Applied Intelligence (2019) 49:44–52
https://doi.org/10.1007/s10489-018-1206-2

Hierarchical attention based long short-term memory for Chinese lyric generation

Xing Wu[1] · Zhikang Du[2] · Yike Guo[1] · Hamido Fujita[3]

Published online: 15 June 2018

Abstract

Automating the process of lyric generation should face the challenge of being meaningful and semantically related to a scenario. Traditional keyword or template based lyric generation systems always ignore the patterns and styles of lyricists, which suffer from improper lyric construction and maintenance. A Chinese lyric generation system is proposed to learn patterns and styles of certain lyricists and generate lyrics automatically. A long short-term memory network is utilized to process each lyric line and generate the next line word by word. A hierarchical attention model is designed to capture the contextual information at both sentence and document level, which could learn high level representations of each lyric line and the entire document. Furthermore, the LSTM decoder decodes all the semantic contextual information into lyric lines word by word. The results of the automatically generated lyrics show that the proposed method can correctly capture the patterns and styles of a certain lyricist and fit into certain scenarios, which also outperforms state-of-the-art models.

Keywords Chinese lyric generation · Long short-term memory · Hierarchical context attention · Contextual representation

1 Introduction

For many centuries, music has made great effort to express life experiences, such as personal emotions, stories, love and hope. Besides rhythms, lyrics also play an important role in music. Each lyricist has their own patterns and styles. Automating the process of lyric generation should face and meet the challenge of being meaningful and semantically related to a certain scenario. Traditional automatic lyric generation systems always go through the following process as defining keywords, choosing a template and generating lyric lines word by word using ontologies or other kinds of human defined rules, which ignore the patterns and styles of certain lyricists and suffer from improper lyric construction and low style maintenance.

To learn the patterns and styles of Chinese lyricists automatically and generate lyrics with full contextual support, a Chinese lyric generation system is built with the state-of-the-art deep learning algorithms. A long short-term memory (LSTM) encoder and decoder network is utilized to process each lyric line and generate next line word by word. A hierarchical context attention model is designed to capture the contextual information at both sentence and document level by learning a high level contextual representation of every lyric line and the entire lyric document. All the contextual attentions are then fed into the LSTM decoder to generate lyrics with contextual support automatically.

For the training process, a lyric with multiple lines is fed into the proposed model at a time to capture the statistic patterns and styles of the certain lyricist into the network memory. For lyric generations, the first lyric line in Chinese is input by users to generate following lyrics which relate to a certain scenario. The proposed system generates the lyrics line by line and word by word. The contextual attention of the entire generated lyric lines is calculated in each

✉ Xing Wu
xingwu@shu.edu.cn

Zhikang Du
duzhikang@shu.edu.cn

Yike Guo
y.guo@imperial.ac.uk

Hamido Fujita
HFujita-799@acm.org

[1] School of Computer Engineering and Science, Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China

[2] School of Computer Engineering and Science, Shanghai University, Shanghai, China

[3] Intelligent Software Systems Laboratory, Iwate Prefectural University, Iwate, Japan

generation step and then used to generate the next lyric line, which stops at the end token. Compared with the previous research works, the proposed system does not rely on other kinds of techniques and resources like ontologies, rhymes, templates or word frequencies. Furthermore, it has easier implementation and better understanding of the contextual information of lyrics than state-of-the-art methods.

2 Related works

A multitude of methods of automatic lyric generation have been proposed in recent years. Most of the traditional methods are based on keywords, rhymes and templates, which try to obtain the semantics of lyrics by using some rule-based constraints. Barbieri et al. [1] utilized the constrained Markov process and rhyme templates to generate lyrics with style. Rajeswari et al. [2] proposed an ontology-based method for interpreting semantics of lyrics and generate Tamil lyrics with the n-gram model. Karteek et al. [3] used an unsupervised Hidden Markov Model (HMM) to identify rhyme schema in hip hop lyrics. These methods took a lot of rules and constraints into consideration, which suffer from both construction and maintenance. Watanabe et al. [4, 5] also proposed a topic transition model based on HMM for lyric generation. All the above mentioned methods are based on the surface forms of words or characters, with barely deep understanding of the meaning of a poem or lyric.

