Sequence-to-Sequence Models for Emphasis Speech Translation

Quoc Truong Do, *Student Member, IEEE,* Sakriani Sakti, *Member, IEEE,* and Satoshi Nakamura, *Member, IEEE*

***Abstract*—Speech-to-speech translation (S2ST) systems are capable of breaking language barriers in cross-lingual communication by translating speech across languages. Recent studies have introduced many improvements that allow existing S2ST systems to handle not only linguistic meaning but also paralinguistic information such as emphasis by proposing additional emphasis estimation and translation components. However, the approach used for emphasis translation is not optimal for sequence translation tasks and fails to easily handle the long-term dependencies of words and emphasis levels. It also requires the quantization of emphasis levels and treats them as discrete labels instead of continuous values. Moreover, the whole translation pipeline is fairly complex and slow because all components are trained separately without joint optimization. In this paper, we make two contributions: (a) we propose an approach that can handle continuous emphasis levels based on sequence-to-sequence models, and (b), we combine machine and emphasis translation into a single model, which greatly simplifies the translation pipeline and make it easier to perform joint optimization. Our results on an emphasis translation task indicate that our translation models outperform previous models by a large margin in both objective and subjective tests. Experiments on a joint translation model also show that our models can perform joint translation of words and emphasis with one-word delays instead of full-sentence delays while preserving the translation performance of both tasks.**

***Index Terms*—Emphasis estimation, emphasis translation, speech-to-speech translation, joint optimization of words and emphasis.**

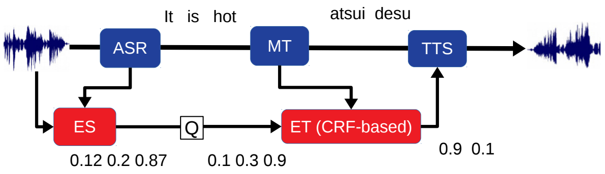

Fig. 1. Existing emphasis speech translation model that consists of many separate components and dependencies and also requires emphasis quantization (Q) before translation.

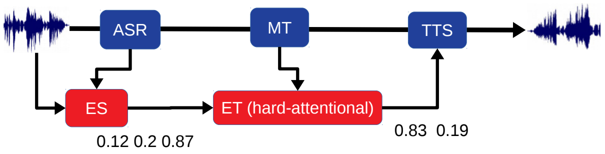

Fig. 2. Proposed hard-attention emphasis speech translation model that can translate continuous emphasis weights without quantization.

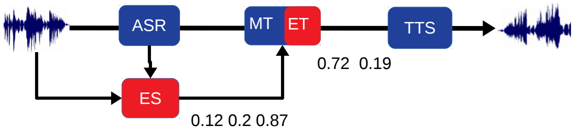

Fig. 3. Proposed joint model simplifies translation pipeline and jointly translates words and emphasis with one-word delay.

I. Introduction

SPEECH is one of the world's richest and most powerful communication channels, allows speakers to express not only the content that they want to convey but also paralinguistic information such as emotion and emphasis. Emphasis is often used to distinguish between the focused and unfocused parts of an utterance [1] and is particularly useful in misheard situations where speakers need to repeat the most important words or phrases of sentences. In speech-to-speech translation tasks, Tsiartas et al. [2] has conducted a study on multilingual speech corpora and argued that emphasis information is a critical factor that contributes to the quality of speech-to-speech translation performance.

Many studies have developed and improved automatic S2ST translation systems [3] that help translate the content of speech across languages. An S2ST system consists of 3 main components: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS). However, since most S2ST systems cannot translate emphasis conveyed in the source language speech, communication through traditional S2ST systems is less engaging than natural speech communication.

The difficulty in handling emphasis is that it can be manifested by changing different types of acoustic features, such as the duration, the power, or the F_0 of the emphasized words [4]. The challenge in developing an S2ST system that can accurately translate emphasis is that we must consider these acoustic features of emphasis in three components: emphasis extraction, emphasis translation, and the synthesis of emphasized speech. Kano et al. [5,6] proposed approaches to translate emphasis in a limited domain of 10 digits. Since their approach models speech differently for each word in the vocabulary, they cannot generalize to unseen words, and they also have difficulty modeling emphasis in large vocabulary systems. Anumanchipalli et al. [7], Aguero et al. [8], and Tsiartas et al. [9] proposed approaches for mapping F_0 into a discrete set of units and translating them across languages. However, other acoustic features such as duration and power,

Q.T. Do is with Nara Institute of Science and Technology, Japan. E-mail: do.truong.dj3@is.naist.jp.

S. Sakti and S. Nakamura are with Nara Institute of Science and Technology and RIKEN Center for Advanced Intelligence Project AIP, Japan. Emails: ssakti@is.naist.jp, s-nakamura@is.naist.jp

were not taken into account.

The most recent work by Do et al. [10] attempts to translate emphasis in an open domain. Their basic idea is to represent emphasis as a continuous real-numbered value (emphasis level) that is estimated using all of the acoustic features. Then they use the conditional random fields to translate the estimated emphasis levels to a target language. Using emphasis levels makes the emphasis representation more intuitive and easier to translate because emphasis translation can be viewed as translating a sequence of numbers.

However, the following three problems remain: (1) Emphasis translation is based on conditional random fields (CRFs) that require emphasis levels of quantization that are treated as discrete labels instead of continuous values. This leads to objective functions that cannot capture the "amount of difference" between emphasis levels. For examples, the levels of 0.7 and 0.6 are quite close if they are considered real-numbered values, but they are completely different if we treat them as discrete labels. (2) CRFs are not optimal for sequence translation tasks and cannot easily handle the long-term dependencies of words and emphasis levels. (3) Since the entire translation pipeline is fairly complex and slow, all downstream components have to wait for the complete result of the upstream components, delaying full-sentence translations (Fig. 1). Moreover, all of the components are trained separately and glued together to perform decoding. Joint optimization cannot be directly applied because of the complexity of the translation pipeline.

In this paper, we make two contributions: (a) we propose an approach that can handle continuous emphasis levels based on sequence-to-sequence (seq-to-seq) models (Fig. 2). The objective function is the mean square error that directly takes into account the "amount of difference" of the emphasis levels. (b) We combine both machine and emphasis translations into a single joint translation model (Fig. 3) based on seq-to-seq attention models that achieved the state-of-the-art performance in machine translation tasks [11,12]. As the result, the translation pipeline is greatly simplified and we can perform joint optimization. We can also avoid one-to-one word alignments, which are a required component of previous works. This mechanism not only reduces the complexity but also speeds up the decoding process[1].

II. CRF-based Emphasis Speech-to-Speech Translation

In this section, we describe the basic components of an emphasis speech translation (E-S2ST) system and the most recent state-of-the-art system based on CRF [10].

As described in the previous section, there are many works on emphasis translation. Although they utilized different approaches, the translation pipeline is the same as illustrated in Fig. 1. It combines 2 main sub-systems: an S2ST system that translates the linguistic meaning of speech with ASR, MT, and TTS modules; and an emphasis translation system that estimates (ES) and translates the emphasis information (ET).

