GENERATING SOUND WORDS FROM AUDIO SIGNALS OF ACOUSTIC EVENTS WITH SEQUENCE-TO-SEQUENCE MODEL

Shota Ikawa[1], *Kunio Kashino*[1,2]

[1] Graduate School of Information Science and Technology, The University of Tokyo
[2]NTT Communication Science Laboratories, NTT Corporation

ABSTRACT

Representing various sounds in language, such as sound words, or onomatopoeias, is not only useful as an auxiliary means for automatic speech recognition, but also essential in emerging fields such as natural human-machine communication, searching audio archives for acoustic events, and abnormality detection based on sounds. This paper proposes a novel method for sound word generation from audio signals. The method is based on an end-to-end, sequence-to-sequence framework to solve the audio segmentation problem to find an appropriate segment of audio signals along time that corresponds to a sequence of phonemes, and the ambiguity problem, where multiple words may correspond to the same sound, depending on the situations or listeners. Our tests show that the method worked efficiently and achieved a 2.8 % mean phoneme error rate (MPER) and a 7.2 % word error rate (WER) in a sound word generation task.

Index Terms— Sound word, onomatopoeia, sequence-to-sequence model, sound transcription

1. INTRODUCTION

Sound words, or onomatopoeias, refer to the words simulating non-speech sounds, such as acoustic events, within the pronunciation of a certain language system[1]. Using sound words is a way to express acoustic information in a form that humans can easily understand. Actually, we can imagine the original sounds from the words[2], since the words are intended to be reasonably similar to the sounds. Such sound words are widely seen in many languages, including English, Chinese, and Japanese, and they effectively support our daily communication.

Sound words are useful not only for communication among humans, but also between humans and machines. For example, automatic speech recognition systems or conversation systems working closely with us are expected to generate sound words to describe some unknown sounds. Since the sounds words can be viewed as a very efficient form of information compression, they will also be essential for searching audio archivies with words for specific acoustic events and detecting abnormalities through sounds to detect accidents or machine failures.

Here, we propose a novel sound-word generation system based on a sequence-to-sequence conversion framework (hereafter, Seq2Seq) [3]. Section 2 first reviews the existing methods and specifies the problem. Section 3 then introduces our model. Section 4 evaluates the proposed system, and Section 5 concludes the paper.

2. PROBLEMS IN SOUND WORD GENERATION

2.1. Previous work

Environmental sound classification has been widely discussed in the literature. If we can appropriately classify all possible sounds, then we might be able to assign some suitable sound words according to their types[4]. For example, we may be able to assign "bow wow" for dog barking. However, in reality, there are some difficulties in this approach. For example, it is difficult to define the sound classes in advance. Second, the sounds will significantly vary even within a specific class – a dog will not necessarily bark "bow wow" at all times. Third, the sound classification approach may not be able to deal with new, unknown sounds appropriately.

Therefore, here we focus on the sound word generation approach rather than sound classification. We aim to achieve the generation for various featured sound events such as declining sound, sustaining sound and repeating sound. In previous research, finer grained classes, such as phonemes, were assigned to finer grained segments along time. In this approach, it is known that using typical automatic speech recognition methods, depending on language models, does not produce meaningful results[5, 6]. Thus, the problem is twofold: one is segmentation of signals to find a section that corresponds to one phoneme (or a certain unit, such as syllable), and the other is dealing with intrinsic ambiguity involved in the sound word generation task. Here, ambiguity means that even a single sound can correspond to multiple sound words depending on the situations and listeners. For example, some may hear a dog as "bow wow," while others hear it as "wang, wang."

Toward onomatopoeia generation, Miyazaki *et al.* [7] re-

cently proposed the use of connectionist temporal classification (CTC)[8]. The CTC has been successfully applied to phoneme recognition for speech recognition[9]. In [7], the segmentation problem was handled through the training of recurrent neural networks (RNN), but the ambiguity problem was not addressed. Moreover, an approach solely based on direct correspondence between short-time sound segments and a phoneme cannot deal with summarization of repeating sounds. The summarization is commonly used in human communication; for example, when a dog barks five times, we tend to represent it with a fewer times of repetition, like "bow wow."

