Korean Song-lyrics Generation by Deep Learning

Sung-Hwan Son
Kookmin University
Dept. of Computer Science
Seoul, South Korea
zldejagkcm@naver.com

Hyun-Young Lee
Kookmin University
Dept. of Computer Science
Seoul, South Korea
le32146@gmail.com

Gyu-Hyeon Nam
Kookmin University
Dept. of Computer Science
Seoul, South Korea
ngh3053@gmail.com

Seung-Shik Kang
Kookmin University
Dept. of Computer Science
Seoul, South Korea
sskang@kookmin.ac.kr

ABSTRACT

In order to create a lyrics based on the characteristics of Korean, we reversed the K-pop lyrics data and use them as learning data. It transforms the incoming data to use certain elements of the sentence, such as predicates and conjunctions, as starting points of the string generation. The proposed song lyrics generation method considers the context between lyrics. Every time the model generates the lyric, the model goes through upper randomization based on a blank. It was confirmed that the lyrics generated using the reverse data, have a more natural context than the lyrics generated using the forward data. It is also possible to generate new lyrics similar to certain lyric structure.

CCS Concepts

• Computing methodologies → Artificial intelligence → Natural language processing→Natural language generation.

Keywords

Song-lyrics generation; deep learning; LSTM; language model; predicate.

1. INTRODUCTION

Machine learning is a process where a pool of test data is used to identify the pattern in the data. Nowadays, machine learning is improving and evolving at a rapid speed since the introduction of a deep learning mechanism. It is widely used, recognized, and studied as a useful tool in various fields, including Natural Language Processing (NLP) research.

In regards to processing natural language via artificial intelligence, natural language generation has long been a difficult problem. There were many attempts, starting with generating Chinese poetry[1,2,3,4], automated Image Captioning[5] where text is assigned to a given image, and Mask Generative Adversarial Networks[6] to fill in the blanks in a sentence.

In case of sequence data such as texts or videos, a variant of Recurrent Neural Network (RNN) is commonly used. In the early days, Basic RNN models could not be used broadly due to vanishing gradient problem that the information of data used at the beginning of learning forgotten as the learning continued. Since then, such issues were handled by the development of Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Despite active research on the topic, natural language generation still has a long way to go. Most language models ultimately distill down to learning and applying the patterns in the texts used in training. As a result, they are as good as humans at generating short, simple sentences. However, they find it difficult to maintain the context and flow when generating long, complex, or multiple sentences or paragraphs, because these require deeper understanding of the natural language.

Machine learning focuses on identifying a pattern from its training data to build a model that categorizes or generates data. In general, the purpose of an RNN is to generate new complete sequence data that is not included in the training data in accordance with the pattern of the training data. But assuming there were no overfitting issues, it is still difficult to generate long, complex, or context-bound sentences with just a basic recurrent neural network (RNN) model; this is Because creative writing with sufficient contents, considering the context requires a high level understanding of the language. Thus, for this paper, we focused on generating song lyrics, where the sentence structure and relationship between words are relatively simple.

Therefore, the implemented model is a neural network language (NNL) [7] model that generates Korean verses while maintaining the context using basic LSTM.

2. RELATED WORKS

There were previous studies on generating poetry or song lyrics[8,9,10,11]. They parsed existing lyrics for sentence structure, and replaced some verbs or nouns of the input seed. Addanki and Wu proposed a system to generate a predetermined length of rap bars given the lyric the length of the bar[12]. Malmi, Takala, Toivonen, Raiko, and Gionis extracted verses from its rap training data to put together lyrics that match the meter in a predetermined 16 verses[13]. Although these methods can produce song lyrics with nearly perfect bars, there is a

ICIIT '19, February 20–23, 2019, Da Nang, Viet Nam

ACM ISBN 978-1-4503-6633-5/19/02…$15.00
DOI: https://doi.org/10.1145/3321454.3321470

disadvantage that the results produced is depend on a too detailed framework.

Most of the generatied models learn and generate in the order of the language they generate. Unlike many other languages, Korean, however, usually locate predicate at the end of the sentence, and there are many variations of the predicate. Of course, the sentence can be generated in the forward direction without much problem in sentence unit, but a problem occurs in a paragraph or document which is much longer than the sentence unit.