A. Automatic Speech Recognition

The ASR component, which is the first element of the S2ST translation pipeline, transcribes input speech signal $\mathbf{x}$ into corresponding word sequence $\mathbf{w}$. Input $\mathbf{x}$ is decomposed into a sequence of feature frames $\mathbf{o}$ that only retain relevant information for the ASR task. Word sequence $\mathbf{w}$ is then predicted to maximize the conditional probability,

$$\hat{\mathbf{w}} = \operatorname*{argmax}_{\mathbf{w}} P(\mathbf{w}|\mathbf{o}). \tag{1}$$

For high accuracy, such speaker-dependent features as emotion and emphasis are either removed or normalized. As a result, emphasis information is lost after the ASR component.

B. Statistical Machine Translation

The MT component translates a word sequence from a source language into the target language. Many methods can be applied to the MT task, including phrase-based [15], tree-based [16], and neural network [11] translation models.

In the S2ST system, the MT component is downstream of the ASR component that takes ASR output $\mathbf{w}^{(s)}$ and finds the highest probability target language sentence $\mathbf{w}^{(t)}$:

$$\hat{\mathbf{w}}^{(t)} = \operatorname*{argmax}_{\mathbf{w}^{(t)}} P(\mathbf{w}^{(t)}|\mathbf{w}^{(s)}). \tag{2}$$

Similar to the ASR component limitation, the MT is only optimized to translate linguistic information and cannot handle emphasis information.

C. Text-to-speech Synthesis

Text-to-speech, which is the last component in the S2ST system, synthesizes the target audio given the translated text from the MT component. Many approaches are also used in TTS such as DNN-based or HMM-based. While DNN-based approaches are recently getting more attention because of their good synthetic speech, HMM-based approaches provide much more flexibility for handling emotion or emphasis information.

The general idea of the HMM-based TTS approach is that output speech parameter vector sequence $\boldsymbol{v}$ is determined by maximizing the likelihood function where the state sequence consists of T states $\boldsymbol{q} = [q_1, \cdots, q_T]$ and HMM model set $\mathbf{M}$

$$\hat{\boldsymbol{v}} = \operatorname*{argmax}_{\boldsymbol{v}} P(\boldsymbol{W}\boldsymbol{v}|\boldsymbol{q}, \mathbf{M}), \tag{3}$$

where $\boldsymbol{W}$ is the weighting matrix for calculating the dynamic features [17].

Unlike ASR and MT, the TTS component can be optimized for synthesizing emphasis speech [1]. However, because the MT component's output only contains text, the output sound cannot be emphasized in a way that reflects the original emphasis of the source language.

D. Emphasis Estimation

To address ASR's limitation, the ES component was proposed to estimate emphasis information using emphasis features including power, F_0, and duration. Do et al. [10] proposed an approach based on a linear-regression hidden semi-Markov model (LR-HSMM) [18] that models emphasis at

[1] Parts of this work were previously presented [13,14]. The current work provides a more comprehensive and systematic description of our method and a deeper analysis of our experiments.

the word-level (emphasis weight). The emphasis weight is a real-numbered value that represents the intensity of a word's emphasis. As illustrated in Fig. 1, since the ES system take emphasis features as inputs and the ASR outputs word sequences, each word from the ASR output has a corresponding emphasis weight number.

The intuition of the emphasis weights is shown in Fig. 4. Given a word sequence, 2 HSMM state sequences, normal and emphasis, can be derived, both of which are respectively trained from normal and emphasized speech. Emphasis weights interpolate the mean component of these 2 HSMM sequences to construct an LR-HSMM sequence. Note that the emphasis weights are shared among all HSMM states that belong to one word.

In the model training stage, only normal and emphasis HSMM parameters are optimized. Inn the emphasis estimation stage, emphasis weights $\boldsymbol{\lambda}$ are optimized using the EM algorithm to maximize the HSMM likelihood as follows:

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}}\, P(\mathbf{o}|\boldsymbol{\lambda}, \mathcal{M}), \tag{4}$$

where $\mathbf{o}$ is an observation speech feature sequence and $\mathcal{M}$ is the LR-HSMM parameters.

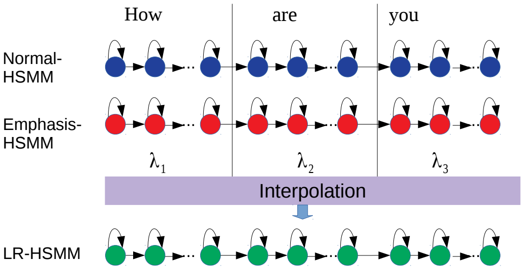

Fig. 4. Continuous emphasis modeling with linear-regression HSMM.

E. Emphasis Translation (ET)

The ET component plays a similar role as the MT component. The difference is that instead of translating a sequence of words, it translates a sequence of estimated emphasis weights described in II-D. As illustrated in Fig. 1, the ET component takes 2 inputs, estimated source-language emphasis sequence $\boldsymbol{\lambda}^{(s)}$ and translated word sequence $\mathbf{w}^{(t)}$, and predicts target-language emphasis sequence $\boldsymbol{\lambda}^{(t)}$.

Do et al. [10] proposed an emphasis translation approach based on conditional random fields (CRFs) [19] that treats emphasis as discrete labels[2] $\boldsymbol{\lambda}^{(s)'}$ and $\boldsymbol{\lambda}^{(t)'}$. Target emphasis weights $\boldsymbol{\lambda}^{(t)'}$ are then modeled by the following likelihood function:

$$P(\boldsymbol{\lambda}^{(t)'}|\mathbf{x}) = \frac{\displaystyle\prod_{n=1}^{N} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(\lambda_{n-1}^{(t)'}, \lambda_n^{(t)'}, \mathbf{x}_n^{(k)})\right\}}{\displaystyle\sum_{\tilde{\boldsymbol{\lambda}}^{(t)'}} \prod_{n=1}^{N} \exp\left\{\sum_{k=1}^{K} \theta_k f_k(\tilde{\lambda}_{n-1}^{(t)'}, \tilde{\lambda}_n^{(t)'}, \mathbf{x}_n^{(k)})\right\}}, \tag{5}$$

where N is the number of training samples, K is the number of feature functions f, and θ_k is the weight parameter of feature function k-th. Feature function f_k combines word-level emphasis bi-gram $\lambda_{n-1}^{(t)'}, \lambda_n^{(t)'}$, and input feature $\mathbf{x}_n^{(k)}$.

III. Limitations

A. CRF-based Emphasis Translation

Even though a CRF-based ET can preserve emphasis, its major problem is that it must quantize continuous emphasis levels into discrete labels. Although this mechanism increases the ratio of the number of labels and their training samples, the translation model is prone to make very bad predictions. For instance, instead of predicting 0.9, it might predict 0.1. Since those values are treated as separate discrete labels, it cannot capture the difference between 0.9 and 0.1.

Another problem with the CRF-based approach is that although it model local dependencies well (by adding more feature functions), it has difficulty handling long-term dependencies. One can use many feature functions to handle this problem, but as they increase, more data are required. And since emphasis translation requires parallel emphasized speech, which is very hard to collect, this approach is not practical.

