NEURAL MACHINE TRANSLATION WITH ACOUSTIC EMBEDDING

Takatomo Kano[1]*, Sakriani Sakti*[1,2]*, and Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan

ABSTRACT

Neural machine translation (NMT) has successfully redefined the state of the art in machine translation on several language pairs. One popular framework models the translation process end-to-end using attentional encoder-decoder architecture and treats each word in the vectors of intermediate representation. These embedding vectors are sensitive to the meaning of words and allow semantically similar words to be near each other in the vector spaces and share their statistical power. Unfortunately, the model often maps such similar words too closely, which complicates distinguishing them. Consequently, NMT systems often mistranslate words that seem natural in the context but do not reflect the content of the source sentence. Incorporating auxiliary information usually enhances the discriminability. In this research, we integrate acoustic information within NMT by multi-task learning. Here, our model learns how to embed and translate word sequences based on their acoustic and semantic differences by helping it choose the correct output word based on its meaning and pronunciation. Our experiment results show that our proposed approach provides more significant improvement than the standard text-based transformer NMT model in BLEU score evaluation.

Index Terms— Neural machine translation, acoustic and semantic embedding representation

1. INTRODUCTION

An end-to-end deep learning framework provides an emerging approach for sequence-to-sequence mapping tasks and allows a model to directly learn the mapping between the variable-length representation of the input and the output. Most successful neural sequence translation models are based on end-to-end attentional encoder-decoder architecture [1, 2, 3, 4]. The words of the input sequences from a source language are first encoded into vectors of intermediate representation and passed to the decoder. A compressed context vector is derived by applying an attention mechanism, which measures the alignment between the source and target texts. The decoder output layer takes the embedded vector and the previously translated word as input and produces a target translated word at the current step.

In such architecture, word embeddings in continuous vector representations are an almost ubiquitous NMT component. These representations are sensitive to the meaning of words and their accurate order and reasonably insensitive to the replacement of the active voice with a passive voice [1]. These representation's behavior assumes that words in similar contexts have similar meanings. Such embeddings represent the semantics of the corresponding words/sequences, allowing semantic similar words to be grouped together in the vector spaces to share statistical power. The embedding layer provides advantages that increase the robustness for rare data and produce more natural outputs than statistical phrase-based translation [5].

Unfortunately, the model often maps such similar words too closely, which complicates distinguishing them. Consequently, NMT often generates words that seem natural in the target sentence that do not reflect the source sentences original meaning. Many studies also argued that NMTs translations are often fluent but lack accuracy [6, 7, 8]. For example, the system mistakenly translated "may I" for "can I", "dog" for "cat," "Norway" for "Tunisia," and so on. Although it does not destroy the overall naturalness, the sentences entire meaning might be completely different, which makes the error critical.

Several studies incorporated auxiliary features that were integrated into the word vectors. To date, such linguistic features as lemmas improved the NMT results when they are appended to the word vector at the encoder [9] or decoder [10]. Incorporating latent Dirichlet allocation (LDA) [11] topic vectors also provides advantages [12]. Auxiliary information in the form of multi-modal streams with images has also been integrated into NMTs [13, 14]. Deena et al. [15] investigated another modality with acoustic information in which audio features as show-level i-vectors [16] and Latent Dirichlet Allocation (LDA) [11] topic vectors are incorporated within NMT and improved the translation quality more than with text-based NMT. Most of these works only focused on the enhancement of the embeddings part but did not changed the NMT architecture.

Another approach is to consider the overall speech translation as a multimodal translation task. An extreme case is training the encoder-decoder architecture for end-to-end speech translation (ST) tasks, which directly translate the speech in one language into text in another [17, 18, 19]. Su et al. [20] and Sperber et al. [21] incorporated an automatic speech recognition (ASR) lattice by replacing the encoder part with a lattice encoder to obtain a lattice-to-sequence model. Although it provided significant advantages, the approach required a large modification for standard NMT systems. Osamura et al. proposed a simpler solution to incorporate speech information by an ASR posterior vector. This might resemble word confusion networks (WCNs) [22] that can directly express the ambiguity of word hypotheses at each time point. Osamura et al. [23] reported that acoustic information helps distinguish such semantic similar words as "cut" and "perm" in the encoder part, which helps the decoder find correct attention points and output correct words in the target language. However, these works only focused on the source speech from the source language that was incorporated into the NMT encoder part.

