.894**(+8.9%)	51.22

4.6 Human Evaluation

To further evaluate and understand our model from the point of emotion and topic, we generate 100 poems by four distinct models: baseline, model with emotion, model with topic and model with the both. For each model, we generate 25 poems with the same 25 groups of input keywords and get 100 poems in total. For emotion and topic generation, we assign a specific emotion and topic each time. All these poems are evaluated by 8 highly educated evaluators, poetry enthusiasts. Each evaluator needs to evaluate 100 poems. Five human evaluation standards as previously mentioned are scored between 1 to 5 and a higher score indicates better poem quality. And results are shown in Fig. 3.

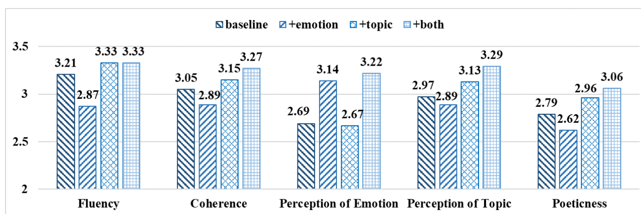

Fig. 3. Human evaluation in Fluency, Coherence, Emotion, Topic and Poeticness.

From Fig. 3, we can get following results. Firstly, the model with emotion obtains 0.45 higher than baseline in Emotion score, meanwhile, the model with topic achieves 0.16 higher than baseline in Topic score. This proves that generated poems are more emotional and thematic respectively. Secondly, we find the model with both emotion and topic outperforms the model with emotion or topic only, which proves that emotion and topic complement each other and enhance each other. Thirdly, model with both emotion and topic gains the overall highest Poeticness score, which proves that our initial motivation is correct that good poems should have topic and emotion simultaneously. Note that Fluency and Coherence scores of the model with emotion decrease because it may employ some highly frequent emotion words, which are possibly not suitable.

4.7 Case Study

We present three poems generated by different models but with the same keywords in Fig. 4 for case study. Fig. 4(a) shows a poem only with Sadness emotion since there are words like “missing”, “disappear” and “memories”. As for Fig. 4(b) with automatically selected topic, it seems to be more colorful and ample in topic, while moderate in emotion. The last one in Fig. 4(c) represents a poem generated with both Sadness emotion and automatically selected topic, which strengthens not only the expression of Sadness emotion, but also the topic of missing. In conclusion, our proposed model can generate creative and impressive poems with rich topic and emotion. We also realize that sometimes there exist some over-expressed issues in generated poems, which may be alleviated by using some post-editing techniques and considering basic composition structures. And there are more complex emotions other than seven emotion types, such as a compounded of Happiness and Surprise. We will try out these ideas in the future.

回首过往云烟
Looking back at the past.
留下的回忆如蜜
Memories look like honey.
如今走进梦里
Now I walked into dreams.
只有对你的思念
There is only **missing** left in my heart.
离别的云儿飘过
Clouds drift away,
如今已经消失
now **disappear**,
如今化为云烟
and become light smoke eventually.
谁为我心动我心弦
Who is touched by me?

(a) Emotion: sadness, Topic: none

往事成云烟
Past events becomes clouds.
你出现在我面前
You appear by my side.
想带你走进我的心里面
I want to take you into my heart.
让月光洒向你的眼
Let the **moon** shine on your eyes
像白云飘过眼前
like pure clouds drifting over.
美丽的桃花就在眼前
Beautiful **peach blossoms** are right here
如云烟飞逝如轻烟
like float clouds and flying smoke
你的琴声也深深牵动我心弦
Your **piano sound** deeply touches my heart.

(b) Emotion: none, automatic selected Topic

前尘往事剥落成云烟
The **former dusty** flakes into clouds.
深夜里谁在悼念
Who is **mourning** in the dark night?
如今我走进了从前
It seems that I walked into the past:
思念像月光在蔓延
missing spreads like the **moonlight**.
回忆里飘过你的窗前
I **remember** standing outside your window.
潺潺的流水如今消失不见
Flowing water disappears now.
回想那往事如云烟
Recall that the **past** is like a cloud
别离的小雨拨乱我的心弦
The **final rain** disturbed my heartstrings.

(c) Emotion: sadness , automatic selected Topic

Fig. 4. Examples generated by models with emotion, topic and both. The keywords are identical.

5 Related Work

Poetry generation is a vital task in NLG. Considering semantics and textual structure, [16] proposes a poetry generation system based on a set of grammar rules, sentence templates and strategies. The second type of researches is based on genetic algorithms [13], which aims to meet the restricted poetry properties including grammaticality, meaningfulness and poeticness. Besides, there are approaches guided by statistical machine translation methods [5].

With increasing popularity of deep learning methods, neural network based methods have been proved to be valid to deal with this problem. [25] utilize a recurrent neural network (RNN) to jointly learn content selection and surface realization. [19] propose a two-stage poetry generation method which first plans keywords for poems and then generates poem lines according to these keywords based on seq2seq model. By integrating a finite state acceptor with basic RNN, [4] propose a method that allows users to revise and polish the generated poems in different styles. And [23] propose a SPG model to generate stylistic poems.