B. Complex Translation Pipeline

To translate emphasis, the translation pipeline requires ES and ET components in addition to the S2ST system, which is now very complex: 5 components, and 6 internal dependencies (represents by black arrows) (Fig. 1). All the downstream components have to wait for the upstream outputs, resulting in large translation delays. Moreover, each component uses very different techniques, complicating joint training and decoding.

IV. Sequence-to-Sequence Approaches for Emphasis Translation

In this section, we describe our proposed approaches to tackle the above limitations. We first propose a hard-attentional seq-to-seq long short-term memory (LSTM) model that does not require emphasis quantization while retaining the ablility to capture the complex dependencies of emphasis and other linguistic information such as words and part-of-speech (PoS) tags. Then we propose a new emphasis speech translation pipeline that combines MT and ET into a new single model, which eliminates complex dependencies and makes the entire translation faster.

We chose the seq-to-seq based approach because it has achieved impressive results for many tasks, such as speech recognition [20,21] and machine translation (MT) [11]. Particularly, attentional-based seq-to-seq [22,23] achieved state-of-the-art performances for MT and ASR tasks and can model long-term dependencies, overcoming the problems of local dependencies in CRFs. In addition, models can be defined that can simultaneously handle both continuous and discrete variables, as well as cost functions that take into account label distances, for example, mean squared errors.

Moreover, since the state-of-the-art MT is based on seq-to-seq approaches and ET also resembles a translation task,

[2] Estimated continuous emphasis weights are quantized into sets of buckets (labels)

better performance can probably be achieved utilizing seq-to-seq for ET tasks. In addition, integrating MT and ET tasks will become easier because similar techniques are applied for both tasks.

A. Sequence-to-sequence LSTM Models

LSTM [24] is a special kind of recurrent neural network model that can capture long-term dependencies by special units called *memory blocks* and also manages the information going through it using forget, input, and output gates. Given input vector $\mathbf{x}_t$ at time t and hidden vector $\mathbf{h}_{t-1}$ and cell state $\mathbf{C}_{t-1}$ at time $t-1$, the information flow can be described:

- Calculate forget gate $\mathbf{f}_t$:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \times [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f). \tag{6}$$

- Calculate input gate $\mathbf{i}_t$ and estimate cell state $\widetilde{\mathbf{C}}_t$:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \times [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), && (7) \\ \widetilde{\mathbf{C}}_t &= tanh(\mathbf{W}_C \times [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C), && (8) \end{aligned}$$

- Update cell state $\mathbf{C}_t$:

$$\mathbf{C}_t = \mathbf{f}_t \times \mathbf{C}_{t-1} + \mathbf{i}_t \times \widetilde{\mathbf{C}}_t, \tag{9}$$

- Calculate output vector $\mathbf{h}_t$:

$$\begin{aligned} \mathbf{v}_t &= \sigma(\mathbf{W}_v \times [\mathbf{h}_{t-1,\mathbf{x}_t}] + \mathbf{b}_v), && (10) \\ \mathbf{h}_t &= \mathbf{v}_t \times tanh(\mathbf{C}_t). && (11) \end{aligned}$$

Here $\mathbf{W}$ and $\mathbf{b}$ are the matrix and bias vectors of the neural network layers. The core component of an LSTM is cell state $\mathbf{C}_t$ (Eq. (9)), which is controlled by forget gate $\mathbf{f}_t$ that is multiplied by the previous cell state values to decide which history information it should forget, and input gate $\mathbf{i}_t$, which is multiplied to the estimated cell state to decide which information is sent to the cell state.

An attention seq-to-seq LSTM model consists of an LSTM encoder, which encodes the input information, an LSTM decoder, which takes the encoded output to make a prediction, and an attention layer, which calculates an attention vector. The seq-to-seq translation model can be written as follows:

- Encode the input features to obtain hidden states $\mathbf{h}^{(s)}$:

$$\mathbf{h}_i^{(s)} = enc(\mathbf{h}_{i-1}^{(s)}, \mathbf{x}). \tag{12}$$

- Compute the attention vector $\mathbf{a}_j^{(t)}$ and context vector $\mathbf{c}_j$:

$$\mathbf{a}_j^{(t)} = att(\mathbf{h}_j^{(t)}, \mathbf{h}_i^{(s)}), \tag{13}$$

  where j is the prediction time step, and $\mathbf{h}_j^{(t)}$ is the decoder hidden state. Given $\mathbf{a}_j^{(t)}$ as weights, context vector $\mathbf{c}_j$ is computed as the weighted average over all source hidden states $\mathbf{h}^{(s)}$.
- Predicts target labels y_j,

$$\begin{aligned} P(y_j | y_{<j}, \mathbf{x}) &= softmax(\mathbf{W}_t \widetilde{\mathbf{h}}_j^{(t)}), && (14) \\ \widetilde{\mathbf{h}}_j^{(t)} &= tanh(\mathbf{W}_c [\mathbf{c}_j ; \mathbf{h}_j^{(t)}]). && (15) \end{aligned}$$

B. Hard-attentional Seq-to-seq Emphasis Translation

Our proposed hard-attentional model for emphasis translation is a modified version of the seq-to-seq model described in the above section, based on an assumption that we have a target language word sequence that was predicted from an external MT model and word alignments from an external word alignment model.

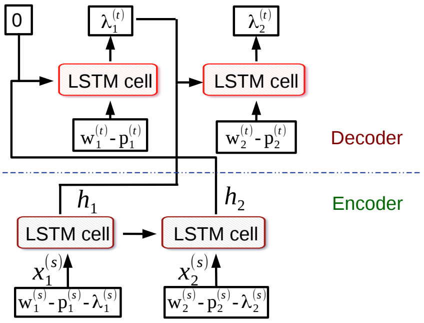

Fig. 5. Unfolded hard-attentional encoder-decoder LSTM model for translating emphasis sequence $\boldsymbol{\lambda}^{(e)}$ into target output sequence $\boldsymbol{o}^{(f)}$. It considers many linguistic features including word sequence $\mathbf{w}_i^{(e,f)}$ and part of speech sequence $\mathrm{p}_i^{(e,f)}$ from both source and target languages.

The entire encoder-decoder process can be written as a function of input features:

$$\lambda^{(t)} = f(\mathbf{x}^{(s)}), \tag{16}$$

where $\lambda^{(t)}$ is the target output emphasis sequence, $\mathbf{x}^{(s)}$ is the sequence of the source-language input features including words $\mathbf{w}^{(s)}$, PoS $\boldsymbol{p}^{(s)}$, and emphasis weights $\boldsymbol{\lambda}^{(s)}$. A previous work [10] reported that both words and PoS tags play a crucial role in good translation models.

1) The encoder: As illustrated in Fig. 5, the encoder is a standard LSTM model that takes input vector $\mathbf{x}_i^{(e)}$, which consists of words ($w_i^{(e)}$), part-of-speech tags ($\mathrm{p}_i^{(e)}$), and emphasis levels ($\lambda_i^{(e)}$), and encodes them into a single vector that is suitable for predicting emphasis levels.