In fact, if you create a sentence by giving the seed that correspond to the subject or object of the sentences, the model creates sentences that match the seed. Awkwardness, however, may still exist within the sentence; in many cases, some of the sentence generated with the same seed often does not fit the context. In case of Korean, which is an agglutinative language, there are many different ways of describing predicate of the same meaning. For example, a simple word like 'Hello!' have many different forms such as '안녕,' '안녕하세요,' and '안녕하시오' Thus, awkwardness may be formed due to the lack of unification; this is because predicates often play important roles of connecting sentences[14].

Therefore, this paper focuses on predicates and describes a model for generating Flexible song lyrics fitting to the context while maintaining the structure of the entered lyrics.

3. DEEP LEARNING MODEL

The key idea of the model is to create song lyrics that fit the Korean context. Therefore, we focused on generating a song lyric based on the predicate, which is an important contextual factor, according to the characteristics of Korean as an agglutinative language. So, each time a lyric bar was created, we tried to generate a bar from the predicate. Because the Korean language has a predicate in the latter part of the sentence, it reversed the sentence so that the predicate was in the forward of the sentence so that the model could learn. As a result, the predicate is fixed and the appropriate clauses are generated for the corresponding predicate, so that the context of the predicate between the nodes was expected to be naturally connected. Thus, In this section, we outline how we applied LSTM in our language model to generate context-aware lyric bars by reversing bar-length text.

3.1 Training Model

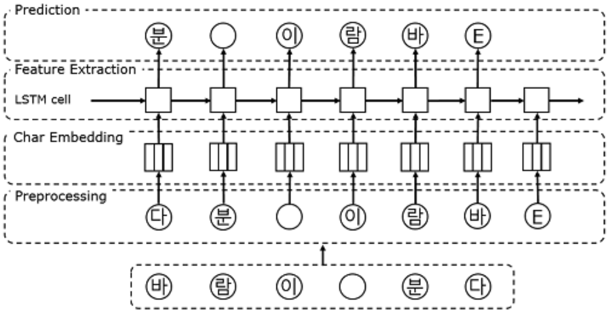

Figure 1. LSTM model, learning structure diagram.

The learning model is a language model using LSTM cell. The formula is as follows.:

$$\hat{P}(x_t|context) \approx P(x_t|LSTM(x_0, x_1, \dots, x_{t-1})) \quad (1)$$

x in the formula means one character, and the next character x_t is predicted as shown in the formula. Therefore, x_t is predicted based on the characters entered so far.

Figure 1 shows the structure of the training model. It is a character unit Korean language model with LSTM, predicting next character based on the input text. The training data we used to treat space and new line characters as separate characters. The empty circles in Figure 1 represent blank characters; and 'E' is end of lyric bar.

For example, ‘da(다)’ is predicted with a single space character, and ‘bun(분)’ is predicted using both ‘다’ and a space character. The space after ‘분’ is predicted with the initial space character, ‘다’, and ‘분’.

Training starts with the calculation of the state values of LSTM and total cost from the difference of the embedding values of predicted and correct character. The next step is plug in the calculated total cost to the partial differential equation, put the results into a vector, and apply a gradient clipping to adjust the maximum value of the gradients. After adjusting the gradients, we kick off learning sessions that update the weight to minimize the losses. State values are calculated through LSTM cells and updated as the learning continues.

3.2 Song-Lyrics Generation

One notable characteristic of our training data is that it consists of a bar-length lines, not sentences. A bar in a song is a text category bigger than a sentence. For example, ‘the wind blows’ is a sentence and a bar. However, ‘I thought this love was perpetual / but it floats away yet another day’ is a sentence made of two bars.

This means that during sampling, it is possible for the input seed to contain additional text corresponding to the end of the bar in addition to the predicates. So we tried several different ways on how to determine the seed.