In this research, we learn how to incorporate acoustic information from the target language in collaboration with a text-to-speech (TTS) system. We integrate acoustic information within the NMT decoder by multi-task learning. Our model learns how to embed and translate the word sequences based on their acoustic and seman-

tic differences to help the model choose the correct output word by considering its meaning and pronunciation. To the best of our knowledge, this is the first study that improved NMT in collaboration with TTS. In our proposed method we use a state of the art sequential translation model transformer with tied-embedding as a baseline. Our experiment results show that our proposed approach provides greater improvements than the standard text-based NMT model.

2. TRANSFORMER FRAMEWORK

2.1. Standard transformer architecture

A transformer is a encoder-decoder sequence-to-sequence transaction model without recurrent mechanics. The encoder maps an input sequence of symbol representations $\boldsymbol{x} = [x_1, ..., x_n]$ to a sequence of continuous representations $\boldsymbol{h} = [h_1, ..., h_n]$. Given $\boldsymbol{h}$, the decoder generates output sequence $\boldsymbol{y} = [y_i, ..., y_I]$ of the symbols one element at a time.

The transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder [24]. The encoder is composed of a stack of multiple layers, each of which has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a position-wise fully connected feed-forward network (FNN). The transformer has a residual connection around each of the two sub-layers, followed by layer normalization [25, 26]. The decoder is also composed of multiple layers like the encoder. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the encoder stacks output. The attention function resembles mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Fig. 1 illustrates the overall architecture of the transformers.

Basically, the transformer model offers two benefits: (1) it enables parallel training by removing recurrent connections, such as an input sequence so that a decoder can be provided in parallel; (2) self-attention provides an opportunity for injecting the global context of the whole sequence into each input frame to directly build long-range dependencies. When perform forward and backward processing for current input and output, Transformer only makes the calculation graph path for relative states that attended by self-attention mechanism. But the RNN based encoder decoder model makes the calculation graph path for all previous states because it models the global context with recurrent structures. This provides great support to reduce memory resource in such long sequence-to-sequence tasks as the prosody of synthesized speech and character-based translation, which depends on both several neighboring components and the overall sequence. In this research, we construct a character-based transformer for NMT and TTS and describe it in more detail in the following section.

2.2. Transformer NMT

We constructed a Japanese-to-English text translation system in which the input is Japanese sub-word sequences, and the output is English character sequences. Most encoder-decoder systems perform teacher-forcing during training and auto-regression [27] during tests. The decoder part is usually affected by its decoding error during autoregression. This phenomenon has been well observed in character-based decoding. There are two solutions for this problem:

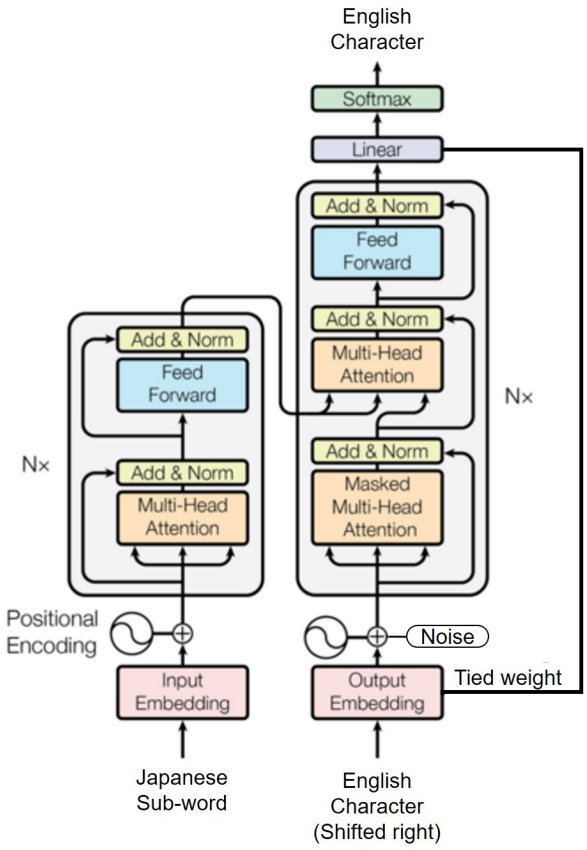


Fig. 1. Transformer architecture in this research slightly modified from original transformer [24]

.