Recent works have shown the effectiveness of deep learning [6] methods for automatic text generation. Mikolov et al. [7] showed that the recurrent neural network (RNN) based language models outperform the standard back-off n-gram models. Sutskever et al. [8] used recurrent neural network to generate text at character level, which significantly learns the grammatical and punctuation rules of language. These RNN models are widely used for the generation of lyrics and poetries recently. Potash et al. [9] proposed a rap lyric generation system using LSTM language model [10]. Zhang et al. [11] used the recurrent context and generation to capture the context information and generate Chinese poetries. These kinds of methods are quite similar, which still lacks some document level contextual support. An encoder and decoder network model (also known as sequence to sequence model) [12] is able to take in a sequence of words and generate another word sequences, which is widely used in neural machine translation [13] recently. The attention mechanism proposed by Luong et al. [14] shows a great effectiveness in capturing the contextual attention of sentences, which it is now widely used in many deep learning approaches. Hu Z et al. [15] propose a new neural generative model which combines variational auto-encoders (VAEs) and holistic attribute discriminators for effective imposition of semantic structures, to generate plausible text sentences, whose attributes are controlled by learning disentangled latent representations with designated semantics.

Compared with previous research works, our proposed method extends the attention mechanism to a higher document level, combining with the LSTM encoder and decoder network to generate lyric lines automatically. With the extended contextual attention model, the system can capture both the semantic meanings of sentences and the entire generated lyric document. The LSTM decoder decodes all the semantic contextual information into lyric lines word by word, which shows great advantage for the lyric generation.

3 Model structure

This section shows the detailed construction of the hierarchical context attention based lyric generation model, as shown in Fig. 1. There are five key components in our model: an LSTM sentence encoder (with a built-in word embedding layer), a sentence-level context attention model, an LSTM lyric encoder, a hierarchical lyric-level context attention model and an LSTM sentence decoder.

3.1 LSTM sentence encoder

The LSTM sentence encoder is designed to encode the input words (i.e. one lyric line), denoted by the embedding of compositional words, into a sequence of hidden states. The LSTM uses a gating mechanism to track the states of the sequence as context information. There are four key components in an LSTM cell: an input gate i_t, a forget gate f_t, a self-recurrent connection C_t and an output gate o_t which control the information update of the state all together. At time t, the values of i_t, f_t, C_t are computed as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$\widetilde{C_t} = tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$

$$C_t = i_t * \widetilde{C_t} + f_t C_{t-1} \tag{4}$$

Here W, U, V represent the weight matrices of the LSTM network, and b is the bias vector. The output value and hidden state of are then computed as follows:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

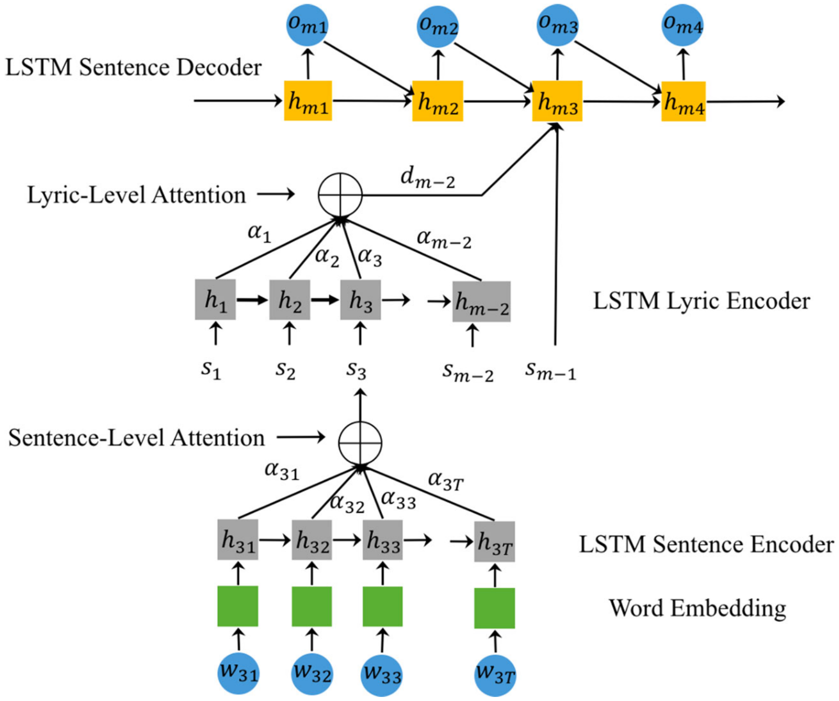


Fig. 1 The structure of the lyric generation model

The hidden states and output values of the LSTM sentence encoder represent how much context information each word contributes to the whole sentence.

