한국어 문장 생성을 위한 Variational Recurrent Auto-Encoder 개선 및 활용
(Application of Improved Variational Recurrent Auto-Encoder for Korean Sentence Generation)

한상철 †
(Hanhong Hahn)

홍석진 ††
(Seokjin Hong)

최희열 †††
(Heeyoul Choi)

요약
딥러닝의 급속한 발전은 패턴인식 분야의 성장을 확신했으며 몇몇 문제에서는 인간 수준을 넘어가는 결과들을 보여주고 있다. 데이터를 분류하는 패턴인식과 달리 본 논문에서는 주어진 몇개의 한국어 문장으로부터 비슷한 문장을 생성하는 문제를 다룬다. 이를 위해 생성모델 중 하나인 Variational Auto-Encoder 기반의 모형을 한국어 생성에 맞게 개선하고 적용하는 방법들을 논의한다. 첫째, 교착어인 한국어의 특성상 헷갈리기 쉽고 단어 생성시 단어의 개수가 너무 많아 이를 줄이기 위해 조사 및 어미들을 분리할 필요가 있다. 둘째, 한국어는 어순이 비교적 자유롭고 주어 목적어 등이 생략되는 경우가 많아 일관성 떨어진 인코더를 양방향으로 확장한다. 마지막으로, 주어진 문장들을 기반으로 비슷하지만 새로운 문장을 생성하기 위해 기존 문장들의 인코딩된 벡터들을 이용하여 새로운 벡터를 찾아내고, 이 벡터를 디코딩하여 문장을 생성한다. 실험 결과를 통해 제안한 방법의 성능을 확인한다.

키워드: 딥러닝, 생성모델, 변분순환오토인코더, 한국어 문장 생성, 보간법

Abstract

Due to the revolutionary advances in deep learning, performance of pattern recognition has increased significantly in many applications like speech recognition and image recognition, and some systems outperform human-level intelligence in specific domains. Unlike pattern recognition, in this paper, we focus on generating Korean sentences based on a few Korean sentences. We apply variational recurrent auto-encoder (VRAE) and modify the model considering some characteristics of Korean sentences. To reduce the number of words in the model, we apply a word spacing model. Also, there are many Korean sentences which have the same meaning but different word order, even without subjects or objects: therefore we change the unidirectional encoder of VRAE into a bidirectional encoder. In addition, we apply an interpolation method on the encoded vectors from the given sentences, so that we can generate new sentences which are similar to the given sentences. In experiments, we confirm that our proposed method generates better sentences which are semantically more similar to the given sentences.

Keywords: deep learning, generative models, variational recurrent auto-encoder, Korean sentence generation, interpolation
1. Introduction

Recently, due to the revolutionary advances of deep learning, performance of pattern recognition has been dramatically improved especially in image recognition and speech recognition [1–3]. Actually, when the task is well-defined with enough data for training, deep learning has outperformed human level intelligence in such specific application problems.

In addition to pattern recognition, there is another deep learning research direction, which is to generate samples instead of recognizing samples [4]. After training, generative models generate samples from random noise or a context. There are two major approaches in generative models: generative adversarial networks (GANs) [5] and variational auto-encoders (VAEs) [6].

In this paper, we focus on generating Korean sentences which are similar to given samples. That is, given a few samples, our proposed system is to generate conceptually or semantically similar sentences. Since GANs cannot find similar sentences to a given sentence in the latent space, we use the VAE based approach in this paper.

To model sequences in neural networks, generally recurrent connections are added which leads to recurrent neural networks (RNNs), which is hard to train [7]. If VAEs are equipped with RNNs as the encoder and decoder, it makes variational recurrent auto-encoders (VRAEs) [8].

In addition to the fact that training sequences is harder than static input like images, we need to take the following points account when modeling Korean sentences. First, as an agglutinative language, segmentation of Korean words is not an easy task. If words are segmented by spaces, the number of words becomes too large to fit into any neural networks. Thus, we need a word segmentation method to reduce the number of words for neural networks.