2.2. Formulation of the problem

When a sound event has a latent variable $\boldsymbol{z}$ which is the summarized feature containing enough information for generating sound words, a sequence $\boldsymbol{l}$, which representing a sound word, is generated based on probability distribution $p(\boldsymbol{l}|\boldsymbol{z})$. The latent variable $\boldsymbol{z}$ is extracted from acoustic features $\boldsymbol{X}$ by mapping $f(\boldsymbol{X} \rightarrow \boldsymbol{z})$. Mapping f corresponds to sound-to-word conversion by humans. Given $p(\boldsymbol{l}|\boldsymbol{z})$ and z, the optimum sequence $\bar{\boldsymbol{l}}$ is obtained by

$$\bar{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l}|\boldsymbol{z}). \tag{1}$$

The problem is then considered as estimating f and probability distribution $p(\boldsymbol{l}|\boldsymbol{z})$. The system generates sound words $\hat{\boldsymbol{l}}$ with estimated mapping $\hat{f}$ and estimated probability distribution $\hat{p}(\boldsymbol{l}|\boldsymbol{z})$.

$$\hat{\boldsymbol{l}} = \arg\max_{\boldsymbol{l}} \hat{p}(\boldsymbol{l}|\hat{f}(\boldsymbol{X})) \tag{2}$$

Note that the segmentation and ambiguity problems are both incorporated in this formulation.

3. METHOD

3.1. System configuration

To solve the problem, we employ the Seq2Seq model[3][10]. It is a combination of the recurrent language model [11] and feature extraction by RNN, and has been successfully applied to inter-sequence conversion[12, 13]. The advantage of this approach is that the model can simply learn correspondence between the observation and the target without determining how to extract the latent features in advance.

As shown in Fig. 1, our system comprises Bi-directional LSTM (BDLSTM)[14] as the encoder, and two-layer LSTM as the decoder. The system takes an audio signal as an input and outputs a series of *sound word units*, terminated with the "EOS" (end of the sequence) symbol. In the following discussion, for simplicity, we use a series of mel-frequency cepstral coefficients (MFCC) as the input[15], and a phoneme as the sound word unit.

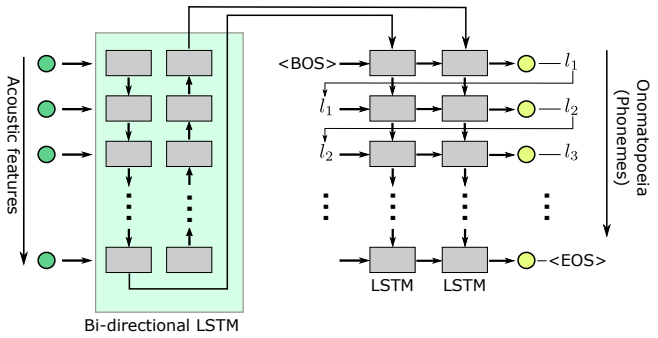

Fig. 1. Block diagram of proposed sound word (onomatopoeia) generation system based on Seq2Seq model.

The encoder further extracts the summarized features from the input MFCC features, and the decoder generates sequence of phonemes based on the extracted features. The encoder learns the mapping $\hat{f}$ and estimates the latent variable $\hat{\boldsymbol{z}}$. Then, the decoder estimates $\hat{\boldsymbol{l}}$ based on $\hat{\boldsymbol{z}}$. Regarding loss functions, we consider two kinds of training methods: vanilla training and multi-task training.

