Chinese Lyrics Generation Using Long Short-Term Memory Neural Network

Xing Wu[1,2](✉), Zhikang Du[1], Mingyu Zhong[1], Shuji Dai[1], and Yazhou Liu[2]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
{xingwu,duzhikang,zhongmingyu,daishuji}@shu.edu.cn

[2] Key Laboratory of Image and Video Understanding for Social Safety, Nanjing University of Science and Technology, Nanjing 210094, China
yazhouliu@njust.edu.cn

Abstract. Lyrics take a great role to express users' feelings. Every user has its own patterns and styles of songs. This paper proposes a method to capture the patterns and styles of users and generates lyrics automatically, using Long Short-Term Memory network combined with language model. The Long Short-Term memory network can capture long-term context information into the memory, this paper trains the context representation of each line of lyrics as a sentence vector. And with the recurrent neural network-based language model, lyrics can be generated automatically. Compared to the previous systems based on word frequency, melodies and templates which are hard to be built, the model in this paper is much easier and fully unsupervised. With this model, some patterns and styles can be seen in the generated lyrics of every single user.

Keywords: Lyric generation · Long Short-Term memory · Language model · Sentence vector

1 Introduction

Writing songs is a good way for users to express their personal emotions, lives, hopes, and attitudes towards things. Lyrics take a great role to do that job, as well as rhythms. All the writers have their own patterns and styles, and their own ways to show their love, dreams and so on. Writing lyrics is not an easy task for human and it usually comes with rhythms. Automatic lyric generation tasks always start from defining keywords, choosing a template, matching the rhythm and generating words using ontologies with these kinds of constraints, it's hard for system construction and maintenance, and to capture the patterns and styles of a single user.

The work of this paper tries to learn the patterns and styles of every single user automatically, using state-of-art machine learning algorithms, and generate lyrics of specified singers automatically, using recurrent neural network-based language modeling. With Long Short-Term Memory network (LSTM), context information of long texts can be captured into the memory (the parameters of the network). The context information gathered from the previous texts is useful for the model to generate new texts with context support. The model takes all the lyrics of a person as input, to capture

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part II, LNAI 10351, pp. 419–427, 2017.
DOI: 10.1007/978-3-319-60045-1_43

its statistical patterns and generates new lyrics just like its own style. Contrary to the previous work, this model doesn't need to rely on other kinds of techniques and resources like ontologies, rhymes, templates, word frequencies and so on, thus it's easy to implement and train, and gain better result based on the formal evaluation methods.

The remaining paper is separated into four parts. Section 2 shows some related work on poem and lyric generation, which are mainly based on ontologies, templates and word frequencies. Some work uses machine translation approach to generate lyrics line by line. Section 3 shows the details of the model of this paper, training the context representations of words and sentences, and generating lyrics word by word using Long Short-Term Memory neural network-based language model. Section 4 shows some experimental results on lyrics of three Chinese singers, which seems good to capture some styles of them. Section 5 draws the conclusion on this model, to show that it is easy to be built and trained, with the help of word embedding trained on Chinese Wikipedia data, the model can generate many other words, which makes it more flexible.

2 Related Work

The generation of lyrics are much more like the generation of poems, which has been popular for many years. Most of the work is based on word frequency, melodies or templates, which seems to have gained many difficulties. The report of Manurung et al. [1] shows that most of the difficulties comes from the difficulty of natural language generating system, the problems of architectural rigidness, lack of resources supplied to satisfy the multitude of syntactic and semantic constraints, and the objective evaluation of the output text.

To deal with all these kinds of difficulties, many researchers proposed their own methods in different aspects. Diaz-Agudo et al. [2] uses case-based reasoning ontology to design a knowledge intensive system to capture knowledge in cases of texts provided by user, and tries to gather the knowledge to form a regular line of texts using ontologies indexing, this system has a highly dependency on the quality of the ontology knowledge base, which is hard for construction and maintains. Manurung [3] then utilities an evolutionary algorithm approach to generate poems, which uses a linguistic representation based on Lexicalized Tree Adjoining Grammar, the structure of the tree adjoining grammar is complicated and may even be bad when the text becomes oral. Oliveira et al. [4] uses a kind of system to generate lyrics automatically for given melodies, they present two strategies to generate words for this system, random words and generative grammar, tests show that the later one gains better result, it may be a good idea but it takes the melodies into consideration. Tosa et al. [5] proposes a method to combine user queries with rules extracted from a corpus and additional lexical resources to generate new poems, the quality of rules definition of the corpus shows a highly influence on the poem generation. Colton et al. [6] describes a full-face corpus-based poetry generation system which uses templates to construct poems according to given constraints on rhyme, meter, stress, sentiment, word frequency and word similarity. The full-face poetry generator takes advantages of many previous techniques, thus it's complicated and hard to be implemented.