The input PoS tags are converted into one-hot vectors whose size equals the PoS vocabulary size. Word embeddings [25] are applied to map words onto vectors that capture the similarity between words. All these input features are concatenated into a single vector and fed to the encoder.

The encoder is pre-trained by appending a linear neural-net layer on top of it with an output size of 1 to predict the emphasis level that is fed into the input layer, similar to an auto-encoder model [26] (Fig. 6 (a)). We want output hidden layer $\mathbf{h}$ to represent the features that are the most useful to predict the emphasis levels (called emphasis representations).

2) The decoder: The decoder is also a standard LSTM model, and the input layer contains both linguistic information (words, PoS) and vector representations calculated by the encoder, based on a novel hard-attentional model.

The name hard-attentional reflects how the decoder calculates the emphasis representation vectors used as input. The example in Fig. 5 demonstrates this mechanism. Assume that word pairs $w_1^{(t)}$-$w_2^{(s)}$ and $w_3^{(t)}$-$w_1^{(s)}$ are aligned based on word alignments. To generate output $\lambda_2^{(t)}$, linguistic features

$w_2^{(t)}$ and $\mathrm{p}_2^{(t)}$, and previous output $\lambda_1^{(t)}$, the decoder takes encoded $\mathbf{h}_1$ from the encoder output, because word pair $w_1^{(s)}$-$w_2^{(t)}$ is aligned. For unaligned words, we use zero vectors as emphasis representation vectors.

We propose 2 decoders as follows:

- **LSTM_emph**: The model directly predicts target emphasis sequence $\boldsymbol{\lambda}^{(t)}$.
- **LSTM_diff**: The model's output is treated as the difference from the input emphasis level. The target emphasis level of the j-th word is calculated by, $\lambda_j^{(t)} = f(\mathbf{x}^{(s)}) + \lambda_i^{(s)}$, where the model gets "attention" from word $w_i^{(s)}$.

The intuition behind **LSTM_diff** is an intention to put stronger weight on the corresponding source-language emphasis when predicting target emphasis.

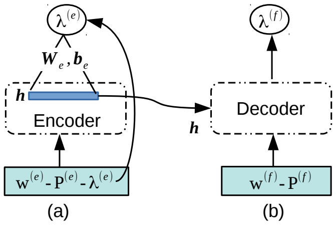
$\lambda^{(e)}$
$\boldsymbol{W}_e, \boldsymbol{b}_e$
$\boldsymbol{h}$
Encoder
$\boldsymbol{h}$
Decoder
$\lambda^{(f)}$
$\mathrm{w}^{(e)}$-$\mathrm{P}^{(e)}$-$\lambda^{(e)}$
$\mathrm{w}^{(f)}$-$\mathrm{P}^{(f)}$
(a)
(b)

Fig. 6. Training procedure for the hard-attentional model.

C. Joint Translation Model

Even though the hard-attention seq-to-seq model solves the problems of the CRF-based approach, which requires emphasis quantization and suffers from the long-term dependency problem described in Section III-A, the problem with a complex translation pipeline remains. In this section, we propose a joint translation framework based on an attentional NMT that simultaneously combines MT and ET to translate words and emphasis. Our approach is also based on seq-to-seq approaches, as is the hard-attention approach. The difference is that we do not require external MT or word alignment models anymore. All components are combined into a single joint translation model, allowing us to perform joint optimization and inference.

When integrating emphasis with word translation, the major difficulty is that the amount of text data usually overwhelms the amount of emphasis data, because the latter are derived from parallel emphasized speech that is much harder to collect than parallel text data, which can be massively collected by crawling websites [27].

We define the joint translation model as follows. Given a source language word and an emphasis sequence are denoted as $\boldsymbol{W}^{(s)}$ and $\boldsymbol{e}^{(s)}$, respectively. The model predicts one target word $w^{(t)}$ at a time followed by a prediction of its emphasis weight $e^{(t)}$. Next, we detail how the encoder and decoder handle both words and emphasis weights.

1) Encoder with emphasis weights: One way to embed emphasis weights into the encoder is to concatenate them with word representation to form input vector $[w_i^{(s)}, e_i^{(s)}]$ of the encoder (*Emp-Enc*) and compute the hidden unit:

$$\boldsymbol{h}_i^{(s)} = enc([w_i^{(s)}, e_i^{(s)}]). \tag{17}$$

By doing this, we ensure that the emphasis weights are also encoded with words. However, since the effect of emphasis on MT remains unknown, we need to explore alternative ways to incorporate emphasis into the encoder to analyze this effect. Therefore, we propose adding emphasis after encoding words (*SkipEnc*) as follows:

$$\boldsymbol{h}_i^{(s)} = [enc(w_i^{(s)}), e_i^{(s)}] \tag{18}$$

The *SkipEnc* idea is that if emphasis weights negatively affect machine translation, adding them after the encoder might weaken the effect.

2) Decoder with emphasis weights: As illustrated in Fig. 7, the decoder has two components. A word prediction layer follows the standard NMT, and emphasis prediction layer $\boldsymbol{W}_e$ that takes input is the combined vector of the predicted word and the decoder hidden activation as follows:

$$e_i^{(t)} = \boldsymbol{W}_e([\widetilde{\boldsymbol{h}}_i^{(t)}, w_i^{(t)}]). \tag{19}$$

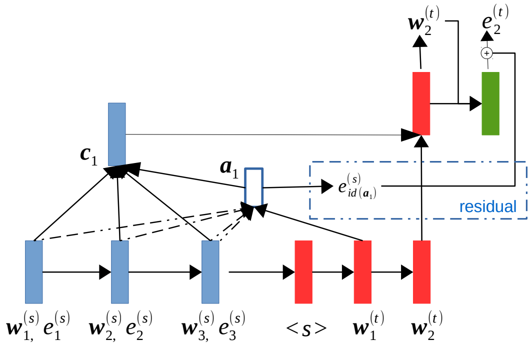
$\boldsymbol{w}_2^{(t)}$
$\boldsymbol{e}_2^{(t)}$
$\boldsymbol{c}_1$
$\boldsymbol{a}_1$
$e_{id(\boldsymbol{a}_1)}^{(s)}$
residual
$\boldsymbol{w}_{1,}^{(s)}\, e_1^{(s)}$
$\boldsymbol{w}_{2,}^{(s)}\, e_2^{(s)}$
$\boldsymbol{w}_{3,}^{(s)}\, e_3^{(s)}$
$<s>$
$\boldsymbol{w}_1^{(t)}$
$\boldsymbol{w}_2^{(t)}$

Fig. 7. Joint word-emphasis translation framework with word dependencies and residual connection.

However, as described above, the lack of emphasis data compared with the text data might saturate the effect of the source emphasis when going through many hidden layers. To overcome this problem, we utilize a residual connection in the way that the source emphasis weight is also used when predicting target emphasis weights (Fig. 7),

$$e_i^{(t)} = \boldsymbol{W}_e([\widetilde{\boldsymbol{h}}_i^{(t)}, w_i^{(t)}]) + e_{id(\boldsymbol{a}_i)}^{(s)}, \tag{20}$$

where function $id(\boldsymbol{a}_i)$ returns the index of the largest value of weighted vector $\boldsymbol{a}_i$ that indicates the source aligned word.