3.2 Sentence-level context attention model

The attention mechanism provides a general way to group all the hidden states of a sentence to form a contextual attention vector. A sentence i with words $w_{it}, t \in [1, T]$ is firstly embedded into a sequence of word vectors using an embedding matrix W_e, and then fed into the LSTM sentence encoder to obtain the hidden states. The computation can be simplified as follows, where the LSTM calculation is based on Section 3.1:

$$x_{it} = W_e w_{it} \tag{7}$$

$$h_{it} = LSTM_h(x_{it}) \tag{8}$$

The contextual attention vector s_i can be computed as follows:

$$s_i = \sum_t^T \alpha_{it} h_{it} \tag{9}$$

Here α_{it} is the weight of each hidden state, which is derived by comparing the word embedding with the current hidden state. The α_{it} is computed as follows:

$$\alpha_{it} = \frac{e^{h_{it}^\top x_{it}}}{\sum_t^T e^{h_{it}^\top x_{it}}} \tag{10}$$

The contextual attention vectors of each sentence are calculated with the above equations and then fed into the lyric-level context attention model.

3.3 Hierarchical lyric-level context attention model

With the contextual attention vector and hidden states of a sentence, the next sentence can be generated word by word using the LSTM sentence decoder. However, the generated sentence only has the contextual information of its former input sentence. Since the system always needs to generate multiple lines of lyrics, obtaining the contextual information of every generated sentence would be very useful for the entire lyric generation process.

The contextual attention vector at lyric document level can be computed using the LSTM lyric encoder, which takes in the attention vector of the former sentences to obtain the hidden state and output value of each sentence. For the generation of the sentence m, the former $m-1$ sentences are used. The contextual attention vector of the former $m-2$ generated lyric lines can be computed as follow, which is similar to the computation of the contextual attention vector of sentences.

$$h_i = LSTM_h(s_i), i \in [1, m-2] \tag{11}$$

$$\alpha_i = \frac{e^{h_i^\top s_i}}{\sum_i^{m-2} e^{h_i^\top s_i}} \tag{12}$$

$$d_{m-2} = \sum_i^{m-2} \alpha_i h_i \tag{13}$$

For every generation step, the lyric-level context attention vector of the former lyric lines is calculated and then used to predict the next line of lyric.

3.4 LSTM sentence decoder

Given an input vector, the LSTM sentence decoder is used to generate a lyric line word by word. To make full use of the information of attention context, the input vector is calculated using the concatenation of the lyric context attention of last $m-2$ sentences d_{m-2}, the attention context vector of the former sentence s_{m-1}, and the embedding vector of last generated word x_{mt}. The hidden states and the output values of the LSTM decoder are computed using the LSTM cells:

$$h_{mt} = LSTM_h([d_{m-2} : s_{m-1} : x_{mt}]) \tag{14}$$

$$o_{mt} = LSTM_o([d_{m-2} : s_{m-1} : x_{mt}]) \tag{15}$$

A fully connected layer (16) is utilized to map each output of the LSTM decoder into a vector of vocabulary size, and a softmax classifier (17) is used to predict the most appropriate word.

$$u_{mt} = W_e^\top o_{mt} + b_e \tag{16}$$

$$p_{mt} = \frac{e^{u_{mt}}}{\sum_t^T e^{u_{mt}}} \tag{17}$$

As it is shown in (16), the weight matrix of the fully connected layer is tied with the embedding matrix to map the word vector to the appropriate word index, which shows a better performance than the re-training of this layer.

4 Datasets and experiments

4.1 Datasets

The experiments are carried out on four famous Chinese artists, which are Eason Chan, Jay Chou, Teresa Teng, and Faye Wong. The number of song lyrics and total lines for each artist are shown in Table 1.

Table 1 The number of song lyrics used from each singer

Name	Number of lyrics	Total lines
Eason Chan	320	11218
Jay Chou	183	8927
Teresa Teng	310	8220
Faye Wong	312	9842

4.2 Comparison models

To verify the effectiveness of the proposed hierarchical attention based model, and offer fair comparisons between different models, a series of experiments are conducted on these datasets.

Lyric-No-Attention (LNA) A basic LSTM encoder-decoder network without attention. The model takes in the previous line of lyric as input and generate the next line word by word. No attention mechanism is used in the model to obtain contextual attention.