There are also some researches use summarization methods to do the text-generation job inspired by the statistical machine techniques. Genzel et al. [7] implements the ability to produce translations with meter and rhyme for phrase-based MT to gather the poetic constraints. He J et al. [8] uses a phrase-based SMT approach which translates the first line into the second, to generate Chinese poems.

The above methods take a lot about knowledge and rules extracted by human into consideration, which may take a great deal of time for the preparation work to construct their systems. Compared with all these methods, our method is a totally automatic and unsupervised one. With the help of recurrent neural network model like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), the system can automatically capture the semantic meanings and grammatic structures of lyrics from users, then generate new lyrics word by word using recurrent neural network-based language model.

3 Lyric Generation

The work of generating lyrics of specified users can be separated of 3 parts. Section 3.1 shows a way to learn the vector representations of words using word2vec model, to capture some semantic meanings of words and word similarities. Section 3.2 tries to learn the vector representations of sentences using long short-term memory neural networks. The sentence vectors learned from the recurrent neural network model significantly captures the semantic meanings of sentences which are useful to introduce the context information into the generative model. Section 3.3 tells the detail about the recurrent neural network-based language model to generate lyrics from the vocabulary list of word2vec model given the lyrics data of users. The model uses a LSTM neural network to train the higher representation of the sequential input and maps the representation to a single word (phrase level) index in the vocabulary list, using a softmax classifier. Section 3.4 and 3.5 shows the details of how to train the LSTM context model and LSTM generative model, and how to generate new words with the combination of these two models.

3.1 Learning Context of Words

In order to capture some semantic meanings of words and word similarities, we use word embedding to represent words. The concept of word embedding was first introduced in neural probabilistic language model [9] to learn a distributed representation for words. In 2013 a fast training method of word embedding called word2vec was proposed by Mikolov et al. [10] The word embedding of a word is a dense vector, the paper shows that words with similar context seems to be closer in vector space.

Another advantage of using word embedding is that it is a dense vector of a fixed size, always with a dimension of 128, 256 and so on. The traditional one-hot representation of words is a very large vector of vocabulary size, where there is only one 1 in the vector and others are all 0. It loses a lot of semantic meanings of words like word order and word similarities, and the large size of vector may even lead to the curse of dimensions, as it is hard to perform the matrix calculation in the neural networks. Word embedding is a great way to fix all these problems and gain good result among many applications.

For English, words are separated by space, and each word always represent one meaning. But for Chinese, words are connected together, and the meaning of a sentence is always represented by two or more words, one-word level word embedding is meaningless than phrases. In order to train Chinese word embedding, we first segment all the data into two or more words-level phrases, and feed them to the word2vec model, so each word embedding represents a phrase, other than only one character. The result word vectors trained from Chinese Wikipedia data shows that similar words are closer, as shown in Table 1.

Table 1. Five most similar words of four words

Words	**Five most similar words**				
幸福 Happy	快乐 Happy	美好 wonderful	甜蜜 sweet	喜乐 happy	欢乐 joyful
流行 Popular	风行 Popular	盛行 popular	普及 popular	风靡 popular	受欢迎 popular
美丽 Beautiful	迷人 Fancy	可爱 cute	漂亮 beautiful	最美 most beautiful	优美 beautiful
句子 Sentence	短语 Phrase	语句 sentence	词语 word	词组 phrase	问句 question

3.2 Learning Context of Sentences

Recurrent neural network is a kind of artificial neural network where connections between units form a directed cycle. Unlike feedforward neural networks, recurrent neural networks can use their internal memory to process sequence of inputs, and store all the internal states in the memory as useful information. With the help of the internal memory, researchers even find out that it can capture some context information of words like word orders, word meanings and even some grammatic structure, thus it can be a good solution for our model to capture the style of users.