Second, another characteristic of the Korean language is that the order of words in a sentence is not fixed grammatically. For example, an adverb can move around in a sentence without much restriction. It can imply that bidirectional modeling for a sentence might be better than unidirectional one, so that we extend the unidirectional encoder in VRAE to bidirectional one.

Third, to generate new sentences similar to given sentences, we interpolate the given sentences in the encoded space. Considering that a sentence is composed of discrete words (or symbols), it is not possible to interpolate the sentences as sequences of symbols. However, once they are encoded into a continuous vector space as in the skip-thought [9], a point in the middle of two vector points can represent a new sentence which is semantically in the middle of the two corresponding sentences.

In experiments, we show that the bidirectional encoder is better than the unidirectional one. Also, we present many examples of generated sentences to confirm that our proposed method works well.

The rest of this paper is organized as follows. In Section 2, we define our problem statements. Section 3 describes background including RNN and VRAE. In Section 4, our method is described in detail. Section 5 presents experiment results. Finally, Section 6 concludes our paper.

2. Problem Statements

Given $N$ sentences $s_1, s_2, \ldots s_N$, the model is to generate $M$ new sentences, $t_1, t_2, \ldots t_M$. To see how the model works, we first generate sentences from two sentences using the linear interpolation in the encoded space. Then, we interpolate $N$ sentences, expecting the generated sentences are located within the convex of the $N$ sentences in the encoded space.

3. Background

Since we are extending the unidirectional encoder to bidirectional one in VRAE, we briefly review RNN and VRAE in this section.

3.1 Recurrent Neural Networks

RNNs have recurrent connections from the output of certain time frame to the next time frame, keeping historical inputs as internal states. Such connections allow the network to memorize the previous word history. However, RNNs suffer from the vanishing gradient problem that limits the accessible range of inputs (or the number of previous words). While training RNNs, the gradient of error needs to be propagated back through the network but it easily
either decays or blows up exponentially. Long short-term memory (LSTM) was proposed as a solution for this problem [10]. In this paper, we use LSTM to encode and decode an arbitrarily long context in VRAE.

The ordinary RNN with \( m \) hidden nodes computes the sequence of hidden node output vectors \( \mathbf{h} = (h_1, ..., h_T), \mathbf{h} \in \mathbb{R}^m \) by iterating the following equations throughout the sequence

\[
\begin{align*}
  h_t &= \sigma (W_{xh}x_t + W_{hh}h_{t-1} + b_h), \\
  y_t &= W_{hy}h_t + b_y,
\end{align*}
\]

where \( W \) denotes the weight matrices, \( b \) denotes the bias vectors, and \( x \) and \( y \) stand for the network input and output vectors, respectively. Here, \( h \) memorizes the history of input sequence. RNNs with such memory structure have been successfully applied to language models as in speech recognition and machine translation [11–14].

LSTM neural network [10,15] is an RNN with memory blocks in the hidden layer, avoiding the exploding and vanishing gradient problems of RNNs [15]. As in Fig. 1, each LSTM memory block has multiple gates to store information over long time periods, mitigating the vanishing gradient problem. The forward propagation to obtain the sequence of hidden node outputs, \( \mathbf{h} = (h_1, ..., h_T), \mathbf{h} \in \mathbb{R}^m \) is given by

\[
\begin{align*}
  i_t &= \sigma (W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \\
  f_t &= \sigma (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \\
  c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\
  o_t &= \sigma (W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \\
  h_t &= o_t \tanh(c_t).
\end{align*}
\]

\( \sigma(.) \) stands for the element-wise sigmoid function, and the symbols \( i, f, o \) and \( c \), respectively, stand for the input gate, forget gate, output gate, and memory cell state vectors. \( W \) and \( b \) stand for the weight matrices and bias vectors as before. The LSTM parameters are trained with conventional backpropation through time (BPTT) algorithm using the stochastic gradient descent (SGD) method.

One can combine two separate LSTM networks that run in forward and reverse directions for implementing bidirectional architecture.