applying a high dropout ratio (over 0.7) to the decoder embedding vector or adding Gaussian noise to the decoder embedding vector. In this research, we use Gaussian noise to make the decoder robust to decoding error. Our decoder embedding part accommodates the following input vector, the noise, and the position information: First, the input embedding is defined:

$$e_i = \boldsymbol{W} y_i, \tag{1}$$

where $\boldsymbol{W}$ is the embedding weight and y_i is the ith input one-hot vector. Here we use noise features based on the embedding vector scale:

$$noise_i = \alpha * \mathit{Noise}(e_i), \tag{2}$$

where *Noise* is the Gaussian noise distribution and α is the noise rate, which in this research we set to 0.2. Then we defined the position embedding vector:

$$pe_i = \mathit{PE}(i), \tag{3}$$

where *PE* means position encoding [28] that provides the current decoding position information as a vector. Finally we computed our decoder embedding vector:

$$emb_i = e_i + noise_i + \beta * pe_i. \tag{4}$$

We employ a trainable parameter to the weighting position information, the same as Transformer TTS [29].

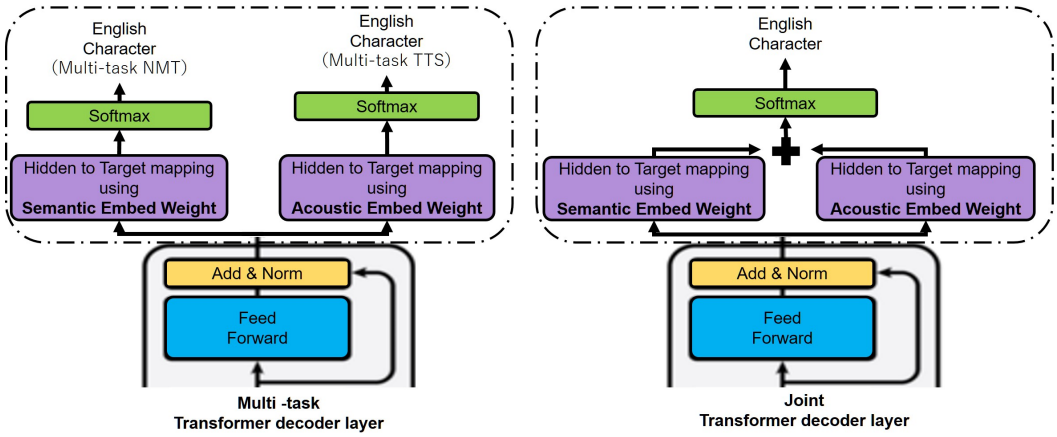


Fig. 2. Proposed models architectures

2.3. Transformer TTS

We also created a Transformer TTS [29]. Transformer TTS has basicaly same architecture with original Transformer except the pre-process and post-process part. Transformer TTS utilizes Tacotron2 [30] pre-process and post-process part. Tacotron2 employs three layers CNN. We applied a convolutional neural network (CNN) to the input text embeddings to handle the longer-term context in the input character sequence. The original Transformer TTS model [29] using the English phoneme sequence as input. But in this model, we input the English character sequence like Tacotron2.

In the decoder part, the Mel spectrogram is first consumed by a neural network that is composed of two fully connected layers with ReLU activation. Since input the English characters have trainable embeddings, their subspace is adaptive, and that of the Mel spectrograms is fixed. The decoder pre-net is responsible for projecting the Mel spectrograms into the same subspace as the encoder character embeddings to measure the similarity of a characters pronunciation and its Mel frame pair. Thus the attention mechanism can work. We also tried two fully connected layers without non-linear activation, but no reasonable attention matrix was generated that aligns the hidden states of the encoder and the decoder [29]. We simply use the Griffin-Lim algorithm [31] instead of WaveNet [32]. Ultimately we only apply the embedding weight of the TTS encoder since improving the synthesized speech quality is not our main focus.

3. PROPOSED APPROACH: INCORPORATING ACOUSTIC EMBEDDING INTO NMT

An encoder-decoder translation model maps an input sequence into a fixed-dimension vector [1]. Such representations are sensitive to the meaning of the sequence and accurate word order, but they are insensitive to the replacement of the active voice with the passive voice. In speech synthesis models, these representations are sensitive to the replacement of the active voice with the passive voice but insensitive to the meaning of the sequence.