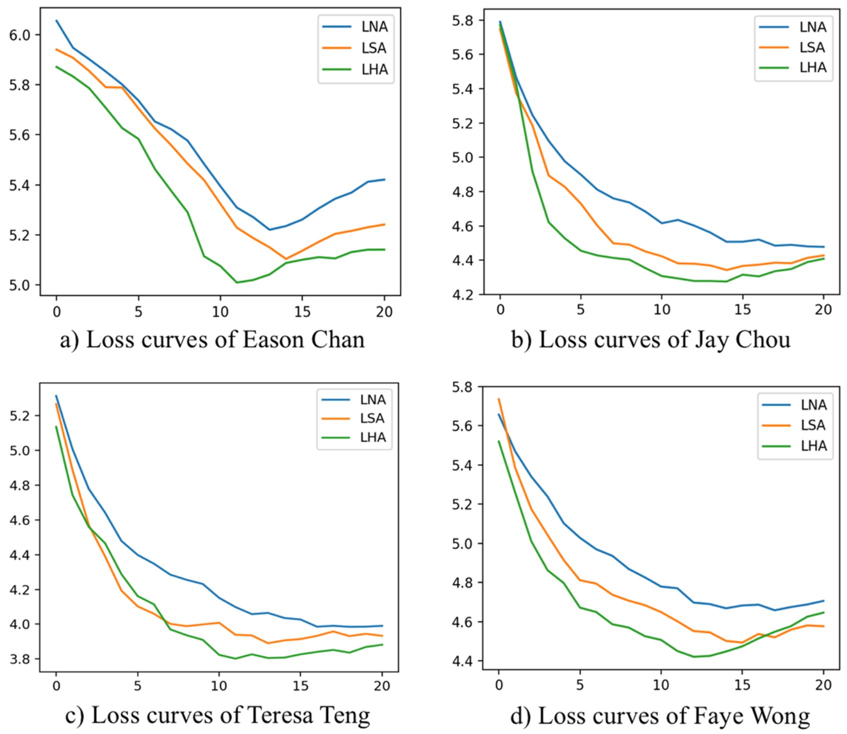
Lyric-Single-Attention (LSA) This model is similar to the **LNA** model, but a sentence-level attention is used to capture the contextual attention of the input sentence, which is described in Section 3.2.

Lyric-Hierarchical-Attention (LHA) The proposed hierarchical attention based model is able to process multiple lines of a lyric at a time, and generate the next line of the lyric. The sentence-level attention (Section 3.2) and the hierarchical lyric-level attention (Section 3.3) are both used to obtain sentence attention vectors of the lyric lines and a lyric attention vector of these lines.

4.3 Experimental settings

Each lyric document consists of multiple lyric lines, and each line contains multiple words. There are some difficulties in Chinese lyric preprocessing. For some Chinese artists, other languages like English, Korean and Japanese are also included in their lyrics, which may influence the performance of the model. By using regex expression, most of the words from other languages than Chinese have been cleaned. If a lyric contains more than half of the words from other languages, it would be removed from the entire dataset. There are also some words written in traditional Chinese, which should be transformed into simplified Chinese, to reduce the parameters of the vocabulary. Furthermore, Chinese characters are connected together, which needs to be segmented. All the Chinese lyric lines are segmented using $jieba$ segmenter. All the datasets are randomly shuffled and 10% of each dataset are been used for testing.

Various values of embedding dimension (50, 100, 150, 200), hidden units (50, 100, 150, 200) and number of layers (1, 2, 4) of LSTM networks are tested to find the best parameters for the models. The cross-validations over all datasets show that the larger numbers of embedding dimension and hidden units are used, the smaller validation losses could be obtained. Therefore, both of these two parameters are set to 200. And the number of layers of stacked LSTM networks is set to 2.



a) Loss curves of Eason Chan

b) Loss curves of Jay Chou

c) Loss curves of Teresa Teng

d) Loss curves of Faye Wong

Fig. 2 Loss curves of three models on four test datasets

No pre-trained word vector is used in each model. To learn word vectors which are related to different datasets from scratch, an embedding layer is used in the sentence encoder and decoder, to map the index of each word into a fixed size dense vector. The weight of the embedding layer is randomly initialized and then trained with the other part of the model during the training phase. Therefore, the word vectors can be obtained after the training on each dataset.