### 3.2 Variational Recurrent Auto-Encoders

Since variational recurrent auto-encoders (VRAEs) are extensions of variational auto-encoders (VAEs) to encode and decode sequences, we review VAEs, first. For the details, see [6] and [8] for VAEs and VRAEs, respectively.

VAEs are composed of an encoder and a decoder, and the objective function is log likelihood, which is hard to maximize. So, given an input sample \( x^i \), VAEs set a lower bound for the log likelihood as Eq. (8), and training is done by maximizing the lower bound.

\[
L(\theta, \phi, x^i) = -KL(q_\phi(z|x^i)p_\theta(z)) + \frac{1}{l} \sum_i \log p_\theta(x^i|z^i)),
\]

where \( q_\phi(z|x^i) \) and \( p_\theta(x^i|z^i) \) are the encoder and the decoder networks, and \( p_\theta(z) \) is the prior distribution which is usually a centered isotropic multivariate Gaussian distribution. Here, \( z \) is the encoded vector or latent state vector.

However, in the networks there is a sampling process after the encoder, which prevents error information from back-propagating through the network. To avoid such a problem, the reparameterization trick is adapted, so that VAEs can be optimized by the stochastic gradient descent method.

Note that the first and second terms in the right side of Eq. (8) is the regularization and reconstruction term, respectively. Without the first term and the sampling process, it is basically the traditional auto-encoder.
Fig. 2 The VRAE network architecture. The left and right sides of the latent variable \( z \) are the encoder and decoder, respectively. FF means feed forward networks but here we used a simple linear layer. There are two FFs because mean and variances might need different biases. ‘0’ means the initial state of RNN.

VAEs were applied to images at first. To apply VAEs to sequential data like sentences, recurrent neural networks can be considered for the encoder and decoder in VAEs which leads to variational recurrent auto-encoders (VRAEs) [8] as shown in Fig. 2. In other words, VRAEs are variations of VAEs to model sequences with RNNs. We use LSTM as the RNN encoder and decoder in our VRAE. The latent variable is concatenated to the input at every time step in the decoder RNNs.

As generate models for sequence with stochastic behavior, stochastic recurrent networks (STORNs) and variational recurrent neural networks (VRNNs) can also be considered [18,19]. However, STORNs cannot include a priori information contrary to VAE, and VRNNs do not have sentence representation so that we can interpolate points. Thus, we focus on VRAEs.

4. Korean Sentence Generation

In this section, we describe our methods for Korean sentence generation, given a few sample sentences.

4.1 Word Segmentation

As an agglutinative language, segmentation of Korean words is not an easy task. To reduce the number of words which are inputs to our VRAE networks, we need a word segmentation method.

For word segmentation, we use Morfessor [17] to tokenize input sentences. Morfessor is one of probabilistic machine learning algorithms which find morphological segmentations for words of input sentence. With this method, we tokenized input sentences and some examples are presented in Table 1.

4.2 Bidirectional Encoder for VRAE

In the Korean language, there are many cases that the order of words in a sentence has not so much impact on the meaning. See some examples in Table 2, which have the same meaning regardless of the order of the words.

In addition, sometime subject and objects are omitted when they are obvious in the context. It means unidirectional RNN as in language model (LM) might miss some information which can be captured in the opposite direction.

We expect that bidirectional modeling for a sentence might be better than unidirectional one, so that we extend the unidirectional encoder in VRAE to bidirectional one, especially for Korean language. See Fig. 3 for the bidirectional encoder in our VRAE.
4.3 Interpolation in the Encoded Space

To generate new sentences similar to given sentences, we interpolate the given sentences in the encoded space. Actually, a sentence is composed of discrete words (or symbols), it is not possible to interpolate the sentences as sequences of symbols. However, once sentences are encoded into a continuous vector space, interpolation can be applied to the vectors. For example, given two sentences, a point between the two vector points \( p_1 \) and \( p_2 \) can represent a new sentence in the latent space as in Eq. (9), which can be decoded into a new sentence.

\[
p = \alpha p_1 + (1 - \alpha) p_2 \tag{9}
\]

In addition to the interpolation of two points, in general, given multiple points as in Fig. 4, any point \( p \) in the convex of the points can be represented by a weighted sum of the samples as follows.

\[
p = \sum_i w_i p_i \quad \text{where} \quad \sum_i w_i = 1. \tag{10}
\]

To generate samples from \( K \) given sentences, we need to sample \( K \) weights corresponding to the sentences as in Eq. (10). The weights are obtained by sampling \( K \) positive numbers and normalizing the numbers by the sum of the numbers.