3) Training procedure: To train our model, we utilize two objective functions, cross entropy (CE) for word prediction and mean square error (MSE) for emphasis prediction, because the CE function greatly outperforms MSE with discrete labels, which is the case for word prediction. Since emphasis weights are continuous, the CE function cannot be utilized as the objective function for emphasis prediction.

The training algorithm is a standard back propagation through time (BPTT) scheme in which the errors from the machine and emphasis translations are sequentially back–propagated. Note that the errors are not joint because their scales are different.

V. Experiments

In our experiment, we first evaluated the effectiveness of using a continuous emphasis level in a translation model and the ability to handle the long-term dependencies of our proposed hard-attentional seq-to-seq model against the previous CRF-based approach (Section V-B).

Regarding the evaluation of the joint translation model, since our proposed model is our first attempt, to the best of our knowledge, for integrating emphasis and word features in a single model, analysis must be conducted on the effect of emphasis on the standard MT translation model (Section V-D). This critical step not only shows the effect of emphasis as a feature, but also provides vital cues for optimally combining ET and MT.

To further reduce the complexity of the input features and the network structure, we also evaluated the effect of PoS tags on the ET and MT models (Section V-E). Do et al. [10] argued that PoS tags are crucial features that can boost the ET performance by a 4% F-measure. However, it creates another dependency for the translation model. On the other hand, the seq-to-seq translation model is capable not only of learning how to translate but also learning the semantic meaning of words, and since the semantic meaning are closely related to syntactic meaning (PoS tags) [28], we expect that it can avoid the need of PoS tag features.

Finally, based on the analysis result, we conduct experiments with the joint translation model and compared both the hard-attention and CRF-based approaches (Section V-F).

A. Experimental setup

1) Corpus: The corpus consists of emphasis and machine translation data. The former contain 966 parallel English and Japanese utterances [29]. In each language, at least one of the content words in the sentence is emphasized, and the number of emphasized words is identical between languages. The number of speakers is 8, including 3 native English ($\mathrm{En}^{\{1,2,3\}}$) and 5 native Japanese ($\mathrm{Ja}^{\{1,2,3,4,5\}}$) speakers.

To create training and testing data for our emphasis translation evaluation, we divided 966 utterances of each speaker into 2 sets of 866 and 100 samples such that the same sentences are used for all speakers. We then paired the 866 utterances of each English speaker with those of all 5 Japanese speakers, resulting in 4330 ($866 * 5$) training, and 100 testing samples for each English speaker. The testing data consist of 157 emphasized words, in which 30 exist in the training data and 127 do not.

Regarding to the machine translation data, we utilized 2 sets, the BTEC and BTEC+TED corpora, which contain ~450k and ~670k parallel sentences, respectively. We created 2 training MT datasets to evaluate the effectiveness of the emphasis information on the MT task with more varieties of testing conditions.

2) Emphasis translation procedure & measurement: In this paper, to evaluate the performance of the emphasis translation in isolation, we assumed that the MT system produces 100% correct translation outputs. Word alignments

To measure the emphasis translation accuracy, we first performed emphasis translation to derive the target emphasis sequences and then measured its accuracy in the target language both objectively or subjectively (Fig. 8). In the objective evaluation, the target emphasis values are classified as "emphasized" or "not emphasized" using a threshold of 0.5[3] and compared them with true values. In the subjective evaluation, we first synthesized the audio from the translated emphasis sequences, and gave the output audio to 7 Japanese native listeners to predict the emphasized words[4]. In both evaluations, we calculated the F-measure, which ranged from 0 to 100 representing how accurately the system preserved emphasis in the target language.

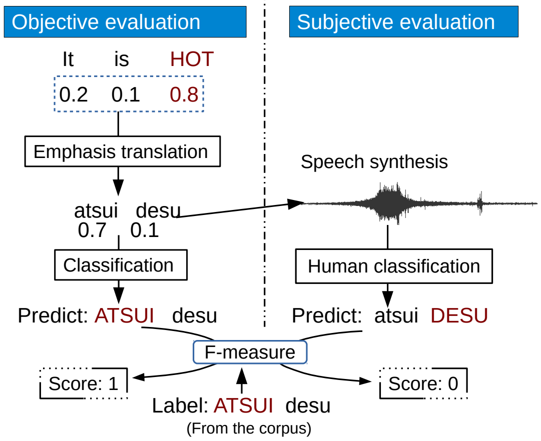

Fig. 8. Example of the emphasis translation procedure and measurement methods.

3) CRFs: We retained the identical configuration of the CRF models as in previous work [30]. The word-level emphasis was quantized to the closest $\{0, 0.3, 0.6, 0.9\}$. The input features are words, PoS tags, and PoS contexts in the target language side. The model directly predicts the target side emphasis sequence. This setting achieved the best performance compared to other features combinations.

4) Hard-attentional Seq-to-seq model:

- **The encoder:** The encoder input consists of words, PoS tags, and emphasis levels. The input layer has 138 dimensions including 100 word embeddings, 37 one-hot PoS tags, and the emphasis levels. The hidden layer has 100 dimensions.
- **The decoder:** The input gate consists of 100 word embedding dimensions and 17 one-hot PoS dimensions. The attentional vector taken from the encoder was added to the input gate's output. The input words and PoS are also respectively converted into word-embedding and one-hot vectors.

The word embeddings for both the encoder and decoder were pre-trained using the BTEC travel conversation corpus [31] using word2vec toolkit [25].

5) Joint translation model: Our encoder and decoder models have 1 layer (unless stated otherwise), 512 cells, and 512-dimensional word embeddings. We trained for a maximum

[3] This has been reported in the previous work [30] as having the best performance to classify emphasized and normal words.

[4] There is no constraint on how emphasized words are expressed, it is up to the listeners to make a binary decision on whether a word is emphasized.

of 20 epochs using the RMSprop algorithm [20]. Emphasis prediction layer $\boldsymbol{W}_e$ was frozen when trained with fake emphasis data to avoid learning from unrealistic emphasis weights.

When trained with text data, the learning rate was set to $1e-4$ and $5e-5$ when trained with emphasis data. We employed an early stop learning rate schedule and reduced the learning by a factor of 2 whenever loss increased on the development set and stopped the training when the learning rate fell below 1e-5. Our mini-batches were 128 and 10 for the word translation and the emphasis translation task, respectively. The batches were shuffled before every training epoch.

B. Hard-attentional models: objective evaluation

We performed a preliminary experiment using the same corpus as in a previous work [30] with 916 training samples and 50 testing samples. The results showed that our proposed method achieved a 92.6% F-measure, which exceeds the previous work by 1%. Although the dataset was too small to conclude that the proposed method is better than CRFs by such a small margin, it demonstrates that the proposed method performs comparably with the previous work on the same corpus. To make the result more reliable, we conducted larger scale experiments with the dataset described in the Section V-A1.