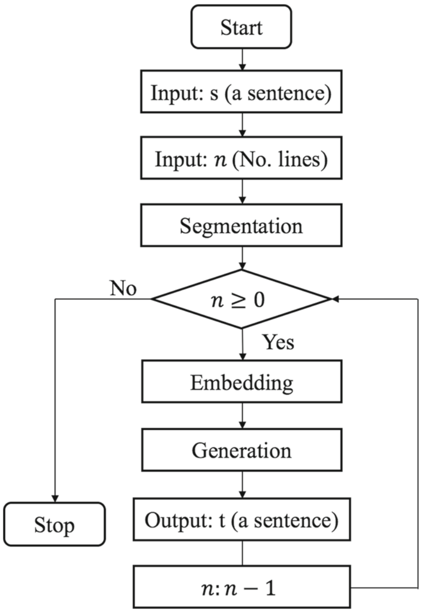


Fig. 3 The workflow of lyric generation process for the LHA model

4.4 Evaluation metrics

To offer objective and quantitative evaluation of the proposed model, two evaluation metrics are adopted to evaluate the performance of the models on all dataset, which are similar to Kawthekar et al. [16]. These metrics are described below:

Loss The cross-entropy loss is calculated for both training and test sets. The test losses of different epochs of training are recorded to draw the loss curves of different models.

Table 2 Text perplexities of different models on all datasets

Dataset	Model	Perplexity
Eason Chan	**LNA**	185
	LSA	164
	LHA	**150**
Jay Chou	**LNA**	88
	LSA	76
	LHA	**72**
Teresa Teng	**LNA**	54
	LSA	49
	LHA	**45**
Faye Wong	**LNA**	105
	LSA	89
	LHA	**83**

The bold emphasis means proposed methods outperform other baseline methods

Table 3 The generated ten lines for Eason Chan

Chinese	English
转身 锁上 从前，阳光 非常 刺眼	Turn and lock on the past, the sun is dazzling.
忙着 做戏，好戏 都 上演	Busy in pretending, good plays are all on.
回忆 里 的 蜘蛛 网，爱情 在 挣脱	Love is breaking the cobwebs in the memory.
烛光 上 的 时间，不再 有 你 的 离开	Time on the candlelight, no longer has your leave.
擦身 而过 的 所有 画面	All the pictures passing by.
我 却 不能 停 在 原点	But I can't stop at the original place.
流落 的 时间 好多，无人 前来 认领	So much time left, no one comes to claim.
谁怕 暴露 了，怎么 有 那么 无奈	Who is afraid of exposure, how so helpless?
后悔 不 珍惜，为何 总要 变 尘埃	Regret not cherish, why always turning to dust?
我 庆幸 我 谁 都 没 明白	I am glad I don't understand anybody.

The calculation of cross entropy for each sample is shown in (18).

$$H = -\frac{1}{T}\sum_{t}^{T}\sum_{i}^{C} y_{ti} ln \hat{y_{ti}} \tag{18}$$

Here, T is the length of the output sequence, C is the vocabulary size, y and $\hat{y}$ represent the true and predicted value of output sequence respectively.

Perplexity Perplexity is a measurement of how well a model predicts a sample. A low perplexity indicates the probability distribution is good at predicting the sample. It is a common evaluation metric for text generation tasks. The perplexity of the generated lyrics is calculated as e^H, where H is the cross-entropy loss.

4.5 Experimental results

After running for 20 epochs on each dataset, the losses and text perplexity for each model are recorded for detail comparison.

4.5.1 Loss curves

The loss curves of three different models on four different datasets are shown in Fig. 2. Results on all datasets show that the attention based methods gains smaller losses than the baseline model without attention, which means better performance. The proposed LHA model has the best learning loss on all datasets. By capturing the context attention of both sentences and the entire document, the proposed hierarchical context attention based model gains the best results. That is to say, the proposed method is significant in capturing semantic context information of lyrics and generating proper lines.

Meanwhile, the LHA model converges at about 10 to 15 epochs, which is slightly faster than the baseline models. The slight increases of test losses show signs of over fitting on further training.

4.5.2 Perplexity

The perplexities of the three models on all datasets are shown in Table 2. The proposed LHA model gains the lowest perplexities on all datasets, which means best performance comparing to the baseline models.