Table 1. Lyric generation by seed syllables

Input Seeds	Results
One character	
야 일 에 듯 가	알 아주는 것 예 지금 흩어져**야** 어떨 땐 눈물만**일** 하늘의 별보다**에** 너무 어렸어**듯** 내 마음을 지켜보던 **가**
Two character	
말야 _일 곳에 앉듯 러가	어제도 좋았던 **말야** 그게 이렇지 **일** 추억이 보이긴 할 **곳에** 나를 다시 **앉듯** 마지막 창문이 흘**러가**
One to Two word	
_건_말야 _맞바꾸는_일 _곳곳에 _내려앉듯 _흘러가	아파 하는 **건 말야** 내 마음을 다 **맞바꾸는 일** 아무 말 없이 **곳곳에** 한숨 속에 **내려앉듯** 우릴 보며 눈물이 **흘러가**

As demonstrated in Table 1, seed are used as the last character, the last two characters, and the last word, or the last two words. In case of the seed is used as a word, if the last word of bar is a single character, we included the previous one of last word in seed. This is a way for our model to use the complete set of predicates or conjunctions possible to get enough information. Furthermore, if a word before the last word is a character, it should be combined with predicates such as '때 있지,' '건 말야,' and many others. So if the last word is two or three characters and the previous word is a single character, we also included previous word of the last word as an input. After the analysis of the sampling results, we concluded that selecting the one to two words from the bar as the seed yielded relatively better results.

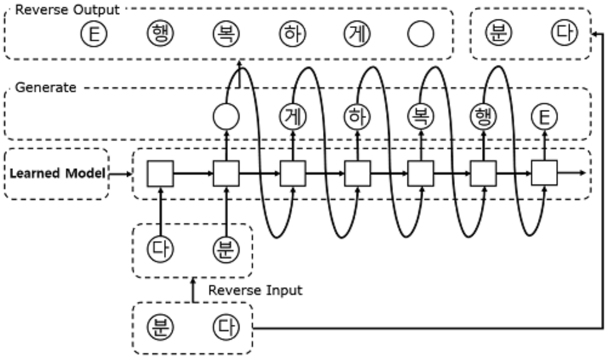

Figure 2. Model sampling structure diagram.

Figure 2 depicts the specification of the sampling that generates the texts with the LSTM language model. For example, if you want to create a lyric bar using a predicate ‘blowing’ as input: First, reverse the order of ‘blowing’ into ‘gniwolb’ to use as a seed. Then, the model generates texts using ‘gniwolb’. Each generated bar is separated by a newline character. Reverse the generated bar, connect ‘blowing’ at the end of the reversed text, and we complete to lyric bar. Most of the time, generated texts, such as ‘typhoon are blowing’ or ‘sweet love is blowing’, fit in the context of ‘blowing’, the predicate. Eventually, the song lyric is a set of bars that the model has created by receiving the seeds one by one.

4. EXPERIMENTAL RESULTS

4.1 Lyrics Dataset and Model Configuration

The training data used is a set of ballad lyrics. The song lyrics data were collected 10,000 songs by crawls web from a music site called 'MELON'. As shown in Table 2, there are approximately 334,000 bars and 4.2 million characters in total. For character unit learning and predictions, we reversed the texts in each bar. For example, ‘조금씩 잊혀져 간다’(it is gradually forgotten) is reversed and becomes ‘다간 져혀잊 씩금조’(nettogrof yllaudarg si ti).

Table 2. Composition of test data

Number of Bars	334,077
Number of Characters	4,272,113

The learning model used is the LSTM model. There were some trial and errors to find the best hyperparameter for LSTM model. At the beginning, we tried the values lined out in Table 3:

On our first attempt, we ran into overfitting, and the generated texts were almost identical to the training data. But, our goal was to generate new texts with patterns similar to the training data. Therefore, we concluded that the embedding size of one character was too large and the LSTM layer was deeper than necessary, resulting in excessive learning. The research was conducted by changing the hyperparameter accordingly.

Table 3. Hyper parameters for initial model

Char Embedding Size	700
Batch Size	60
Epoch	250
LSTM Layer	3
Length of Sequences	100
Dropout Rate	0.8

Table 4. Hyper parameters for adjusted model

Char Embedding Size	250
Batch Size	60
Epoch	150
LSTM Layer	2
Length of Sequences	25
Dropout Rate	0.8

Table 4 shows the best hyperparameter values we found after several attempts. Most modifications were on decreasing embedding space per character and depth of LSTM layers. Since our text generation unit has been the song’s bar, we also adjusted the sequence length so the model’s learning unit was as close to a single bar as possible.