These attributes create models that are robust to test sets and generate natural sequences. On the other hand, the model sometimes confuses output with similar meaning words and context like a "dog" and "cat" and "may I" and "can I." In this research we map an input sequence into vectors of intermediate representation that is sensitive to both the sequences meaning and its pronunciation. We expect the pronunciation information to help discriminate among words with similar meanings and contexts in translation. We consider meaning and pronunciation using speech as either input or output. But end-to-end speech translation usually decreases the translation quality and preparing a natural speech parallel corpus is difficult. In this research we used a pre-trained TTS embedding weight for the NMT output layer. The transformer decoder has two modules that handle the target word: a target word embedding layer and an output layer. We treat these two as inverse mappings and tied their weights [33]. In this research, we tied a decoder embedding layer weight and a decoder output layer weight. We added a new output layer where the mapping decoder was hidden using TTS embedding weights that were not updated during training. This model has two types of output layers. The standard decoder output layer weight is tied to the decoder embedding weight. This output layer maps decoder hidden to output for a sensitive sequence meaning. The output layer, is tied with a TTS embedding weight, maps decoder hidden for sensitive sequence pronunciations. We use these to output the results and back-propagate the loss:

$$\boldsymbol{o}_{nmt} = \boldsymbol{W}_{nmt}\boldsymbol{h}_{dec}, \quad (5)$$

$$\boldsymbol{o}_{tts} = \boldsymbol{W}_{tts}\boldsymbol{h}_{dec}, \quad (6)$$

$$loss = (1-\lambda)CE(\boldsymbol{o}_{nmt}, \boldsymbol{y}) + \lambda CE(\boldsymbol{o}_{tts}, \boldsymbol{y}). \quad (7)$$

Here h_{dec} is a decoder hidden sequence, $\boldsymbol{W}_{nmt}$ denotes a decoder word embedding weight, and $\boldsymbol{W}_{tts}$ denotes the TTS encoder embedding weight. In this work, $\boldsymbol{W}_{tts}$ is not update during training. We only update $\boldsymbol{W}_{nmt}$ throgh training. We using soft-max cross entropy (CE) to individually calculate the loss for each output, and λ is the weight for each loss. We call our proposed method multi-task learning:

$$\boldsymbol{o}_{y} = \boldsymbol{o}_{nmt} + \boldsymbol{o}_{tts}, \quad (8)$$

$$loss = CE(\boldsymbol{o}_{y}, \boldsymbol{y}). \quad (9)$$

We sum both NMT and TTS weight mapping. Since we do not update the TTS embedding weight during training, the model updates the NMT embedding weight scale based on the degree of each layer's contribution. If the output from the NMT embedding scale greatly exceeds the output from the TTS embedding, then the proposed model resembles a standard NMT. We call our proposed method joint learning. We summary this section in Fig3.

Table 1. Translation results of Japanese-to-English

Source	Miruk wo mou sukoshi ku dasa i
Target	a little more milk please
Text-based NMT	**let me have some** more milk
Multi-task$_{NMT}$	a little more milk please
Multi-task$_{TTS}$	a little more milk please
Joint	a little more milk please
Source	Gyuniku to Chikin no dochira ga yoroshi i de su ka
Target	which would you like beef or chicken ?
Text-based NMT	** beef or chicken ?
Multi-task$_{NMT}$	which would you like beef or chicken ?
Multi-task$_{TTS}$	which would you like beef or chicken ?
Joint	which would you like beef or chicken ?
Source	i i o tenki de su ne
Target	it 's a lovely day is n't it ?
Text-based NMT	**beautiful weather** is n't it ?
Multi-task$_{NMT}$	**nice** day is n't it ?
Multi-task$_{TTS}$	**nice** day is n't it ?
Joint	**nice weather** is n't it ?
Source	Zyo han shin wo kita e ta i nn de su kedo dono ma shin wo tuka e ba i de su ka
Target	i 'd like to work on my upper torso which machines should i use ?
Text based NMT	i would like to **build you medicine** my upper body ?
Multi-task$_{NMT}$	i 'd like to work on my upper body **what** machine should i use ?
Multi-task$_{TTS}$	i 'd like to work on my upper body **what** machine should i use ?
Joint	**excuse me** i 'd like to **keep** my upper body which machine should i use ?
Source	san de su
Target	three
Text-based NMT	three
Multi-task$_{NMT}$	**from** three
Multi-task$_{TTS}$	three**of us**
Joint	**us** three

