Generalizing Continuous-space Translation of Paralinguistic Information

Takatomo Kano, Shinnosuke Takamichi, Sakriani Sakti, Graham Neubig, Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science, Nara Institute of Science and Technology

{takatomo-k,shinnosuke-t,ssakti,neubig,tomoki,s-nakamura}@is.naist.jp

Abstract

In previous work, we proposed a model for speech-to-speech translation that is sensitive to paralinguistic information such as duration and power of spoken words [1]. This model uses linear regression to map source acoustic features to target acoustic features directly and in continuous space. However, while the model is effective, it faces scalability issues as a single model must be trained for every word, which makes it difficult to generalize to words for which we do not have parallel speech. In this work we first demonstrate that simply training a linear regression model on all words is not sufficient to express paralinguistic translation. We next describe a neural network model that has sufficient expressive power to perform paralinguistic translation with a single model. We evaluate the proposed method on a digit translation task and show that we achieve similar results with a single neural network-based model as previous work did using word-dependent models.

Index Terms: speech translation, paralinguistic information, linear regression, neural network

1. Introduction

Speech-to-speech translation (S2ST) consists of automatic speech recognition (ASR), machine translation (MT), and text to speech (TTS) components. The most commonly used speech translation model uses words as the basic unit for information sharing between these three components, but there are several major limitations of this approach. For example, in human communication, speakers use many different varieties of information to convey their thoughts and emotions. This paralinguistic information is not a factor in written communication, but in spoken communication it has great importance. However, this information is not translated in standard speech translation systems. For example, even if the input of ASR contains rich prosody information conveying emphasis or emotion, the words output by TTS will be given the canonical prosody for the input text, not reflecting these traits.

In previous work we proposed a method that overcomes this problem by translating paralinguistic information [1]. In this method we constructed a regression matrix for each word in the vocabulary to map its acoustic features. However, in this framework it is difficult to model for large vocabulary tasks, as it is necessary to construct separate models that represent acoustic feature mappings for individual word. This is because simple linear regression lacks the expressive power to map multiple words in a single model. In this paper we demonstrate that, as expected, making a single regression matrix mapping all words' acoustic features barely exceeds a baseline of not translating paralinguistic information at all.

In this work we expand the paralinguistic translation model to adapt to more general tasks by training a single model that is applicable to all words using neural networks. There are two merits to using neural networks. First, neural network possess sufficient power to express difficult regression problems such as translation of acoustic features for multiple words. Second, neural network can be expanded with features expressing additional information such as the input word and translated word, the position of both words, parts of speech, and so on.

We evaluate the proposed method by using parallel emphasized utterances to train and test our paralinguistic translation model. We measure the emphasis recognition rate and intensity by objective and subjective assessment, and find that the proposed generalized paralinguistic translation method is just as effective in translating this paralinguistic information as the previous word-dependent approach.

2. Related Work

There have been several studies demonstrating improved speech translation performance by utilizing paralinguistic information. [2] focus on the input speech's prosody, extracting F0 from the source speech at the sentence level and clustering accent groups. These are then translated into target side accent groups. [3] consider the prosody encoded as factors in a factored translation model [4] to convey prosody from source to target.

In our previous work [1], we also focus on source speech paralinguistic features, but unlike previous work we extract them and translate to target paralinguistic features directly and in continuous space. This allows for relatively simple, language-independent implementation and is more appropriate for continuous features such as duration and power. However, this requires a separate model for each word, a restriction we try to lift in this work.

3. Continuous-Space Paralinguistic Translation

We first briefly overview the model using linear regression to perform translation of acoustic information. In order to focus specifically on paralinguistic translation we describe the model in the context of a small-vocabulary lexical S2ST task: digit translation.

3.1. Speech Recognition

The first step of the process uses ASR to recognize the lexical and paralinguistic features of the input speech. This can be

represented formally as

$$\hat{\boldsymbol{E}}, \hat{\boldsymbol{X}} = \underset{\boldsymbol{E},\boldsymbol{X}}{\operatorname{argmax}} P(\boldsymbol{E}, \boldsymbol{X}|S), \tag{1}$$

where S indicates the input speech, $\boldsymbol{E}$ indicates the words included in the utterance and $\boldsymbol{X}$ indicates paralinguistic features of the words in $\boldsymbol{E}$. In order to recognize this information, we perform speech recognition using an HMM acoustic model and a language model that assigns a uniform probability to all digits. Viterbi decoding can be used to find $\hat{\boldsymbol{E}}$.