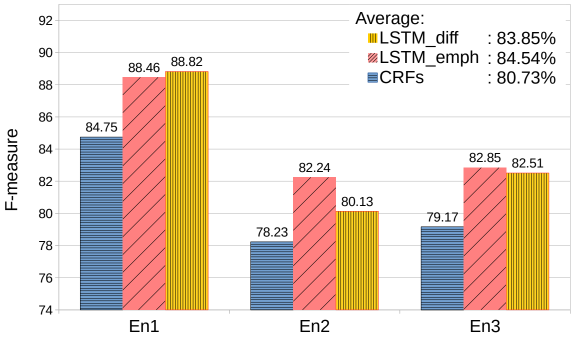

Fig. 9. Objective emphasis prediction of hard-attentional enc-dec with LSTM_diff and LSTM_emph architectures.

Fig. 9 shows the objective F-measure for emphasis prediction on this larger amount of data. In all 3 test sets and on average, the proposed methods outperformed the CRFs. According to the bootstrap resampling significance test [32], both results are significant at the $p < 0.01$ level. On the other hand, the difference between *LSTM_diff* and *LSTM_emph* was not significant, demonstrating that the LSTM model can learn emphasis level differences between aligned words without explicitly defining them in the equations.

Furthermore, we scrutinized the advantage of the proposed model with respect to using continuous variables. If they are useful, we expect that the emphasis values in the middle of the range will be modeled better by the proposed method. To test this hypothesis, we split the input emphasis levels into 3 sets based on the emphasis level of the word: < 0.3, 0.3-0.6, > 0.6. Then we calculated F-measure for the *CRFs* and *LSTM_emph* on individual sets[5]. The result in Table I indicates that both systems have equivalent performance when a word is considered normal or emphasized (emphasis levels below 0.3 or over 0.6), but when the emphasis levels fall between 0.3-0.6, *LSTM_emph* outperformed the *CRFs*. This demonstrates the limitation of the *CRFs*, which require emphasis level quantization to handle continuous variables, but LSTMs do not.

TABLE I
F-MEASURE FOR CRF AND LSTM_EMPH EMPHASIS TRANSLATION ON DIFFERENT INPUT EMPHASIS LEVELS.

<0.3		0.3-0.6		>0.6	
CRF	LSTM	CRF	LSTM	CRF	LSTM
88.05	87.69	70.85	81.41	92.53	92.75

C. Hard-attentional models: subjective evaluation

Finally, we performed a subjective evaluation to verify whether human listeners can perceive the same improvement between *CRFs* and *LSTM_emph* as in the objective evaluation. We used the "En1" test set for this evaluation.

We obtained a result of 83.0% for *LSTM_emph* and 81.0% for *CRFs* indicating that humans perceived a slightly smaller improvement compared to the objective result. Moreover, the *CRF* system's performance dropped with a smaller margin (3.70%) than the proposed method (5.82%). The reason is because in the *LSTM_emph* approach, 268 emphasized words were recognized correctly in the objective evaluation, but 14 of them having emphasis levels fall between 0.5-0.8 are mis-recognized by human listeners while this does not happen in the *CRF* approach since these emphasis levels are just slightly higher than the threshold, leading to slightly emphasized synthetic speech that is hard to perceive by human listeners. In the *CRF* approach, the emphasis levels are quantized into buckets of $\{0, 0.3, 0.6, 0.9, \dots\}$, so when a word is considered as emphasized (larger than the threshold 0.5), the distance to the threshold is usually large.

D. Effect of using emphasis as additional features on standard NMT systems

Even though previous works translated emphasis weights separately from NMT, no analysis has addressed whether emphasis weights in NMT have a positive or negative effect. Such analysis, however, is important before integrating emphasis translation into NMTs. To address this oversight, we explored the effect of emphasis as an input feature on machine translation performance.

We kept the same decoder structure like standard NMT systems so that no emphasis prediction was performed and evaluated two encoders with emphasis weights added in different positions as described in Section IV-C1. The baseline is the standard NMT system without emphasis weights (*Std. NMT*). Fig. 10 shows the result of the cross entropy loss of the word prediction performance on the training and development

[5] Because the accuracies of *LSTM_diff* and *LSTM_emph* are similar, below we only show the results of *CRFs* and *LSTM_emph*.

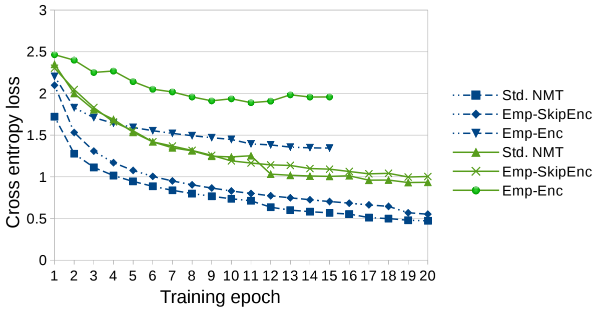

Fig. 10. Effect of emphasis on standard NMT systems: Solid and dash lines denote MT performance on development and the training sets, respectively.

sets. The loss is higher in both approaches (*SkipEnc* and *Emp-Enc*) than in *Std. NMT*, indicating that emphasis did not improve the NMT performance. We hypothesize that such a negative effect is due to the fact that emphasis weights are paralinguistic, but NMT is translating linguistic information. Only using emphasis weights as an additional feature without translating is insufficient for the model to learn anything useful from emphasis.

Although the NMT performance was degraded when using the emphasis weight features, *SkipEnc* has a minimal effect compared with *Emp-Enc*. This is because in *SkipEnc*, the encoder avoids excessive influence from the negative effect of the faked emphasis weights; therefore, we can preserve the performance of the standard NMT. The rest of our experiments used the *SkipEnc* model.

E. Joint translation models: Effect of PoS tags on ET and MT models

Figure 11 shows the performance of the ET model using the *EmpEnc* joint translation approach with and without the PoS tag feature. With PoS tags, the model converges faster and provides better performance in the first 10 iterations. But both systems eventually converge to a similar point when we train them for 15 iterations. We also observed the same tendency in the MT task (Fig. 12). The result indicates that PoS tags still help the translation model, but if we train it on a sufficient amount of iterations, such help is minimalized. We hypothesize that this is because the model can learn semantic meaning of a word that is similar to what the PoS tags represent.

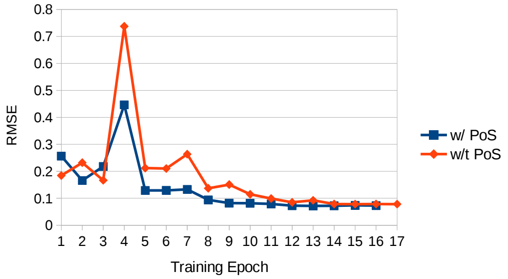

Fig. 11. ET performance in joint translation models on a development set with/without PoS tags.