4. EXPERIMENT

We conducted our experiments using a basic travel expression corpus (BTEC) [34, 35]. The BTEC Japanese-English parallel corpus consists of 480-k utterances. We removed the sentences that have more than 100 characters and used this dataset to build a baseline and proposed sub-words for the characters for the Transformer NMT. Since the corresponding speech utterances for this text corpus are unavailable, we used the Google text-to-speech synthesis[1] to generate a speech corpus of the target language. We segmented the speech utterances into multiple frames with a 50-ms window and 12.5-ms steps and extracted 80-dimension Mel-spectrogram features using Librosa[2]. We also used these data to build a Transformer TTS. Our proposed model uses this pre-trained TTS acoustic embedding weight. Figure 4 illustrates the attention matrix of the pre-trained transformer. Our TTS model shows clear monotonic shape attention and achieves a 0.05 L1 loss of the TTS Mel-spectrogram from grand truth decoding. The TTS module embedding layer is well trained from these results.

Next we demonstrate our proposed translation performance and compare it with the baseline text-based NMT model. We used OpenNMT[3] to make a baseline and implemented our proposed model on it. Here is a summary of baseline and our proposed models:

Baseline Text-based NMT
This is a baseline text-to-text translation model. Input Japanese (Fig. 1).

Proposed Multi-task$_{NMT}$
Proposed model with multi-task learning and NMT embedding weight output layer in test decoding (Fig. 2.2).

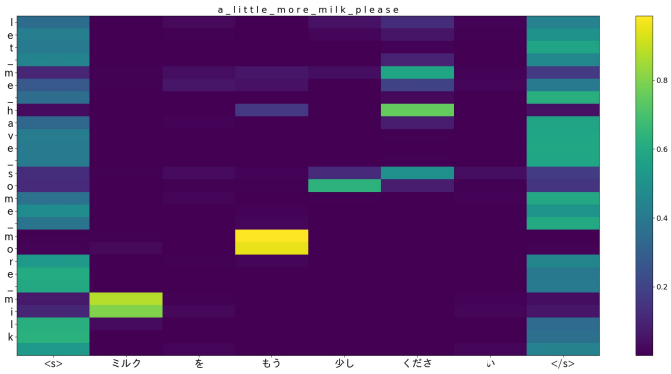
Proposed Multi-task$_{TTS}$
Proposed model with multi-task learning and TTS embedding weight output layer in test decoding (Fig. 2.2).

Proposed Joint
Proposed model with joint learning and both output layer in test decoding (Seq. 3).

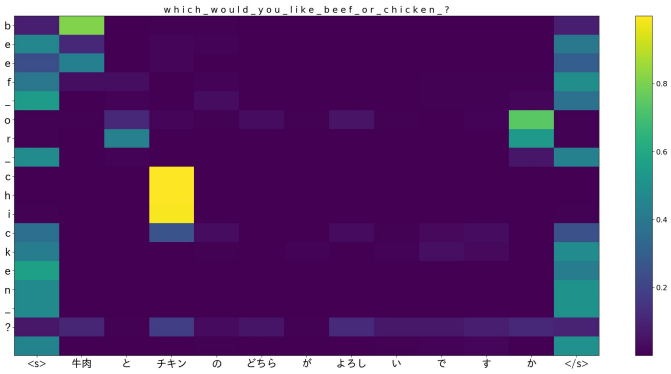
All models performed a beam search (beam size is 5) algorithm for character sequence auto-regressive decoding. Here we summary the model parameters in Tables 2 and 3 2-3. The baseline text-based NMT and our proposed model used the same settings, and the trainable number parameters are the same between the proposed model and the baseline.

Table 4 shows that our proposed method successfully improved the BLEU scores by 5-points from the text-based NMT. For further discussion of the model behaviors, Table 1 lists the translation results from each model. Each proposed model output a sentence whose meaning was very similar to the meaning of the target sentence. This means that each proposed model extracted the meaning of the source sentence and mapped it to the decoder state. But in