A lower text perplexity means that the model can infer the generated words more correctly. Models with high perplexity seem to generate lyric lines by randomly selecting words from the vocabulary list, which lacks of patterns and styles of the artists. Among all these datasets,

Table 4 The generated ten lines for Jay Chou

Chinese	English
我的 世界，你 比作 一只 小狗	You compare my world to a puppy.
天真 着，你的 愿望 就像 幅 画	Innocently, your wish is like a painting.
你的 脸上 没有 难过	Sorrow never occurs on your face.
这 感觉，我 写在 我的 音乐 里	This feeling, I write it in my music.
你的 夜曲，是 最美 的 风景	Your nocturne is the most beautiful scenery.
有 多少 幸福 的 味道	So many taste of happiness.
让我 爱你，爱 在一起	Let me love you, love together.
把 手心 连在 一起	Connect the palm of our hands together.
我会 回到 过去，留住 甜蜜	I will return to the past, and keep the sweet.
和 在你 圆周 的 情绪	And the emotions in your circle.

Table 5 The generated ten lines for Teresa Teng

Chinese	English
那夜 满天 星光 闪闪	That night starlight filled the sky.
我 痴痴 地 在你 身旁	I stayed dazedly beside you.
怕你 可能 匆匆 离去	Afraid you might leave in a hurry.
我 表达 真的 心意	I expressed my true kindly feeling.
不管 你 身在 何方	No matter wherever you are.
为什么 阵阵 秋雨 美丽	Why is the autumn rain so beautiful?
而 我 在 秋风 边 辗转	But I loiter around the wind.
多少 相偎，可恨 又 别离	So much time together, but then you leave again.
我 告诉 我 不要 叹息	I tell myself not to sigh.
把 爱 凝聚 在 四季	To cohere the love in the four seasons.

the perplexities of Eason Chan are much higher than other artists. One main reason is that it has larger numbers of lyric lines than others, which results in poor performance of all models. Another key reason is that most of the lyrics of Eason Chan are written by different artists, the pattern and style of him are not clear enough to learn. For other artists like Teresa Teng, her songs are always talking about love and friendship, which are much easier to learn. Therefore, the reason why Teresa Teng gains the best results is easy to be explained.

5 Lyric generation line by line

The workflow of the lyric generation process of the LHA model is shown in Fig. 3. To generate lyric lines with human intention, a starting Chinese sentence (s) and the number of lines which need to be generated (n) are inputted into the generative model. The input sentence is firstly segmented into word sequences and then embedded as word vectors. The generation process takes in the sequential word vectors and generates a lyric line word by word until it meets the end token. The generation of the next line is based on the input and contextual attention of the former line, as well as the contextual attention of the entire lyric document. After generating one line, the n should be set to $n-1$, if n equals zero, the generation process will stop, the generated n lines of lyrics are treated as the final result.

After training for proper epochs for each artist, the generative model is able to generate some lines of lyrics with a good manner.Tables 3, 4, 5 and 6 show the generated ten lines of lyric for each singer. For example, the generative model learnt from lyrics of Eason Chan is able to generate some lyrics as shown in Table 3, which seem to be mature and sad about love and always look back for most of time. For Jay Chou, the generated lyric lines seem to be young, sweet and mysterious, which is just like the style of the singer. For the famous singer Teresa Teng, sweet love and missing someone are the major theme of the generated lyrics. And for Faye Wong, the generated lyric lines express her loneliness and pursuit of love, which seems that she got hurt a lot in most of the relationships. One important thing is that all the generated lyric lines of each singer share similar styles and patterns, which shows the great effectiveness of the hierarchical context attention model.

Table 6 The generated ten lines for Faye Wong

Chinese	English
盲目 欢喜 与 憎恨，定 让 我 疯狂	Blind joy and hate, it makes me crazy.
是 哪样 意义，是否 纯情	To what kind of meaning, is it innocent?
漆黑 中 无力 清醒	Unable to sorb up in the dark.
让 我 害怕 总是 爱	Love always makes me afraid.
但愿 能 共 你 编排	Wish I could arrange it with you.
纵然 那 篇章 美好	Even if the chapter is beautiful.
我 只 接受 一段 旧情	I only accept an old affection.
谁 明白 我 爱过，有 你的 温暖	To cohere the love in the four seasons.
爱情 中 谁 也是 孤独	To cohere the love in the four seasons.
我 却 受控 在 一起	To cohere the love in the four seasons.