As for incorporating emotions and topics in NLG, there exists some similar studies in conversation generation. [10] propose a persona-based model to encode personas and speaking styles of different speakers, and then response consistently during multiple conversational interactions. Considering incorporating topic information into generation procedure, [21] leverage prior topic knowledge and make use of a joint attention mechanism to generate more informative and interesting responses. [26] propose a more complete method to express emotional responses by emotional chatting machine (ECM), which can generate proper responses in both content and emotion. Facing real customer care conversation problem, [7] create a tone-aware model with tone indicator added for generating not only grammatically correct, but also highly user experienced responses. Different from previous studies, we propose a poetry generation model incorporating topic and emotion simultaneously and use coverage decoder to enhance.

6 Conclusion

In this paper, we propose a novel model for modern Chinese poetry generation. To generate poems with rich topic and emotion, we employ mainly three mechanisms, including emotion embeddings, a joint attention mechanism and a coverage decoder. Both automatic and human evaluation results show our model can generate long and diverse poems not only with specific emotions, but also with rich topics. And the poeticness of generated poems is also improved a lot. Considering poets usually have refinement process during creation, we will explore some automatic post-editing techniques to further improve the poem quality. Apart from that, we're also trying to learn the basic composition thinking of poets, to bring some content structures (introduction, elucidation, transition and summing up) into poetry generation. We will explore these ideas in future.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2014), version 7
2. Bui, Q.V., Sayadi, K., Amor, S.B., Bui, M.: Combining latent dirichlet allocation and k-means for documents clustering: Effect of probabilistic based distance measures. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) Intelligent Information and Database Systems. pp. 248–257. Springer International Publishing, Cham (2017)
3. Cheng, W.F., Wu, C.C., Song, R., Fu, J., Xie, X., Nie, J.Y.: Image inspired poetry generation in xiaoice. CoRR **abs/1808.03090** (2018), version 1
4. Ghazvininejad, M., Shi, X., Priyadarshi, J., Knight, K.: Hafez: an interactive poetry generation system. In: Proceedings of ACL 2017, System Demonstrations. pp. 43–48. Association for Computational Linguistics (2017)
5. He, J., Zhou, M., Jiang, L.: Generating chinese classical poems with statistical machine translation models. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. pp. 1650–1656. AAAI'12, AAAI Press (2012)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
7. Hu, T., Xu, A., Liu, Z., You, Q., Guo, Y., Sinha, V., Luo, J., Akkiraju, R.: Touch your heart: A tone-aware chatbot for customer care on social media. CoRR **abs/1803.02952** (2018), version 2
8. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics (April 2017)

9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), version 1
10. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A persona-based neural conversation model. CoRR **abs/1603.06155** (2016)
11. Li, Y., Xiao, T., Li, Y., Wang, Q., Xu, C., Zhu, J.: A simple and effective approach to coverage-aware neural machine translation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 292–297. Association for Computational Linguistics (2018)
12. Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2012)
13. Manurung, R., Ritchie, G., Thompson, H.: Using genetic algorithms to create meaningful poetic text. Journal of Experimental & Theoretical Artificial Intelligence **24**(1), 43–64 (2012)
14. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing (2004)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013), version 3
16. Oliveira, H.G.: Poetryme: a versatile platform for poetry generation. Computational Creativity, Concept Invention, and General Intelligence **1**, 21 (2012)
17. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. pp. 3104–3112 (2014)
18. Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H.: Modeling coverage for neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 76–85. Association for Computational Linguistics (2016)
19. Wang, Z., He, W., Wu, H., Wu, H., Li, W., Wang, H., Chen, E.: Chinese poetry generation with planning based neural network. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 1051–1060. The COLING 2016 Organizing Committee (2016)
20. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR **abs/1609.08144** (2016), version2
21. Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W.: Topic augmented neural response generation with a joint attention mechanism. CoRR **abs/1606.08340** (2016)
22. Xu, L., Lin, H., Pan, Y., Ren, H., Chen, J.: Constructing the affective lexicon ontology [j]. Journal of the China Society for Scientific and Technical Information **2**, 006 (2008)
23. Yang, C., Sun, M., Yi, X., Li, W.: Stylistic chinese poetry generation via unsupervised style disentanglement. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3960–3969 (2018)
24. Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E.P., Liu, T.Y., Ma, W.Y.: Lightlda: Big topic models on modest computer clusters. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1351–1361. International World Wide Web Conferences Steering Committee (2015)
25. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 670–680. Association for Computational Linguistics, Doha, Qatar (October 2014)
26. Zhou, H., Huang, M., Zhang, T., Zhu, X., Liu, B.: Emotional chatting machine: Emotional conversation generation with internal and external memory. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18). pp. 730–739 (2018)