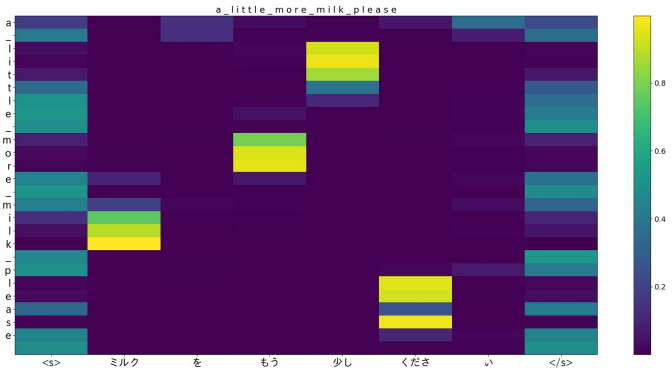
[1]Google TTS:https://pypi.python.org/pypi/gTTS
[2]Librosa: https://librosa.github.io/librosa/
[3]OpenNMT: http://opennmt.net/



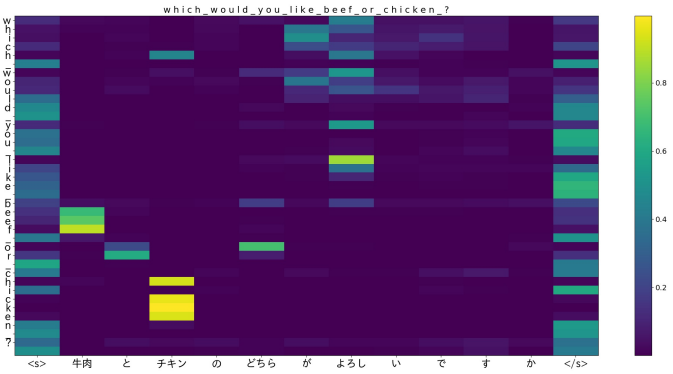
(a) Text-based NMT with correct attention



(b) Text-based NMT with wrong attention



(a) Proposed model with correct attention



(b) Proposed model with correct attention

Fig. 3. Attention table of text-based NMT and proposed model

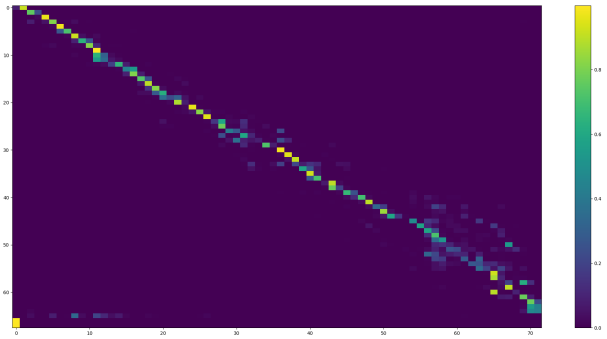


Fig. 4. TTS attention table: TTS Mel-spectrogram L1 loss is 0.05 with grand truth

Table 2. Transformer settings

Embeddings	
Source vocabulary	6439
Target vocabulary	35
Embedding size	512
Noisy ratio	0.2
Transformer block	
Hidden size	512
Number of layers	3
Transformer FFN	2048
Self-attention head	8
Dropout ratio	0.1
Attention mechanism	Multi-head

contrast in the text-based NMT baseline, the text-based NMT model failed to choose the correct word from the decoder state. The output layer is usually one simple linear regression layer that maps a vector from a continuous narrow space to a large discrete space. If the model maps a similar word too closely, then the output layer cannot separate it again. On the other hand, our proposed model output a correct word for each sentence. This reveals that by incorporating acoustic embedding and constructing a model in a multi-task fashion with two output layers, each layer can map the decoder state to different output with different weights. The hidden representation might be sensitive for both semantic and pronunciation similarities. Therefore, our proposed model can choose the correct word that not only depends on its meaning but also on its pronunciation.

We also show two attention table pairs to compare our proposed model multi-task and the baseline. Both models generated similar sentences with correct attention. We believe translation error occurred at the decoder output layer, not at the encoder or attention

Table 3. Optimizer setting

Optimizer	
Method	Adam [36]
Adam $\beta 1$	0.9
Adam $\beta 2$	0.998

Table 4. Translation quality of Japanese-to-English

Model	BLEU score	WER
Text-based NMT	45.10	35.5%
Multi-task$_{NMT}$	**50.51**	30.5%
Multi-task$_{TTS}$	50.23	**30.1%**
Joint	48.12	32.4%

sides. Our model also attended to the same word but generated correct words. Our decoder part separated sequences with similar meanings, as we expected.