6 Conclusion

To meet the challenge of being meaningful and semantically related to a certain scenario, a Chinese lyric generation system is constructed to learn the patterns and styles of certain lyricists and generate lyrics automatically. A Long short-tern memory network encoder-decoder network is utilized to encode each lyric line into higher representation and generate the next line word by word. The LSTM network shows a great power in processing sequential text. A hierarchical context attention model is designed to capture the contextual information at both sentence and document level to learn high level contextual representations of every lyric lines and the entire lyric document. Experimental results show that the proposed hierarchical context attention based model gains the best test losses and text perplexities than other baseline models. The proposed method shows great effectiveness in the lyric generation process, which gains a deep understanding of the lyric document and generates the target lyric line with full context support.

Although the proposed lyric generation system has achieved certain results, it is still insufficient in several aspects such as diversity and stability. There is still a long way to go to meet the standard of professional artists. In recent years, deep reinforcement learning as well as generative adversarial networks has shown great effects in natural language processing, which could be further studied for automatic Chinese lyric generation processes.

Acknowledgements This paper is supported by the project 61303094 supported by National Natural Science Foundation of China, by the Science and Technology Commission of Shanghai Municipality 16511102400, by Innovation Program of Shanghai Municipal Education Commission (14YZ024).

References

1. Barbieri G, Pachet F, Roy P et al (2012) Markov constraints for generating lyrics with style[C], pp 115–120
2. Sridhar R, Ganga K, Prabha G D (2014) Automatic Tamil lyric generation based on ontological interpretation for semantics[J]. Sadhana 39(1):97–121
3. Addanki K, Wu D (2013) Unsupervised rhyme scheme identification in hip hop lyrics using hidden Markov models[C]. In: International conference on statistical language and speech processing, pp 39–50
4. Watanabe K, Matsubayashi Y, Inui K et al (2014) Modeling structural topic transitions for automatic lyrics generation[C]. In: Proceedings of the 28th Pacific Asia conference on language, information and computing, pp. 422–431
5. Watanabe K, Matsubayashi Y, Inui K et al (2017) Lyrisys: an interactive support system for writing lyrics based on topic transition[C]. In: Proceedings of the 22nd international conference on intelligent user interfaces, pp 559–563
6. LeCun Y, Bengio Y, Hinton G (2015) Deep learning[J]. Nature 521(7553):436–444
7. Mikolov T, Karafiát M, Burget L et al (2010) Recurrent neural network based language model[C]. Interspeech 2:3
8. Sutskever I, Martens J, Hinton GE (2011) Generating text with recurrent neural networks[C], pp 1017–1024
9. Potash P, Romanov A, Rumshisky A (2015) Ghostwriter: using an LSTM for automatic RAP lyric generation[C], pp 1919–1924
10. Sundermeyer M, Schlüter R, Ney H (2012) LSTM Neural networks for language modeling[C]. In: Thirteenth annual conference of the international speech communication association
11. Zhang X, Lapata M (2014) Chinese poetry generation with recurrent neural networks[C]. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 670–680
12. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks[C]. In: Advances in neural information processing systems, pp. 3104–3112
13. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate[J], arXiv:1409.0473
14. Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation[J], arXiv:1508.04025
15. Hu Z, Yang Z, Liang X et al (2017) Toward controlled generation of text[C]. In: International conference on machine learning, pp 1587–1596
16. Kawthekar P, Rewari R (2016) Bhooshan s evaluating generative models for text Generation[J]

Xing Wu received his PhD in Computer Science at the Shanghai Jiao Tong University. He is an associate professor at the School of Computer Engineering and Science, Shanghai University. His research interests include issues related to artificial intelligence, data mining. He is the author of a great deal of research studies published at national and international journals, conference proceedings as well as book chapters.

Zhikang Du received his master degree in School of Computer Engineering and Science, Shanghai University. His current research interests include computer vision, artificial intelligence, deep learning and natural language processing. He has published some research studies on international journals and conferences.

Yike Guo is a professor of Computing Science in the Department of Computing at Imperial College London. He is the founding Director of the Data Science Institute at Imperial College. Professor Guo also holds the position of dean of the School of Computer Engineering and Science, Shanghai University. Professor Guo has published over 200 articles, papers and reports.

Hamido Fujita is the professor and director of Intelligent Software Systems Laboratory at Iwate Prefectural University. He is researching on designing new software techniques based on cognitive interaction between human and machines, and intelligent decision making based on Medical Knowledge for medical decision making application.