5. Experiments

5.1 Datasets

We use monolingual (Korean) sentence dataset which is used for training language model for speech recognition. This dataset has 30M sentences that are tokenized with Morfessor. Table 3 presents a few examples of the sentences in the dataset.

5.2 Bidirectional Encoder

We designed four different models. The first two models use a unidirectional encoder and their dimensions of the latent variable are 128 and 256, respectively. The encoder of the other two models is designed as a bidirectional encoder and their latent variable’s dimensions are 128 and 256, respectively.

The Table 4 summarizes the results of our four different models. See Fig. 5 for the training curves. As the results, the bidirectional encoder models have better performance than the unidirectional encoder models in both perplexity and KL divergence. In other

<table>
<thead>
<tr>
<th>Encoder</th>
<th>Perplexity (Evaluation)</th>
<th>KL divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unidirectional (dim of z=128)</td>
<td>1.076 (1.603)</td>
<td>0.6245</td>
</tr>
<tr>
<td>Unidirectional (dim of z=256)</td>
<td>1.078 (1.522)</td>
<td>0.6101</td>
</tr>
<tr>
<td>Bidirectional (dim of z=128)</td>
<td>1.029 (1.279)</td>
<td>0.6098</td>
</tr>
<tr>
<td>Bidirectional (dim of z=256)</td>
<td>1.030 (1.304)</td>
<td>0.6055</td>
</tr>
</tbody>
</table>
words, as we expected in section 4.2, the bidirectional model can capture more semantic information of input sentences than the unidirectional model. Also, as the dimension of the latent variable become larger, the latent variable can represent more about the input sentences.

5.3 Sentence Generated from Interpolation

In this section, we generate and compare new sentences with two models: unidirectional and bidirectional encoders.

5.3.1 Interpolation between two sentences

In this section, first we generate 20 new sentences with two models (unidirectional and bidirectional encoders) given the following two sentences:

“거기 날씨 가 어때 요 요즘”
“요즘 거기 날씨 가 어때 요”

which have the same meaning but in different word orders.

The generated sentences are generated from the linear interpolation between them. That is, the first generated sample is supposed to be close to “거기 날씨 가 어때 요 요즘”. After generating 20 new sentences, we removed the repeated same sentences generated in a row.

From the results presented in Tables 5 and 6, we can see that the bidirectional encoder improves the quality of the generated sentences, which means that the bidirectional RNNs are more proper to model Korean language. Actually, since there should not be other sentences with different meanings between the given two sentences which already have the same meaning, the generated sentences should be just

<table>
<thead>
<tr>
<th>Table 5 Generated sentences from interpolation between two sentences with a unidirectional encoder (Dim of z = 256)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unidirectional Encoder</strong></td>
</tr>
<tr>
<td>거기 날씨 가 뭐야 요 요즘</td>
</tr>
<tr>
<td>거기 날씨 가 어때 요 요즘</td>
</tr>
<tr>
<td>거기 날씨 가 어때 요 뭐야</td>
</tr>
<tr>
<td>거기 날씨 가 어때 _UNK 뭐야</td>
</tr>
<tr>
<td>거기 날씨 날씨 어때 요 요즈</td>
</tr>
<tr>
<td>거기 날씨 날씨 뭔지 뭐야 뭐야</td>
</tr>
<tr>
<td>거기 날씨 날씨 뭔지 어때</td>
</tr>
<tr>
<td>요즘 날씨 날씨 가 어때</td>
</tr>
<tr>
<td>요즘 거기 날씨 가 어때 요</td>
</tr>
<tr>
<td>요즘 거기 날씨 가 어때 요</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6 Generated sentences from interpolation between two sentences with a bidirectional encoder (Dim of z = 256)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bidirectional Encoder</strong></td>
</tr>
<tr>
<td>거기 날씨 가 어때 요 요즘</td>
</tr>
<tr>
<td>요즘 날씨 가 어때 요 요즘</td>
</tr>
<tr>
<td>요즘 날씨 가 어때 요 요즘</td>
</tr>
<tr>
<td>요즘 날씨 날씨 요 어때 요</td>
</tr>
<tr>
<td>요즘 날씨 날씨 요 어때 요</td>
</tr>
<tr>
<td>요즘 거기 날씨 가 어때 요</td>
</tr>
<tr>
<td>요즘 거기 날씨 가 어때 요</td>
</tr>
</tbody>
</table>

