*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Transferring Emphasis in Speech Translation Using Hard-Attentional Neural Network Models

*Quoc Truong Do, Sakriani Sakti, Graham Neubig, Satoshi Nakamura*

Graduate School of Information Science
Nara Institute of Science and Technology, Japan
{do.truong.dj3,ssakti,neubig,s-nakamura}@is.naist.jp

## Abstract

While traditional speech translation systems are oblivious to paralinguistic information, there has been a recent focus on speech translation systems that transfer not only the linguistic content but also emphasis information across languages. A recent work has tried to tackle this task by developing a method for mapping emphasis between languages utilizing conditional random fields (CRFs). Although CRFs allow for consideration of rich features and local context, they have difficulty in handling continuous variables, and cannot capture long-distance dependencies easily. In this paper, we propose a new model for emphasis transfer in speech translation using an approach based on neural networks. The proposed model can handle long-distance dependencies by using long short-term memory (LSTM) neural networks, and is able to handle continuous emphasis values through a novel hard-attention mechanism, which uses word alignments to decide which emphasis values to map from the source to the target sentence. Our experiments on the emphasis translation task showed a significant improvement of the proposed model over the previous state-of-the-art model by 4% target-language emphasis prediction $F$-measure according to objective evaluation and 2% $F$-measure according to subjective evaluation.

**Index Terms**: Speech translation, emphasis translation, paralinguistic transfer, paralinguistic translation.

## 1. Introduction

Speech translation technologies [1] have been gradually starting to break down the language barriers by translating linguistic information (meaning) of speech across languages. However, in addition to linguistic information, paralinguistic information also has a significant effect on human communication. Among the many types of paralinguistic information, *emphasis* is an important feature of speech that helps to convey focus or new information of an utterance. For example, in a conversation where words or phrases are misheard due to distracting factors such as noisy environments, people often put more focus on the misheard parts, helping other interlocutors capture this information more easily. If this emphasis information could be translated across languages, communication via speech translation systems could be a more smooth and natural experience.

Several works [2, 3, 4] have attempted to solve the problem of emphasis translation in different ways, but they all have similar model structures: an emphasis estimation system estimates emphasis information from speech signals, an emphasis translation system translates extracted emphasis values to another language, and an emphasized speech synthesis system synthesizes target language speech using the translated target language text and emphasis values. Although the structures are similar, approaches behind each of the components are different. Anumanchipalli et al. [2] translate emphasis by extracting and mapping $F_0$ patterns. However, emphasis is manifested by not only $F_0$, but also by changing the duration and power as well [5, 6]. Do et al. [7] have proposed another approach to estimate and translate emphasis considering all acoustic features such as power, duration, and $F_0$ patterns. The emphasis estimation system estimates a real-numbered value representing how emphasized a word is, and emphasis is translated using conditional random fields (CRFs). However, because CRFs require discrete labels, continuous emphasis levels must be quantized into discrete values. Moreover, while CRFs have the ability to capture local dependencies between neighboring labels, they cannot easily handle longer distance dependencies between words in separate parts of the sentence.

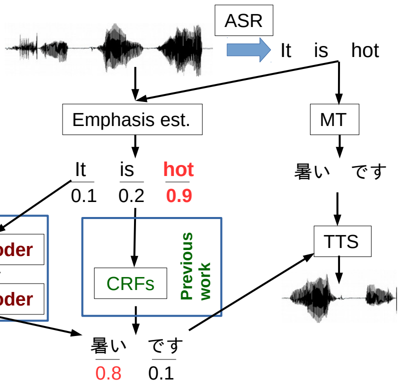

Figure 1: The proposed hard-attentional encoder-decoder emphasis translation model in the context of previous work.

In this paper, we propose a model that solves these problems using long short-term memory neural networks (LSTMs) [8]. LSTMs are a type of recurrent neural networks (RNNs) that have achieved impressive results for many tasks such as speech recognition [9, 10] and machine translation (MT) [11]. Particularly, LSTMs are capable of model long-term dependencies, overcoming the problems of local dependencies in CRFs. In addition, it is possible to define models that can handle continuous variables, and cost functions taking into account label distances, for example, mean squared errors.

Specifically, we design an emphasis translation model using attention-based encoder-decoder LSTMs and propose a novel *hard-attention* mechanism tailored to the emphasis translation task. The model consists of 2 components: an encoder that consists of LSTM cells that encode input features from the source language into vectors, and a decoder that uses these vectors to generate the target language emphasis sequence. Specifically, attention-based decoders take encoded vectors from all encoded input word vectors, and combines them together according to

http://dx.doi.org/10.21437/Interspeech.2016-898

alignment weights that are calculated on-the-fly during translation [12, 13]. However, when translating emphasis values, it is easy to obtain word alignments from an up-stream system that has translated the lexical content of utterances. Our proposed *hard*-attentional approach directly uses these alignments to generate emphasis sequences in the target language.

# 2. Word-level Emphasis Modeling and Word Alignment

## 2.1. Emphasis modeling

As mentioned, emphasized speech translation requires emphasis estimation, emphasis translation, and speech synthesis. For the former and latter, we use a word-level emphasis modeling technique based on linear-regression hidden semi-Markov models (LR-HSMMs) that has been proposed in [7].

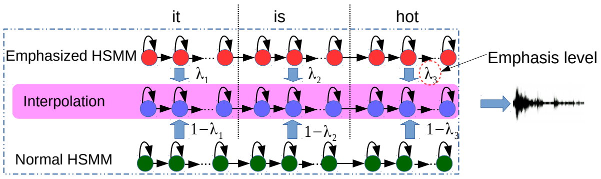

Figure 2: An example of LR-HSMM emphasis modeling.