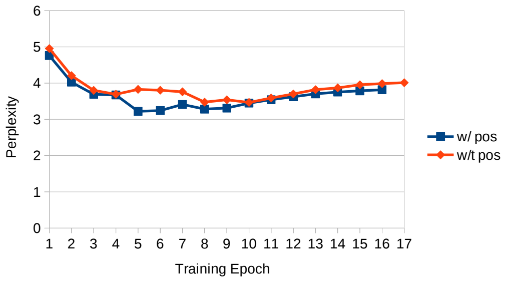

Fig. 12. MT performance in joint translation models with/without PoS tags.

F. Joint translation models: Emphasis translation performance

From the result of the above sections, we conducted the following experiments using *SkipEnc* architecture without PoS tag features and completely trained the model for both emphasis and word prediction. Fig. 13 shows the F-measure, the precision, and the recall for emphasis prediction using the *SkipEnc* encoder with *baseline* and *residual* decoders.

Looking at the F-measure, the *residual* decoder outperformed the *baseline* decoder by a 2.7% F-measure. The *baseline* decoder's precision, however, is higher than the *residual* one, indicating that the *residual* connection mistakenly predicts more high emphasis weights for normal words. Similarly, the high score for the *residual* decoder's recall indicates that it preserves more emphasized words than the *baseline* system.

The contrastive precision and recall performance of the two systems indicates that better performance is gained by combining them. In the next section, we describe our combination technique and compare its result with previous works.

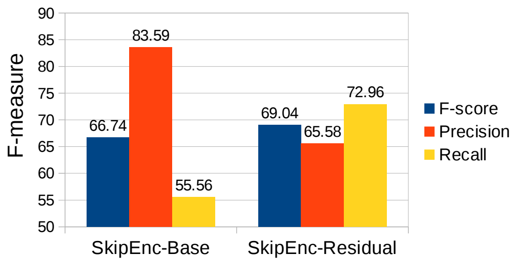

Fig. 13. Emphasis translation performance in joint translation model

G. Joint translation models: model combination for emphasis translation

The model combination works as follows. First, we performed emphasis translation on the development set and calculated the precision and recall scores. Then, for content words, we selected the emphasis weights predicted from the system with higher recall, and for the non-content words, we selected emphasis weights with lower recall.

We also performed emphasis translation using previous approaches based on conditional random fields (CRFs) [30] and LSTM hard-attention models [13]. The input features for these approaches are words and emphasis weights that resemble the proposed approach. The result is shown in Fig. 14. Compared with CRFs, our proposed approaches perform bettered with a ~5% F-measure and have a closed performance with the LSTM hard-attention approach with a ~2% lower F-measure.

The result matches our expectation because both the CRFs and LSTM hard-attention approaches use ground-truth one-to-one word alignments and have independent words and emphasis translation models. On the other hand, our proposed approaches do not require word alignment models and can translate words and emphasis twice as fast as hard-attention models.

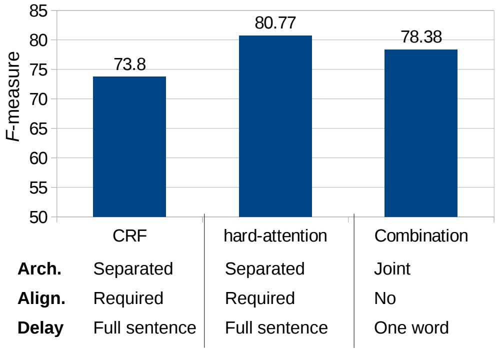

Fig. 14. Comparison of emphasis translation performance of proposed and previous approaches. Graph also shows differences in terms of translation architecture (Arch.), word alignment requirement (Align.), and the translation delay (Delay).

H. Joint translation models: machine translation performance

Next, we evaluated the machine translation performance in joint translation models with 2 dataset, BTEC and BTEC+TED. We are interested to see how the systems behave when the number of parameters (model's depth) increase. The results are shown in Table II. The baseline system is the standard NMT without the emphasis weights used in both the encoder and decoder.

First, we can see that the performance on the BTEC + TED set is lower than the BTEC set. It is because the TED data covers a much more open domain, therefore harder to translate, than the BTEC (only travel) domain. Second, when the model's depth is 1 and 2, the performance difference of the proposed approaches and the baseline is negligible for both test datasets, indicating that optimizing the model with emphasis weights can compensate for the negative effect of emphasis found in Section V-D.

With a hidden layer depth of 3, all of the models seem over-fitted with the training samples, resulting in a loss of performance. However, interestingly, the proposed approaches have smaller performance drops. Specifically, the *SkipEnc-Residual* approach only dropped ~1-2% of BLEU, and the baseline system without emphasis weights dropped ~3-4% of BLEU. We hypothesize that emphasis weights work as regulation parameters that help preventing over-fitting.

VI. Conclusions

In this paper, we proposed methods to accurately translate emphasis, and reduce translation complexity. Unlike previous work where emphasis is considered to be discrete labels and has difficulty handling long-term dependencies, our proposed hard-attention seq-to-seq model can solve both problems in a single model by utilizing the LSTM-based encoder-decoder that can capture long-term dependencies and handle continuous emphasis in its objective function. The evaluation on emphasis translation task demonstrates that our model can translate emphasis significantly better than previous work.

TABLE II
Machine translation performance (BLEU score) in joint translation models. Various depths of hidden layers denoted as *d(1,2,3)* were evaluated.

System	BTEC	BTEC + TED
Baseline (d1)	27.67	24.20
SkipEnc-Base (d1)	27.25	23.16
SkipEnc-Residual (d1)	27.19	-
Baseline (d2)	27.44	24.96
SkipEnc-Base (d2)	27.70	25.43
SkipEnc-Residual (d2)	27.72	25.14
Baseline (d3)	23.68	21.44
SkipEnc-Base (d3)	**25.41**	**22.24**
SkipEnc-Residual (d3)	**26.36**	**23.22**

With regards to effect emphasis and PoS tags on a machine translation task, we discovered that emphasis does not help standard MT systems if it is simply used as an additional feature. Experiments with PoS tags also showed that it helps the model converge faster, but it does not help improve the accuracy if the model is well-trained. Another important outcome of our result is that our proposed model can learn good features from words and emphasis without PoS tag dependencies.

Our work on the joint translation of words and emphasis demonstrated that our proposed joint translation model can accurately translate emphasis and words with one-word delay, but the previous work requires a full-sentence delay. The model significantly reduced the complexity by removing word alignments and PoS tag features. We also found that emphasis can help MT performance prevent over-fitting.

However, some limitations remain. First, although the complexity is already reduced, we still require an emphasis estimation (ES) that works independently with the ASR component. Although this architecture allows us to adopt any ASR technique without interfering with the emphasis estimation component, it creates a delay during which ET and MT models have to wait for the ASR and ES output.

Future work will integrate ES and ASR to completely remove any dependencies introduced by adding emphasis translation to standard S2ST systems. Joint training the whole system is another very interesting topic. Thanks to the seq-to-seq model, we can apply it to all components to seamlessly integrate them into a joint translation model. In addition, recent works on speech recognition have shown that TDNN is comparable (or even better in certain cases) with LSTM. Integrating these models into emphasis estimation and translation will further reduce the model complexity and potentially speed up the translation pipeline.