3.2. Vanilla training

Vanilla training is a general training method for Seq2Seq models. The softmax cross entropy between the model output and the target at each time step is used as the loss function. Let $\boldsymbol{O}(t)$ denote an output vector from the decoder at step t, the loss value is obtained by

$$\mathrm{Loss}_{\mathrm{vanilla}}(l_t, \boldsymbol{O}(t)) = -\log\left(\frac{\exp(O_{l_t}(t))}{\sum_i \exp(O_i(t))}\right). \tag{3}$$

This method is expected to yield a reasonable phoneme reflecting the output history of the decoder. On the other hand, this method does not consider the probabilistic formulation shown in Eq.(1) explicitly.

3.3. Multi-task training

From Eq.(1), the decoder should learn $p(\boldsymbol{l}|\boldsymbol{z})$, but directly calculating $p(\boldsymbol{l}|\boldsymbol{z})$ is difficult because of the combinatorial explosion of the number of states as the sequences become longer. Therefore, we specifically focus on the probability distribution of the first phoneme, $p(l_1|\boldsymbol{z})$, to avoid it.

In this method, estimating $p(l_1|\boldsymbol{z})$ is considered a subtask, while the main task is the same as in the vanilla training. The loss function is the mean squared error between the target $p(l_1|\boldsymbol{z})$ and the probability distribution obtained by applying the softmax function to the first output.

$$\hat{p}(l_1|\hat{f}(\boldsymbol{X})) = \mathrm{softmax}(\boldsymbol{O}(1))_{l_1} \tag{4}$$

$$\mathrm{Loss}_{\text{sub-task}} = \sum_{l_1}\left(\hat{p}(l_1|\hat{f}(\boldsymbol{X})) - p(l_1|\boldsymbol{z})\right)^2 \tag{5}$$

This scheme enables the model to learn a part of the probabilistic model as in Eq.(1) as an approximation with a fewer number of state labels compared to the full modeling.

4. EXPERIMENTS

4.1. Dataset

To evaluate the proposed method, we used sound signals of acoustic events from the RWCP (Real World Computing Partnership) sound scene database[16] for training and testing. It includes various sounds such as bells, coins, and hitting wood with a stick. The sampling frequency is 48 kHz, and quantization bit rate is 16 bits. A total of 810 sound sources were divided into eight parts for cross-validation to evaluate the system.

Table 1 describes the datasets we used. The sound word labels were collected from human listeners – 73 Japanese speakers were asked to give three onomatopoeias for each sound in Katakana, which is a Japanese syllabary. Each Katakana answer was converted to a string of International Phonetic Alphabet (IPA) to create the target sound words. In Japanese, onomatopoeias are usually written in Katakana, and it is straightforward to convert from Katakana to IPA, and vice versa. We associated 12 sound words for each sound source for the dataset for the main task. The probability distribution $p(l_1|\boldsymbol{X})$ was also calculated from the answers.

4.2. Procedures

Table 2 lists the experimental conditions. As output phonemes, we used 32 kinds of symbols that consist of the ones compliant with IPA phonetic symbols[18] and Japanese-specific morae:

- "ɴ" : moraic nasal
- "H" : second mora of long vowel
- "Q" : moraic silence when emphatic

In addition, we use three special symbols: "BOS (beginning of the sequence)", "EOS", and "UNK (unknown)."

For comparison, we conducted an experiment with the CTC model. The CTC model has three-layer BDLSTM, as reported in [7]. The outputs were the same 32 kinds of symbols, except for a "blank" symbol used instead of the label "UNK."

Table 1. Sound Sources and Dataset

Sound sources	810
Sound word labels	12 per source
Dataset for main-task learning	9720 pairs
Dataset for sub-task learning	810 pairs

Table 2. Experimental Conditions

LSTM cells	128, 256, 512
Batch size	400
Epoch	25
Number of output labels	32
Optimizing method	ADAM [17]
Multi-task pretraining	30
Sub-task iteration	12 per epoch
MFCC dimensions	20
Window length for MFCC	2048 samples
Window shift for MFCC	512 samples

We performed all the tests with the eight-fold cross-validation scheme.

4.3. Evaluation measures

We use the word error rate (WER) and the mean phoneme error rate (MPER) as the evaluation measures.