Using these recognition results we decide the duration and power vector $\boldsymbol{x_i}$ of each word e_i. The duration component of the vector is chosen based on the time spent in each state of the HMM in the Viterbi path. The power component of the vector is chosen in a similar way, and by taking the mean power value of each feature over frames that are aligned to the same state of the acoustic model. We express power as $[power, \Delta power, \Delta\Delta power]$ and join these features together as a super vector to control power in the translation step.

3.2. Lexical Translation

Lexical translation finds the best translation $\boldsymbol{J}$ of sentence $\boldsymbol{E}$.

$$\hat{\boldsymbol{J}} = \underset{\boldsymbol{J}}{\operatorname{argmax}} P(\boldsymbol{J}|\boldsymbol{E}), \tag{2}$$

where $\boldsymbol{J}$ indicates the target language sentence and $\boldsymbol{E}$ indicates the recognized source language sentence. Generally we can use a statistical machine translation tool like Moses [5], to obtain this translation in standard translation tasks, but for digit translation we can simply write one-to-one lexical translation rules with no loss in accuracy.

3.3. Paralinguistic Translation

Paralinguistic translation converts the source-side acoustic feature vector $\boldsymbol{X}$ into the target-side acoustic feature vector $\boldsymbol{Y}$ according to the following equation

$$\hat{\boldsymbol{Y}} = \underset{\boldsymbol{Y}}{\operatorname{argmax}} P(\boldsymbol{Y}|\boldsymbol{X}). \tag{3}$$

In particular, we control duration and power of each word using a source-side duration and power super vector $\boldsymbol{x_i} = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_{N_x}}]^\top$ and a target-side duration and power super vector $\boldsymbol{y_i} = \left[\boldsymbol{y_1}, \cdots, \boldsymbol{y_{N_y}}\right]^\top$. In these vectors N_x represents the number of HMM states on the source side and N_y represents the number of HMM states on the target side. $^\top$ indicates transposition. The sentence duration and power vector consists of the concatenation of the word duration and power vectors such that $\boldsymbol{Y} = [\boldsymbol{y_1}, \cdots, \boldsymbol{y_i}, \cdots, \boldsymbol{y_I}]$ where I is the length of the sentence. We can assume that duration and power translation of each word pair is independent from that of other words, allowing us to find the optimal $\boldsymbol{Y}$ using the following equation:

$$\hat{\boldsymbol{Y}} = \underset{\boldsymbol{Y}}{\operatorname{argmax}} \prod_i P(\boldsymbol{y_i}|\boldsymbol{x_i}). \tag{4}$$

The word-to-word acoustic translation probability $P(\boldsymbol{y_i}|\boldsymbol{x_i})$ is defined according to linear regression matrix that indicates that $\boldsymbol{y_i}$ is distributed according to a normal distribution

$$P(\boldsymbol{y_i}|\boldsymbol{x_i}) = N(\boldsymbol{y_i}; \boldsymbol{W}_{e_i,j_i}\boldsymbol{x_i'}, S) \tag{5}$$

where x' is $\left[1x^\top\right]^\top$ and $\boldsymbol{W}_{e_i,j_i}$ is a regression matrix (including a bias) defining a linear transformation expressing the relationship in duration and power between e_i and j_i. An important point here is how to construct regression matrices for each of the word pairs $\langle e, j\rangle$ we want to translate. In order to do so, we optimize each regression matrix on the translation model training data for $\langle e, j\rangle$ by minimize root mean squared error (RMSE) with a regularization term

$$\hat{\boldsymbol{W}}_{e,j} = \underset{\boldsymbol{W}_{e,j}}{\operatorname{argmin}} \sum_{n=1}^{N} ||\boldsymbol{y^*}_n - \boldsymbol{y}_n||^2 + \alpha||\boldsymbol{W}_{e,j}||^2, \tag{6}$$

where N is the number of training samples for the word pair, n is the ID of each training sample, $\boldsymbol{y^*}$ is target language reference word duration and power vector, and α is a hyper-parameter for the regularization term to prevent over-fitting.[1] This maximization can be solved in closed form using simple matrix operations.