4.2 Model Evaluation and Comparison

To verify whether the seed-based, generated text by reverse model is different from the original input, the BLEU Score was used. In addition, we performed a human evaluation on the generated song lyrics bar to check whether generated context in the reverse model is a natural flow like a normal human-spoken language. The evaluation data was generated 10 times for each of the five ballads with 10 bars using both forward model and reverse model with the same hyperparameter.

Table 5. Evaluation

	Forward Model	**Reverse Model**
BLEU Score	0.18310	**0.17922**
Human Evaluation	0.472	**0.774**

Table 5 shows the BLEU Score and human evaluation of both forward and reverse models. The BLEU Score is low at both model and the scores are similar. Therefore, it means that the two models created new song lyrics that are very different from the original lyrics.

The human evaluation is calculated by dividing the number of bars that are grammatically correct and natural in context by total number of generated bars. As a result, we could see that the reverse model produced a more grammatically correct and contextual song lyrics bar than the forward model.

In sum, it can be concluded that the reverse model creates song lyrics that are correct in context and different from the original lyrics.

The advantage of the reverse model is that the generated contexts are connected to each other as the predicates are used as input for seeds. To verify the advantage, we compared the reverse model with the forward model by using the same song lyrics as input to see if the reverse model had the expected effect that it should produce.

Table 6. Forward model and reverse model

<table>
<tr><th colspan="2">Original Lyrics</th></tr>
<tr><td colspan="2"><b>어쩌면</b> 산다는 <b>건 말야</b>
<b>지금을</b> 추억과 <b>맞바꾸는 일</b>
<b>온종일</b> 치운 집안 <b>곳곳에</b>
<b>어느새</b> 먼지가 또 <b>내려앉듯</b>
<b>하루치의</b> 시간은 <b>흘러가</b>
<b>뭐랄까</b> 그냥 그럴 <b>때 있지</b>
<b>정말</b> <b>아무것도</b>
<b>내 것</b> 같지 않다고 <b>느껴질 때</b></td></tr>
<tr><th>Forward Model</th><th>Reverse Model</th></tr>
<tr><td><b>어쩌면</b> 우리 사랑할 테니
<b>지금은</b> 감춰줄까
<b>온종일</b> 또
<b>어느새</b> 보냈어
<b>하루치의</b> 남자들이
<b>뭐랄까</b> 자꾸 열려나
<b>정말</b> 해주었던
<b>내 것</b> 같아 마냥 솔직했네</td><td>아파하는 <b>건 말야</b>
내 마음을 다 <b>맞바꾸는 일</b>
아무 말 없이 <b>곳곳에</b>
한숨 속에 <b>내려앉듯</b>
우릴 보며 눈물이 <b>흘러가</b>
멀어져 갈 <b>때 있지</b>
사랑이란 영화 속에 <b>아무것도</b>
너무 멀게 <b>느껴질 때</b></td></tr>
</table>

Table 6 is a comparison table between the forward model and the reverse model. The used original lyrics are a part of Hee-Eun Yang's "Always You." The commonality between the forward and reverse models is that they usually produce meaningful segments when viewed in lyrics bar units.

There is, however, a clear difference in the relationship between contexts. The reverse model receives the last word of the original lyrics bar, that is, the portion corresponding to the predicate. That is why song lyrics are created in a way that links the contexts of the bars in the reverse model. In the case of the forward model, the individual bars are generated properly, but in many cases, the contexts between the bars are not appropriate and the style of the predicate is not constant. Therefore, it is confirmed that the reverse model generates song lyrics at correct context and natural.

4.3 Final Model Samples

The ballad songs used for the experiments are parts of 'Around Thirty' by Kwang-seok Kim and 'You Know That' by Jin-ho Kim and 'Always You' by Hee-Eun Yang. The selection criterion is based on the lyrics of songs that are made in Korean and have little or no poetic license word or syntax.

Table 7 shows the generated text results with the seeds set to one to two word at the end of each bar. In general, the reversal model generates a bar corresponding to the input value and outputs a result similar to the original input lyrics structure. However, it was confirmed that the semantics of the output were different from the original input lyrics.