With the great improvements of computation performance, some deep learning recurrent neural networks like Long Short-Term Memory (LSTM) and Gated Recurrent Unit(GRU) become popular among these years. Long Short-Term memory was first published by Hochreiter and Schmidhuber [11] in 1997. Traditional RNNs that suffer from the vanishing gradient problem, when the sequential data is too long, the context information will be lost, it's not very suitable to deal with long data. LSTM is augmented by recurrent gates called forget gates to prevent backpropagated errors from vanishing or exploding, which makes it suitable to deal with very long texts and time steps, and can still capture the context information into the memory. For now LSTM is a very popular deep learning neural network in the field of natural language processing, like machine translation and language modeling. Gated Recurrent Unit [12] is another kind of recurrent neural network which simplify the process of LSTM network.

This paper tries to train a LSTM-RNN context model to capture the context information of every line of lyrics from a user. Every line of lyrics is segmented into phrases list, and then fed into the model word by word (phrase level), and it will generate a context vector representation of this line. The LSTM context model is shown in Fig. 1. As we can see, this model takes a sequence of word vectors as input and output a sequence of vector representations, we take the last output vector as the context of the input line of lyrics, and then feed it into a fully connected neural network layer (the number of the units in this layer is the number of lines in the lyrics data), to project the context vector to the vector space in the line index level. A softmax classifier is then been introduced to predict which line of index it is given the input sample. The softmax classifier layer can be seen as an activation layer to predict the most probable index. The probability of each unit can be calculated using softmax function as it is shown in (1).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{1}$$

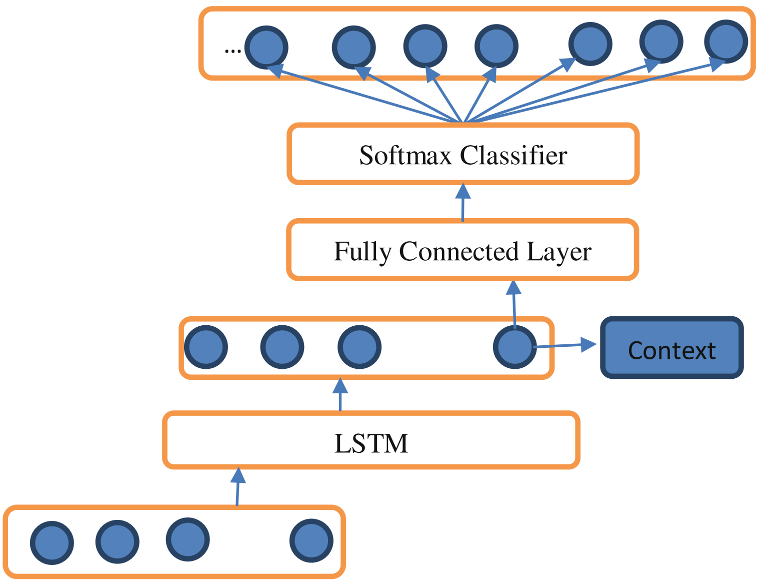

Fig. 1. LSTM context model

3.3 Generating Lyrics Word by Word

In 2010 Mikolov et al. [13] proposed a method to deal with language modeling using vanilla recurrent neural networks. This RNN-based model uses sequence of previous words as input and predict the next word using softmax classifier, which outperforms the standard backoff n-gram models and traditional feedforward neural networks. We take the inspiration of RNN-based language model to generate lyrics word by word, and automatically segmented them line by line using some predefined marks. The generative model is shown in Fig. 2.