3.4. Speech Synthesis

In the TTS part of the system we use an HMM-based speech synthesis system [6], and reflect the duration and power information of the target word paralinguistic information vector onto the output speech. The output speech parameter vector sequence $\boldsymbol{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_T]^\top$ is determined by maximizing the target HMM likelihood function given the target word duration and power vector $\hat{\boldsymbol{Y}}$ and the target language sentence $\hat{\boldsymbol{J}}$ as follows:

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}} P(\boldsymbol{O}|\hat{\boldsymbol{J}}, \hat{\boldsymbol{Y}}) \tag{7}$$

$$\text{subject to } \boldsymbol{O} = \boldsymbol{M}\boldsymbol{C}, \tag{8}$$

where $\boldsymbol{O}$ is a joint static and dynamic feature vector sequence of the target speech parameters and $\boldsymbol{M}$ is a transformation matrix from the static feature vector sequence into the joint static and dynamic feature vector sequence. While HMM TTS generally uses phoneme-based models, we instead used a word based HMM to maintain the consistency of feature extraction and translation. In this task the vocabulary is small, so we construct an independent context model.

4. Generalizing Paralinguistic Translation

In this section we describe two ways to generalize to a single model for all words in the vocabulary: global linear regression models and global neural network models.

4.1. Global Linear Regression Models

In the previous section, we described a method that requires the training of a regression matrix for each word pair $\langle e, j\rangle$. The simplest way to generalize this model is by not training a separate model for each word, but a global model for all words in the vocabulary. This can be done by changing the word-dependent regression matrix $\boldsymbol{W}_{e,j}$ into a single global regression matrix $\boldsymbol{W}$ and training the matrix over all samples in the corpus. However, this model can be expected to not be expressive enough to perform paralinguistic translation properly. For example, the mapping of duration from a one-syllable word to another one-syllable word, and from a one-syllable word to a two-syllable word would vary greatly, but the linear regression model only has the power to perform the same mapping for each word.

[1] We chose α to be 10 based on preliminary tests but the value had little effect on subjective results.

ASR	
Training sentences	8440
HMM states	16
MT	
Training utterances	445
Test utterances	55
Neural net structure	29/25/16
TTS	
Training utterances	445
HMM states	16

Table 1: Experimental Settings

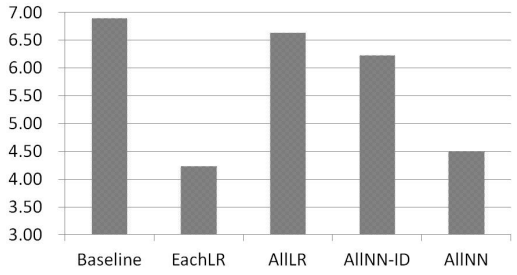

Figure 1: RMSE between the reference and system duration

4.2. Global Neural Network Models

As a solution to the problem of the lack of expressivity in linear regression, we propose a global method for paralinguistic translation using neural networks. Neural networks have higher expressive power due to their ability to handle non-linear mappings, and are thus an ideal candidate for the task. In addition, they allow for adding features for many different types of information following ASR, MT and TTS's common practice, such as word ID vectors, word position, left and right words of input and target words, part of speech, the number of syllables, accent types, etc. This information is known to be useful in TTS [6], so we can likely improve estimation of the output duration and power vector in translation as well.

In this research, we prepare a feed forward neural network that proposes the best output word acoustic feature vector $\hat{\boldsymbol{Y}}$ given input word acoustic feature vector $\boldsymbol{X}$. As additional features, we also add a binary vector with the ID of the present word set to 1, and the position of the output word. In this work, because the task is simple we just use this simple feature set, but this could be expanded easily for more complicated tasks.

For the sake of simplicity in this formulation we show an example with the word acoustic feature vector only. First, we set each input unit ι_i equal to the input vector value:

$$\iota_i = x_i. \tag{9}$$

The hidden units π_j are calculated according to the input-hidden unit weight matrix $\boldsymbol{W^h}$:

$$\pi_j = \frac{1}{1+\exp(-\alpha \sum_i w^h_{ij} \iota_i)}, \tag{10}$$

where α is gradient of sigmoid function. The output units ψ_k and final acoustic feature output y_k are set as

$$\psi_k = \sum_j w^o_{jk} \pi_j \tag{11}$$

$$y_k = \psi_k, \tag{12}$$

where $\boldsymbol{W^o}$ is hidden-output unit weight matrix. As an optimization criterion we use minimization of RMSE, which is achieved through simple back propagation.