We show another attention table where the baseline made some attention error. The baseline only focused on "beef" and "chicken." These are the most important words for translating this input sentence. Other source words (except the last words) are formula words and last words denote questions. In Japanese-English travel conversation translation, since there are many insertions and deletions, not all of the source input are attended during translation. Therefore, our baseline NMT only attended to "beef" and "chicken." On the other hand, our model correctly attended to all of the words. Our model became sensitive to both the meaning and the pronunciation in the decoder hidden state. A benefit also appears in the encoding and attention module through back-propagation. Our model found correct attention in this case. But in another case, at the bottom of Table 1, our proposed method generated unnecessary words.

5. CONCLUSION

We used TTS embedding weight to map translation results. This approach created an NMT model that is sensitive to sequence meaning and pronunciation. Our proposed method outperformed a standard transformer with BLEU scores. We first considered NMT and TTS collaboration. Our proposed method made an NMT that can learn such multi-modal information as text meaning and pronunciation from a text. Future work will consider ASR, NMT, and TTS deep joint optimization to improve the translation performance and improve NMT so that it can handle other kinds of information, such as images. Furthermore, we will apply our proposed approach to more difficult translation data such as TED Talks.

6. ACKNOWLEDGEMENTS

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237.

7. REFERENCES

[1] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 3104–3112.

[2] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 577–585.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[4] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 4006–4010.

[5] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007.

[6] Philip Arthur, Graham Neubig, and Satoshi Nakamura, "Incorporating discrete translation lexicons into neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 1557–1567.

[7] Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang, "Neural machine translation advised by statistical machine translation," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 3330–3336.

[8] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016.

[9] Rico Sennrich and Barry Haddow, "Linguistic input features improve neural machine translation," in *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, 2016, pp. 83–91.

[10] Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares, "Factored neural machine translation architectures," in *2016 Proc. of International Workshop on Spoken Language Translation, 2016 IWSLT 2016*, 2016.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[12] Jian Zhang, Liangyou Li, Andy Way, and Qun Liu, "Topic-informed neural machine translation," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, 2016, pp. 1807–1817.

[13] Po-Yao Huang, Frederick Liu, Sz-Rung Shiang, Jean Oh, and Chris Dyer, "Attention-based multimodal neural machine translation," in *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, 2016, pp. 639–645.

[14] Ozan Caglayan, Loïc Barrault, and Fethi Bougares, "Multimodal attention for neural machine translation," *CoRR*, vol. abs/1609.03976, 2016.

[15] Salil Deena, Raymond W. M. Ng, Pranava Swaroop Madhyastha, Lucia Specia, and Thomas Hain, "Exploring the use of acoustic embeddings in neural machine translation," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*, 2017, pp. 450–457.

[16] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[17] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," *TACL*, vol. 7, pp. 313–325, 2019.

[18] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 2625–2629.

[19] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura, "Structured-based curriculum learning for end-to-end english-japanese speech translation," *CoRR*, vol. abs/1802.06003, 2018.

[20] Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 3302–3308.

[21] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 1380–1389.

[22] Lidia Mangu, Eric Brill, and Andreas Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[23] Kaho Osamura, Takatomo Kano, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura, "Using spoken word posterior features in neural machine translation," vol. 21, pp. 22, 2018.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 6000–6010.

[25] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

[27] Alex Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013.

[28] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 1243–1252.

[29] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou, "Close to human quality TTS with transformer," *CoRR*, vol. abs/1809.08895, 2018.

[30] Yifan Liu and Jin Zheng, "Es-tacotron2: Multi-task tacotron 2 with pre-trained estimated network for reducing the over-smoothness problem," *Information*, vol. 10, no. 4, pp. 131, 2019.

[31] Daniel W. Griffin and Jae S. Lim, "Signal estimation from modified short-time fourier transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, 1983, pp. 804–807.

[32] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[33] Nikolaos Pappas, Lesly Miculicich Werlen, and James Henderson, "Beyond weight tying: Learning joint input-output embeddings for neural machine translation," in *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, 2018, pp. 73–83.

[34] Gen-ichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto, "Creating corpora for speech-to-speech translation," in *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*, 2003.

[35] Gen-ichiro Kikui, Seiichi Yamamoto, Toshiyuki Takezawa, and Eiichiro Sumita, "Comparative study on corpora for speech translation," *IEEE Trans. Audio, Speech & Language Processing*, vol. 14, no. 5, pp. 1674–1682, 2006.

[36] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.