Table 7. Results

Original Lyrics	Generated Samples
또 하루 멀어져 **간다** 내뿜은 담배 **연기처럼** 작기만 한 내 **기억속에** 무얼 채워 살고 **있는지**	결국은 우리 멀어져 **간다** 우리 둘의 마지막 **연기처럼** 둘의 **기억속에** 모든게 흐르고 **있는지**
그렇게 지나간 시간 **속에서** 널 이렇게 **잊어가듯이** 자연스러워 지는 **모든것** 이렇게 우리 추억이 **되어** 서로에게서 **멀어져갔을 때**	사랑했던 그 시간 **속에서** 너를 **잊어가듯이** 기억 속에서 내 **모든것** 내게 추억이 **되어** 점점 **멀어져갔을 때**
어쩌면 산다는 **건 말야** 지금을 추억과 **맞바꾸는 일** 온종일 치운 집안 **곳곳에** 어느새 먼지가 또 **내려앉듯** 하루치의 시간은 **흘러가**	이제 이별이라는 **건 말야** 이 길을 **맞바꾸는 일** 이제는 **곳곳에** 그 비가 **내려앉듯** 눈물이 **흘러가**

5. CONCLUSION

The proposed method in this paper is to deep learning the Korean pop music lyrics with LSTM and to generate the text with sentence fluency in mind. For each original lyric, if the last word of the each bar was entered as seed, the lyrics's structure were similar but the lyrics of different contents was generated. We confirmed that the text generated via the proposed method flowed more smoothly than those generated by forward model. Because our automatic lyric generation model uses a character as the unit of learning, it generated grammatically incorrect word or sentences. We tried to improve our results by developing and applying a probabilistic language model that calculated the frequency of semantic relationship between sentences, but that did not get rid of the issue entirely. There are many reasons for this, but one of them is that it is difficult to clearly extract the predicate or conjunction and to input it as a seed.

Therefore, we will try to generate a better sentence by applying a language model that considers the context between sentences and a method of extracting important factor of the sentence. We will also try to generate a sentence through the Seq2Seq model, which takes as input important factors such as subject, object, and verb.

6. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning of South Korean government. (No.NRF-2017M3C4A7068186)

7. REFERENCES

[1] He, J., Zhou, M., and Jiang, L. 2012. Generating Chinese classical poems with statistical machine translation models. *AAAI*, 1650-1656.

[2] Zhang, X. and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. *EMNLP 2014*, 670-680.

[3] Yu, L., Zhang, W., Wang, J. and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. *AAAI*, 2852-2858.

[4] Yi, X., Li, R. and Sun, M. 2016. Generating chinese classical poems with RNN encoder decoder. a*rXiv preprint* arXiv:1604.01537.

[5] Karpathy, A. and FeiFei, L. 2015. Deep visual-semantic alignments for generating image descriptions. *CVPR,* 3128-3137.

[6] Fedus, W., Goodfellow, I., and Dai, A. 2018. MaskGAN: Better text generation via filling in the __. *arXiv preprint* arXiv:1801.07736.

[7] Bengio, Y., Ducharme, R., Vincent, P., and Jarvin, C. 2003. A neural probabilistic language model. *JMLR*, 1137-1155.

[8] Babieri, G., Pachet, F., Roy, P., and Degli Esposti, M. 2012. Markov constraints for generating lyrics with style. *ECAI*, 115-120.

[9] Oliveira, H. G., Diaz, H. A., and Gervas, P. 2014. Adapting a generic platform for poetry generation to produce Spanish poems. *In 5th International Conference on Computational Creativity, ICCC*.

[10] Potash, P., Romanov, A., and Rumshisky, A. 2015. GhostWriter: Using an LSTM for automatic rap lyric generation. *EMNLP 2015*, 1919-1924.

[11] Yan, R. 2016. i, poet: automatic poetry composition through recurrent neural networks with iterative polishing schema. *IJICAI*, 2238-2244.

[12] Addanki, K. and Wu, D. 2013. Unsupervised rhyme scheme identification in hip hop lyrics using hidden markov models. *Statistical Language and Speech Processing*, 39-50.

[13] Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. 2015. DopeLearning: A computational aproach to rap lyrics generation. *arXiv preprint* arXiv:1505.04771.

[14] Sohn, Ho-Min. 1999. The Korean language. *Cambridge University Press*.