5. Evaluation

5.1. Experimental Setting

We examine the effectiveness of the proposed method through English-Japanese S2ST experiments, summarized in Table 4.2. In these experiments we assume the use of S2ST in a situation where the speaker is attempting to reserve a ticket by phone in a different language. When the listener makes a mistake when listening to the ticket number, the speaker re-speaks, emphasizing the mistaken number. In this situation, if we can translate the paralinguistic information, particularly emphasis, this will provide useful information to the listener about where the mistake is. In order to simulate this situation, we recorded a bilingual speech corpus where an English-Japanese bilingual speaker emphasizes one word during speech in a string of digits. The lexical content to be spoken was 500 sentences from the AURORA2 data set, chosen to be word balanced by greedy search [7]. The training set is 445 utterances and the test set is 55 utterances.[2]

We further used this data to build an English-Japanese speech translation system that include our proposed paralinguistic translation model. We used the AURORA2 8440 utterance bilingual speech corpus to train the ASR module. Speech signals were sampled at 8kHz with utterances from 55 males and 55 females. We set the number of HMM states per word in the ASR acoustic model to 16, the shift length to 5ms, and other various settings for ASR to follow [8][9].

We selected 500 balanced sentence from the 8440 utterances of training data, and divide the utterances into 445 utterances for training and 55 utterances for testing paralinguistic translation. As the utterances were spoken in a noise-free environment with a high-quality close-talking mic, the speaker spoke slowly and clearly, and the utterances are included in the training data for ASR (although with different speakers), we achieved a 100% word accuracy on ASR.[3] For TTS, we use the same 445 utterances for training an independent context synthesis model. In this case, the speech signals were sampled at 16kHz. The shift length and HMM states are identical to the setting for ASR.

In the evaluation, we compare the following two baselines:

None No translation of paralinguistic information

EachLR Linear regression with a model for each word

with three global models of paralinguistic translation:

AllLR A single linear regression model trained on all words

AllNN A single neural network model trained on all words

AllNN -ID The AllNN model without additional features

In addition, we use naturally spoken speech as an oracle output.

[2]Freely available at `http://www.phontron.com/pcbeu`

[3]This simplifies our analysis as we do not need to consider the cases where ASR makes errors, but it will be interesting to investigate the effect of paralinguistic translation on erroneously recognized data in the future.