The WER is simply calculated by n/N, where N is the number of test data and n is the number of generated words that exactly matches the target words associated with the sound. This metric does not reflect the similarity between the words.

The phoneme error rate (PER) serves as a more flexible performance metric than the WER. As in the Eq. 6, the PER is the "edit distance" between two phoneme sequences, normalized by the length of target phonemes.

$$\mathrm{PER} = \frac{\text{Replacement Err.+Insertion Err.+Deletion Err.}}{\text{The Number of Target Phonemes}} \quad (6)$$

To calculate the mean PER (MPER), the PER for each output word and each of the corresponding multiple target words is first calculated to find the minimum PER, and the obtained minimum values are averaged over the whole test set.

4.4. Results

Table 4 lists the error rates obtained in the tests. The model with vanilla training with 512 LSTM cells produced the best results, and the MPER was 2.8 %, which is quite small compared to the result from the CTC model. The difference between the results with the vanilla and multi-task training was not significantly large in this experiment.

Fig. 2 shows the learning curves with respect to WERs and MPERs, obtained from one specific trial with 512 LSTM cells, as an example. We can see that both vanilla (Va.) and multi-task (M.t.) learning converge within about 20 epochs, showing similar curves.

Some examples of generated sound words are listed in Table 3. Among the 105 experimented test sounds, there was

Table 3. Examples of Generated Sound Words

Name	Typical sound words	Seq2Seq (multi-task)	Seq2Seq (vanilla)	CTC	Description
buzzer	buɯH, gaH	b ɯ H	b ɯ H	b ɯ H	A muddy sound with a constant low pitch
wood3	kaQ, kataQ	k a Q	t a Q	k a Q	A light sound by hitting a wooden board
teak3	koQ, toɴ	p o ɴ	p o q	p o ɴ	A sound by hitting a wooden board with a wooden stick
cap1	paQ, poQ	p o k ɯ	p a Q	p a Q	A sound of opening a cap vigorously
metal05	toɴ, baɴ	t o ɴ	p a Q	p a Q	A dark sound by tapping a metal plate with a metal rod
whistle1	piH,çiH	p i H	p i H	p i i H H	A whistle-like sound with a constant high pitch
bells4	ʃarириri	t ʃ i r i ɴ	t ʃ i r i ɴ	t ʃ r r r r r ⋯	A cyclic sound of bells
candybwl	boɴ,koɴ	t o ɴ	d o ɴ	t a ɴ ɴ	A sound by hitting a metal box with a metal rod
coins2	kotʃarириriH	k o r o r i H ɴ	tʃarириririri	k r r ⋯ r i H ɴ	A sound of multiple coins dropped on wood

Table 4. Error Rates

Model	#Cells	WER[%]	MPER[%]
proposed 1: **Seq2Seq (vanilla)**	128	24.4	9.4
	256	9.6	4.0
	512	7.2	2.8
proposed 2: **Seq2Seq (multi-task)**	128	29.1	10.9
	256	12.3	4.9
	512	9.9	4.1
CTC [7]	128	85.1	37.7
	256	79.9	33.0
	512	78.9	37.0

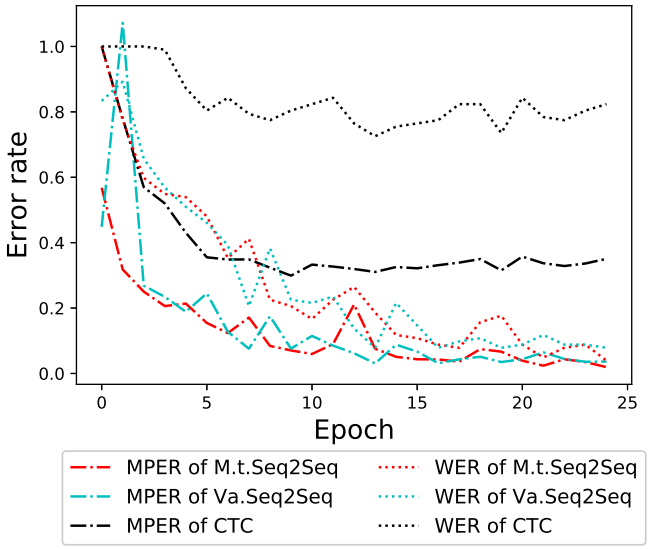

Fig. 2. Learning curves of the experimented models.

only one input sound where the three methods, vanilla, multi-task, and CTC, output the same words, which was the sound of a buzzer. Meanwhile, the number of input sounds for which each method outputs different words was 81.