Fig. 5 Training curves. (Top) perplexity of training data, (Middle) KL divergence of training data, and (Bottom) perplexity of test data
한국어 문장 생성을 위한 Variational Recurrent Auto-Encoder 개선 및 활용

Table 7 Generated sentences from interpolation between two sentences with a bidirectional encoder (Dim of \( z = 128 \))

<table>
<thead>
<tr>
<th>Bidirectional Encoder</th>
<th>alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>거기 날씨 가 어때 요 요즘</td>
<td>1</td>
</tr>
<tr>
<td>학원에서 날씨 온도 어때 요 요즘</td>
<td>0.68421053</td>
</tr>
<tr>
<td>학원에서 날씨 온도 어때 요 심각해</td>
<td>0.63157895</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.52631579</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.47368421</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.42105263</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.36842105</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.31578947</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.26315789</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.21052632</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.15789473</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.10526316</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.05263158</td>
</tr>
<tr>
<td>학원에서 날씨 날씨 는데 어때 요</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>

 diferente form of the same meaning. In the tables, the alpha values for interpolation in Eq. (9) are presented.

In addition, we could see that the dimension of the \( z \) space plays an important role to represent the meaning of sentences. From the results in Table 7, it is obvious that the dimension of 128 is not enough to represent sentences in a vector space, because the generated sentences not only have different forms but also have different meanings.

5.3.2 Interpolation from multiple sentences

In this section, we generated 20 sentences with bidirectional encoder model (latent variable dimension = 256) given the following three sentences.

"서울에 내일 날씨는 어때"
"부산에 오늘 온도는 어때"
"광주에 오늘 날씨는 어때"

We interpolated the three sentences with Eq. (10) where \( p_i \) is a latent variable of the \( i \)-th sentence, and \( w_i \) is a randomly generated weight for the \( i \)-th latent variable where \( 0 < w_i \leq 1 \) and \( \sum w_i = 1 \).

Table 8 shows the results of the interpolation of the three sentences with the interpolation weights in Eq. (10). As the results, most of the generated sentences have similar meaning to the input sentences, but have different words like "합포에 오늘 날씨는 어때."

"동신에 내일 날씨는 어때" is similar to the first sentence, and actually affected dominantly by the first sentence. Also, we can see that "합포에 오늘 비는 어때" is made of the three sentences with almost the same amount of effect, as the evenly distributed weights indicate.

6. Conclusions

We introduced a new method for generating Korean sentences with a few input sentences using a VRAE model. We applied a tokenization to input sentences to reduce the number of words in the model. Also, we changed the unidirectional encoder of the original VRAE model into the bidirectional encoder considering some properties of Korean sentences. Lastly, we presented the interpolation from two and multiple sentences. The experiment results show that our proposed method generates new sentences that are similar to the given sentences.

As a future work, to improve the performance, we will apply other tokenization method like BPE (Byte Pair Encoding) instead of Morphessor, since radix and postix are not distinguishable in Morphessor results. Also, we will analyze the latent space to improve the interpolation method, considering that the difference variances of the given sentences might distort the latent space.
References