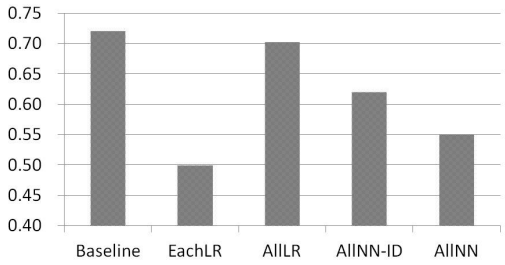

Figure 2: RMSE between the reference and system power

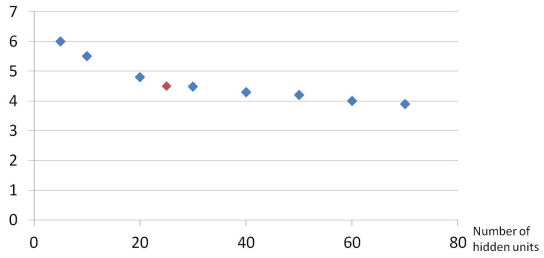

Figure 3: Duration RMSE for each number of hidden units

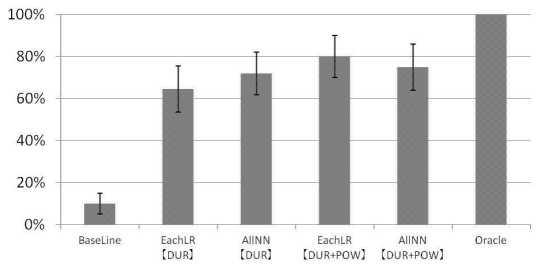

Figure 4: Prediction rate

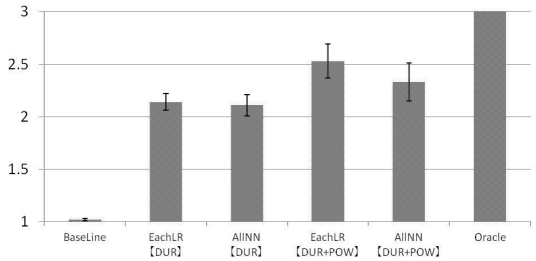

Figure 5: Subjective degree of emphasis

5.2. Automatic Evaluation

We first perform an objective assessment of the translation accuracy of duration and power, the results of which are found in Figure 1 and Figure 2. We compared the difference between the system duration and power and the reference speech duration and power in terms of RMSE.

From these results, we can see that the AllLR model is not effective at mapping duration and power information, achieving results largely equal to the baseline. The AllNN model without linguistic information does slightly better but still falls well short of the EachLR baseline. Finally, AllNN is able to effectively model translation of paralinguistic information, although accuracy of power lags slightly behind that of duration.

We also show the relationship between the number of NN hidden units and RMSE of duration in 3 (the graph for power was similar). It can be seen that RMSE continues to decrease as we add more units, but with diminishing returns after 25 hidden units. When comparing the number of free parameters in the EachLR model of previous work (17*16*11=2992) and the AllNN model with 25 hidden units (28*25+25*16=1100), it can be seen that we were able to significantly decrease the number of parameters with little change in accuracy.

5.3. Perception Tests

As an evaluation we asked native speakers of Japanese to evaluate how well emphasis was translated into the target language for the baseline, oracle, and EachLR and AllNN models when translating duration or duration+power.

The first experiment asked the evaluators to attempt to recognize the identities and positions of the emphasized words in the output speech. The overview of the result for the word and emphasis recognition rates is shown in Figure 4. We can see that all of the paralinguistic translation systems show a clear improvement in the emphasis recognition rate over the baseline. There is no significant difference between previous work and this work, indicating that the neural network learned a paralinguistic information mapping that allows listeners to identify emphasis effectively.

The second experiment asked the evaluators to subjectively judge the strength of emphasis with the following three degrees:

1: not emphasized

2: slightly emphasized

3: emphasized

The overview of the experiment regarding the strength of emphasis is shown in Figure 5. This figure shows that all systems show a significant improvement in the subjective perception of strength of emphasis. There seems to be a slight subjective preference towards EachLR when power is considered, reflecting the slightly larger RMSE found in the automatic evaluation.

6. Conclusion

In this paper we proposed a generalized model to translate duration and power information for S2ST. Experimental results showed that quality is similar to previous work, but without the need to create separate models for each word pair.

In future work we plan to expand beyond the digit translation task to a more general translation task with full sentences. The difficulty here is the procurement of parallel corpora with similar paralinguistic information for large-vocabulary translation tasks. We are currently considering possibilities including simultaneous interpretation corpora and movie dubs. Another avenue for future work is to expand to other acoustic features such as F0, which play an important part in the translation of full sentences. We would also like to empirically compare and contrast the output of our method to actual interpreter speech.

Acknowledgment: Part of this work was supported by JSPS KAKENHI Grant Number 24240032.

7. References

[1] T. Kano, S. Sakti, S. Takamichi, G. Neubig, T. Toda, and S. Nakamura, "A method for translation of paralinguistic information," in *Proceedings of IWSLT*, 2012.

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

[3] V. Kumar, S. Bangalore, and S. Narayanan, "Enriching machine-mediated speech-to-speech translation using contextual information," *Computer Speech and Language*, 2011.

[4] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of EMNLP*, 2007, pp. 868–876.

[5] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL*, 2007.

[6] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, 2009.

[7] J. Zhang and S. Nakamura, "An efficient algorithm to search for a minimum sentence set for collecting speech database," in *Proceedings of the 15th International Congress of Phonetic Sciences*, 2003.

[8] H. G. Hirsh and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000.

[9] R. Leonard, "A database for speaker independent digit recognition," in *Proceedings of ICASSP*, 1984.