VII. Acknowledgements

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

References

[1] K. Yu, F. Mairesse, and S. Young, "Word-level emphasis modelling in HMM-based speech synthesis," in *Processing of ICASSP*, March 2010, pp. 4238–4241.

[2] A. Tsiartas, P. G. Georgiou, and S. S. Narayanan, "A study on the effect of prosodic emphasis transfer on overall speech translation quality," in *Proceedings of ICASSP*, 2013.

[3] S. Nakamura, "Overcoming the language barrier with speech translation technology," *Science & Technology Trends - Quarterly Review No.31*, April 2009.

[4] E. Fudge, *English Word-stress.* Routledge, 2015.

[5] T. Kano, S. Sakti, S. Takamichi, G. Neubig, T. Toda, and S. Nakamura, "A method for translation of paralinguistic information," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2012.

[6] T. Kano, S. Takamichi, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Generalizing continuous-space translation of paralinguistic information." in *Proceedings of INTERSPEECH*, 2013, pp. 2614–2618.

[7] G. K. Anumanchipalli, L. C. Oliveira, and A. W. Black, "Intent transfer in speech-to-speech machine translation," in *Proceedings of SLT*, Dec 2012, pp. 153–158.

[8] P. D. Aguero, J. Adell, and A. Bonafonte, "Prosody generation for speech-to-speech translation," in *Proceedings of ICASSP*, vol. 1, 2006.

[9] A. Tsiartas, P. Georgiou, and S. S. Narayanan, "Toward transfer of acoustic cues of emphasis across languages," in *Proceedings of InterSpeech*, 2013.

[10] Q. T. Do, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Preserving word-level emphasis in speech-to-speech translation," *IEEE*, vol. 25, no. 3, pp. 544–556, March 2017.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of NIPS*, 2014, pp. 3104–3112.

[12] M. T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *Proceedings of ICLR*, 2016.

[13] Q. T. Do, S. Sakti, G. Neubig, and S. Nakamura, "Transferring emphasis in speech translation using hard-attentional neural network models," in *Proceedings of Interspeech*, September 2016.

[14] Q. T. Do, S. Sakti, and S. Nakamura, "Toward expressive speech translation: A unified sequence-to-sequence LSTMs approach for translating words and emphasis," in *Proceedings of Interspeech*, 2017.

[15] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL*, 2003, pp. 48–54.

[16] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *ACL*, 2006, pp. 609–616.

[17] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of ICASSP*, vol. 3, 2000, pp. 1315–1318 vol.3.

[18] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, "A style control technique for HMM-based expressive speech synthesis," *IEICE*, vol. E90-D, no. 9, pp. 1406–1413, Sep. 2007.

[19] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, 2001, pp. 282–289.

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

[21] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of ICASSP*. IEEE, 2013, pp. 6645–6649.

[22] M. T. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[23] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, 2015.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[26] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of ICML*, 2008, pp. 1096–1103.

[27] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of LREC*, 2012.

[28] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *CoRR*, vol. abs/1703.04826, 2017.

[29] D. Q. Truong, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Collection and analysis of a Japanese-English emphasized speech corpus," in *Proceedings of Oriental COCOSDA*, September 2014.

[30] Q. T. Do, S. Takamichi, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Preserving word-level emphasis in speech-to-speech translation using linear regression HSMMs," in *INTERSPEECH*, 2015.

[31] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *Proceedings of EUROSPEECH*, 2003, pp. 381–384.

[32] P. Koehn, "Statistical significance tests for machine translation evaluation." in *Proceedings of EMNLP*, 2004, pp. 388–395.

Quoc Truong Do received his B.E. from University of Engineering and Technology, Hanoi, Vietnam, in 2013, and his M.S. from the Graduate School of Information Science, NAIST, Nara, Japan in 2015. He is currently in the doctoral course at NAIST, Japan. He interested in speech and natural language processing, with a focus on speech recognition, and speech translation. He is a student member of ISCA, and ASJ.

Sakriani Sakti received her B.E. degree in Informatics (cum laude) from Bandung Institute of Technology, Indonesia, in 1999. In 2000, she received DAAD-Siemens Program Asia 21st Century Award to study in Communication Technology, University of Ulm, Germany, and received her MSc degree in 2002. During her thesis work, she worked with Speech Understanding Department, DaimlerChrysler Research Center, Ulm, Germany. Between 2003-2009, she worked as a researcher at ATR SLC Labs, Japan, and during 2006-2011, she worked as an expert researcher at NICT SLC Groups, Japan. While working with ATR-NICT, Japan, she continued her study (2005-2008) with Dialog Systems Group University of Ulm, Germany, and received her PhD degree in 2008. She actively involved in collaboration activities such as Asian Pacific Telecommunity Project (2003-2007), A-STAR and U-STAR (2006-2011). In 2009-2011, she served as a visiting professor of Computer Science Department, University of Indonesia (UI), Indonesia. From 2011, she has been an assistant professor at the Augmented Human Communication Laboratory, NAIST, Japan. She served also as a visiting scientific researcher of INRIA Paris-Rocquencourt, France, in 2015-2016, under "JSPS Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation". She is a member of JNS, SFN, ASJ, ISCA, IEICE and IEEE. She is now also a board member of SLTU (Spoken Language Technologies for Under-resourced languages) and a committee member of SIG ELRA-LRL (Low Resourced Languages). Her research interests include statistical pattern recognition, speech recognition, spoken language translation, cognitive communication, and graphical modeling framework.

Satoshi Nakamura is Professor of Graduate School of Information Science, Nara Institute of Science and Technology, Japan, Honorarprofessor of Karlsruhe Institute of Technology, Germany, and ATR Fellow. He received his B.S. from Kyoto Institute of Technology in 1981 and Ph.D. from Kyoto University in 1992. He was Associate Professor of Graduate School of Information Science at Nara Institute of Science and Technology in 1994-2000. He was Director of ATR Spoken Language Communication Research Laboratories in 2000-2008 and Vice president of ATR in 2007-2008. He was Director General of Keihanna Research Laboratories and the Executive Director of Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan in 2009-2010. He is currently Director of Augmented Human Communication laboratory and a full professor of Graduate School of Information Science at Nara Institute of Science and Technology. He is interested in modeling and systems of speech-to-speech translation and speech recognition. He is one of the leaders of speech-to-speech translation research and has been serving for various speech-to-speech translation research projects in the world including C-STAR, IWSLT and A-STAR. He received Yamashita Research Award, Kiyasu Award from the Information Processing Society of Japan, Telecom System Award, AAMT Nagao Award, Docomo Mobile Science Award in 2007, ASJ Award for Distinguished Achievements in Acoustics. He received the Commendation for Science and Technology by the Minister of Education, Science and Technology, and the Commendation for Science and Technology by the Minister of Internal Affair and Communications. He also received LREC Antonio Zampoli Award 2012. He has been Elected Board Member of International Speech Communication Association, ISCA, since June 2011, IEEE Signal Processing Magazine Editorial Board Member since April 2012, IEEE SPS Speech and Language Technical Committee Member since 2013, and IEEE Fellow since 2016.