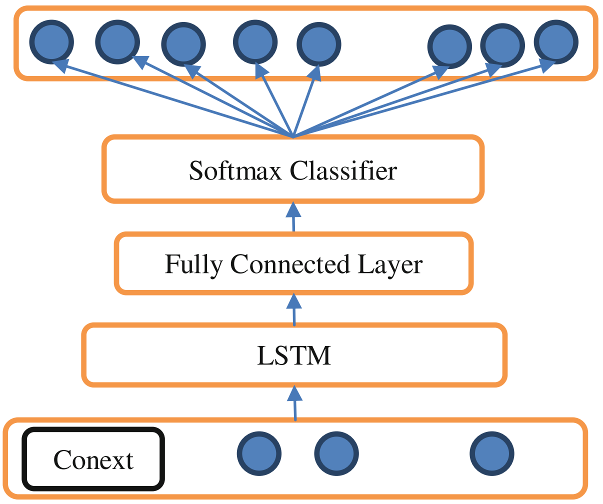

Fig. 2. LSTM generative model

Instead of using traditional recurrent neural network, a LSTM neural network model is used to train the lyrics data. As is represented in Sect. 3.2, the context of each line of lyrics is represented by a context vector, which is called a sentence vector. To import the context information of previous lines, we combine its sentence context with the sequential word vectors together and feed them to the LSTM neural network. A softmax classifier is right after the LSTM network to predict which word to generate at next. The softmax classifier is right like which is in the LSTM context model, but the number of units is the vocabulary size of the word2vec model, in order to pick up the rightful word to be generated.

3.4 Training

The training processes can be separated into two parts, to train the sentence context model and to train the generative model. All the training processes can be treated as unsupervised training without any manual labels.

In order to train the LSTM context model, we separate the training lyrics line by line, each line represents one index. Thus, the input is a sentence fed into the model word by word, and the target output is a class represented as a single line. After the training is done, the context of vector of each line can be saved for the use of LSTM generative model.

As for the training of LSTM generative, not only the word vectors of the current line are fed into the LSTM network, the context vector of the previous line is also imported to capture the context information of previous lyrics. To train the language model, we also need to separate the training lyrics into the input data and the target data. Each line of lyrics is concatenated with a start token and an end token, to specify the start and the end of a line, and then cut into a fixed length of sequential list, the target data is the word right next to the word sequence.

The LSTM context model and the LSTM generative model look quite similar as is shown in Figs. 1 and 2, but there are some main differences. Firstly, the input line of lyrics doesn't need to be cut into fixed length in the LSTM context model, but for the LSTM generative model, the input data should be cut into fixed length to satisfy the need of the language model. Secondly, the LSTM context model is used to train the sentence context vector of every single line of lyrics, which is then fed into the LSTM generative model to capture the context information of the previous line of lyrics. Finally, the target output of the LSTM context model is the index of a line of lyrics, but for the LSTM generative model, it is a word index in the vocabulary, to predict the next word of the current line.

3.5 Generating

The process of generating lyrics is a feedforward neural network which inputs a sequential data and outputs a word index in the vocabulary list. After the training process is done, the model randomly picks some start words (also called as seed words) and looks up to their word vectors to form a sequential input, combined with the previous sentence context. At first time the sentence context is a zero vector since there is no previous word. But later, the sentence vector can be calculated using the LSTM context model, then it can be used to generate the word for now. The LSTM generate model takes the sequential input and outputs a densely-connected vector with the dimension of vocabulary size of the word2vec model, a softmax function is then given to calculate the probability of each word to be generated. The generative model can go again and again to generate as many words as possible. Remember that we introduce the start token, the end token and the unknown token in Sect. 3.4, we can simply separate the generated words line by line using these tokens.

4 Experimental Results

We experiment our models on three famous Chinese singers, Jay Chou, Eason Chan and Faye Wong. The system first segments the lyrics data from each singer line by line, and extracts the vocabularies, then represents each line of lyrics as a sequence of word vectors. As it is shown in Sect. 3.1, the word vectors have been pre-trained on Chinese Wikipedia data using word2vec model, which are quite helpful to capture the word meaning of each sentence. And for the words not shown in the vocabularies of the word2vec model, they are represented as an unknown token. The sequences of word vectors representation of sentences are then fed into the LSTM context model line by line. After training the sentences for many epochs, all the sentence vectors can be projected into a latent vector space, where similar lyric lines are closer to each other.

As the context vector of the previous line has been introduced into the LSTM generative model just as the vector representation of each word, it is fairly easy to train the language model and use softmax classifier to predict the most possible word to be generated. After training for many epochs, the generative model is able to generate some lines of lyrics with a good manner. For example, the generative model learned from 243 songs from Jay Chou, and generated some lyrics in Table 1, which seems to be funny and mysterious, just like the style of himself. As for Eason Chan, 325 songs have been fed into the model to learn his styles, the generated lyrics is shown in Table 2, which seems more mature and sad. And for Faye Wong, 215 songs have been trained to learn her pattern, the lyrics shown in Table 3 express her style of songs, which is nature and free of love, sometimes mysterious (Table 4).