Note that WER and MPER do not weight target labels, and naturally sounding words and barely acceptable ones are treated in the same way. Therefore, subjective observation or evaluation is useful to check which words better represent input sounds.

From our observation, words generated by the CTC model tend to correspond to the input in a flexible manner, but they are sometimes unnatural and hard to pronounce. In particular, the CTC model tends to output poor results for cyclic or repetitive sounds such as coins lightly hitting each other and an alarm clock ringing. With the Seq2Seq models, it is observed that this phenomenon is greatly diminished. We think that this is an advantage arising from incorporating contextual information.

5. CONCLUSION

This paper proposed a novel end-to-end method to generate sound words from audio. We first presented a probabilistic formulation of the problem that incorporates segmentation and ambiguity problems. We then proposed a system based on the Seq2Seq model as a solution to the problems. The experiments show that the method effectively works and yields significantly better results compared to an implementation based on the CTC model. We expect that the sound words or onomatopoeias will open the way to representing sounds in a compact and human-friendly manner and will therefore be useful in applications such as audio search and abnormality detection, as well as auditory scene description systems. Our future work will include tests with languages other than Japanese, subjective evaluation of the model, and a study regarding its usability.

6. ACKNOWLEDGEMENTS

We thank Prof. Hiroshi Saruwatari and Dr. Shinnosuke Takamichi for their valuable comments and support.

7. REFERENCES

[1] S. Sundaram and S. Narayanan, "Classification of sound clips by two schemes: Using onomatopoeia and semantic labels," in *2008 IEEE International Conference on Multimedia and Expo*, June 2008, pp. 1341–1344.

[2] Kohei Hayashida, Yu Mizoguchi, Junpei Ogawa, Masanori Morise, Takanobu Nishiura, and Yoichi Yamashita, "The acoustic sound field dictation with hidden markov model based on an onomatopeia," vol. 5, 01/2010.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 3104–3112. Curran Associates, Inc., 2014.

[4] Fangzhou Wang, Hidehisa Nagano, Kunio Kashino, and Takeo Igarashi, "Visualizing video sounds with sound word animation to enrich user experience," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 418 – 429, 2017.

[5] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in hmm speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, Jan 1994.

[6] K. Ishiguro, Y. Tsubota, and H. Okuno, "Automatic transformation of environmental sounds into sound-imitation words based on japanese syllable structure," in *Proc. EUROSPEECH*, 2003, pp. 3185–3188.

[7] K. Miyazaki, T. Hayashi, and K. Takeda T. Toda, "Conversion from sound event to onomatopoeia representation based on ctc," in *2017 Autumn Meeting*. Sep 2017, Acoustic Society of Japan.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, ICML '06, pp. 369–376, ACM.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013.

[10] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[11] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010, pp. 1045–1048.

[12] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition," *CoRR*, vol. abs/1508.04395, 2015.

[13] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4845–4849.

[14] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602 – 610, 2005, IJCNN 2005.

[15] Michael Cowling and Renate Sitte, "Comparison of techniques for environmental sound recognition," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2895 – 2907, 2003.

[16] S. Nakamura, K. Hiyane, F. Asano, T. Yamada, and T. Endo, "Data collection in real acoustical environments for sound scene understanding and hands-free speech recognition," in *Proc. EUROSPEECH*, Sep. 1999, pp. 2255–2258.

[17] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[18] International Phonetic Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*, Cambridge University Press, 1999.