Table 2. Generated lyrics of Jay Chou

童话 猫 跟你 静静 在 教室 想 了解 初恋 Fairy cat in classroom with you quietly, wants to know first love
千里 沉默 送你 离开 爱着 大海 可怕 See you off silently for a long way, you love scary see
青花 香残 满地 一种 神秘 空气 Green floral residue all over the floor of a mysterious air

Table 3. Generated lyrics of Eason Chan

沉默 难舍 像我 不能 入睡 Silence and hard to drop like me, can't sleep
苦痛 挽回 太难 换取 一只 手表 Recovery from suffering is hard, to exchange a watch
残酷喜剧实在嘲讽路人可惜岁月太沉重 The comedy is a crucial sarcasm to stranger, it's a pity that life is heavy

Table 4. Generated lyrics of Faye Wong

思念 梦中人 爱 得 单纯 天真 Missing the man in the dream, love is pure and naive
生命 无罪 但 偏要 怨 半世 年华 Life is innocent, but one insist on complaining for half of life
故事 结尾 如风 偏离 轨迹 The end of the story is like a wind off-track

5 Conclusion

The results of the generated lyrics of the above three singers show that the models can significantly captures some patterns and styles of the input users automatically. Although the patterns and styles of singers are latent as parameters in their own trained models, they can be shown by generating some lyrics with the generative model. With more times of training and validation, our models can perform even better. One remarkable thing is that the models try to train the previous line of context to improve the performance of the generative model. Introducing the previous context into the current generating process might be a significant way.

Long Short-Term Memory neural network plays a great role in the context model. With its help, sequential text data can be processed line by line to get the context representation of sentences. The results in this paper show it useful in generating lyrics given previous context. This may be a very good method for other applications like text classification, sentiment analysis and so on.

Besides, with the help of word vectors trained on Chinese Wikipedia data, the model can generate words not just from a given singer's lyrics, but some words with similar semantic meanings, so it can generate purely new songs, and it is more flexible.

Acknowledgement. This paper is supported by the Science and Technology Commission of Shanghai Municipality (16511102400), by Innovation Program of Shanghai Municipal Education Commission (14YZ024), and by the Jiangsu Key Laboratory of Image and Video Understanding for Social Safety (Nanjing University of Science and Technology), Grant No. 30920140122007.

References

1. Manurung, H., Ritchie, G., Thompson, H.: Towards a computational model of poetry generation. The University of Edinburgh (2000)
2. Díaz-Agudo, B., Gervás, P., González-Calero, Pedro A.: Poetry generation in COLIBRI. In: Craw, S., Preece, A. (eds.) ECCBR 2002. LNCS, vol. 2416, pp. 73–87. Springer, Heidelberg (2002). doi:10.1007/3-540-46119-1_7
3. Manurung, H.: An evolutionary algorithm approach to poetry generation (2004)
4. Oliveira, H.G., Cardoso, F.A., Pereira, F.C.: Exploring different strategies for the automatic generation of song lyrics with tra-la-lyrics. In: Proceedings of 13th Portuguese Conference on Artificial Intelligence, EPIA, pp. 57–68 (2007)
5. Tosa, N., Obara, H., Minoh, M.: Hitch haiku: an interactive supporting system for composing haiku poem. In: Stevens, S.M., Saldamarco, S.J. (eds.) ICEC 2008. LNCS, vol. 5309, pp. 209–216. Springer, Heidelberg (2008). doi:10.1007/978-3-540-89222-9_26
6. Colton, S., Goodwin, J., Veale, T.: Full face poetry generation. In: Proceedings of the Third International Conference on Computational Creativity, pp. 95–102 (2012)
7. Genzel, D., Uszkoreit, J., Och, F.: Poetic statistical machine translation: rhyme and meter. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 158–166. Association for Computational Linguistics (2010)
8. He, J., Zhou, M., Jiang, L.: Generating Chinese classical poems with statistical machine translation models. In: AAAI (2012)
9. Bengio, Y., Ducharme, R., Vincent, P., et al.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(Feb), 1137–1155 (2003)
10. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Chung, J., Gulcehre, C., Cho, K.H., et al.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
13. Mikolov, T., Karafiát, M., Burget, L., et al.: Recurrent neural network based language model. Interspeech **2**, 3 (2010)