In this method, emphasis is defined at word-level and represented by a real-numbered value indicating how emphasized the word is. For example, given the sentence “it is hot” and an emphasis sequence $\mathbf{\Lambda} = [0.1, 0.2, 0.8]$, the word “hot” is the most emphasized word in the sentence with the highest emphasis value of 0.8. The emphasis sequence $\mathbf{\Lambda}$ is used to construct an LR-HSMM by interpolating an *emphasized HSMM* and *normal HSMM*, the interpolation process is done on the Gaussian component levels. If we set $\lambda_i = 0$, then only the normal HSMM is used, leading normal speech to be synthesized, and vice versa.

The emphasis level sequence is estimated using a modified version of cluster adaptive training (CAT) [14]. At the first step, the emphasis sequence is set to 0, then by each iteration of CAT, the emphasis sequence is optimized to maximize the likelihood function $P(\mathbf{s}|\mathcal{M}, \mathbf{\Lambda})$, where $\mathbf{s}$ is the input audio signal and $\mathcal{M}$ is the set of model parameters.

## 2.2. Word alignment

Word alignments store information of which words correspond to each other in the source and target languages, and are vital to emphasis translation. This is because emphasis translation systems need to know which words in the source language are aligned to which words in the target language in order to accurately transfer emphasis from words to words or phrases to phrases. Specifically, in this work, alignments play a vital role in the hard attentional model proposed in Section 4.2.

In a speech translation system, these alignments can be extracted as a by-product of MT systems used to translate the surface text. At training time, given a parallel text, alignments can be obtained using unsupervised approaches [15].

# 3. Long short-term Memory Neural Nets

The LSTM [8] is a special kind of RNN that can capture long-term dependencies by having special units called *memory blocks*. The memory block manages information going through it using forget, input, and output gates. Given an input vector $\mathbf{x}_t$ at time $t$ and a hidden vector $\mathbf{h}_{t-1}$ and cell state $\mathbf{C}_{t-1}$ at time $t-1$, the flow of information can be described as follows:

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

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

- Calculating input gate $\mathbf{i}_t$ and estimated cell state $\tilde{\mathbf{C}}_t$:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_i \times [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i), & (2)\\ \tilde{\mathbf{C}}_t &= tanh(\mathbf{W}_C \times [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C), & (3)\end{aligned}$$

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

$$\mathbf{C}_t = \mathbf{f}_t \times \mathbf{C}_{t-1} + \mathbf{i}_t \times \tilde{\mathbf{C}}_t, \tag{4}$$

- Calculating the output vector $\mathbf{h}_t$:

$$\begin{aligned} \mathbf{v}_t &= \sigma(\mathbf{W}_v \times [\mathbf{h}_{t-1,\mathbf{x}_t}] + \mathbf{b}_v), & (5)\\ \mathbf{h}_t &= \mathbf{v}_t \times tanh(\mathbf{C}_t), & (6)\end{aligned}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the matrix and bias vectors of neural network layers. The core component of LSTMs is the cell state $\mathbf{C}_t$ (Eq. (4)) controlled by the forget gate $\mathbf{f}_t$ that is multiplied by the previous cell state values to decide which history information it should forget, and the input gate $\mathbf{i}_t$ that is multiplied to the estimated cell state to decide which information we should add to the cell state.

# 4. Emphasis Translation Using Hard-attentional Encoder-Decoders

The proposed emphasis translation system consists of 2 components: an LSTM encoder and an LSTM decoder as illustrated in Fig. 3. The encoder encodes features from the source language, and the decoder takes the encoded features to generate an emphasis sequence in the target language.

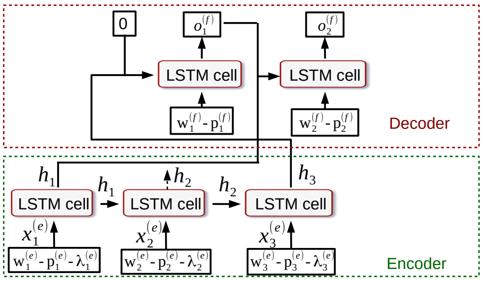

Figure 3: An unfolded hard-attentional encoder-decoder LSTM model for translating emphasis sequence $\boldsymbol{\lambda}^{(e)}$ into a target output sequence $\boldsymbol{o}^{(f)}$. It takes into account many linguistic features including the word sequence $\mathbf{w}_i^{(e,f)}$ and the part of speech sequence $\mathrm{p}_i^{(e,f)}$ from both source and target languages.

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

$$\mathbf{o}^{(f)} = f(\mathbf{x}^{(e)}), \tag{7}$$

where $\mathbf{o}^{(f)}$ is the target output sequence, $\mathbf{x}^{(e)}$ is the sequence of the source-language input vector $\mathbf{x}_i^{(e)}$.

## 4.1. The encoder

The encoder is a standard LSTM model that takes the input vector $\mathbf{x}_i^{(e)}$ consists of words ($w_i^{(e)}$), part-of-speech tags ($\mathrm{p}_i^{(e)}$), and emphasis levels ($\lambda_i^{(e)}$), then encodes them into a single vector that is suitable to predict emphasis levels.

The input PoS tags are converted into one-hot vectors with the size is equal to PoS vocabulary size. Also, word embeddings [16] are applied to map words into vectors that capture the similarity between the words. All these input features are concatenated into a single vector and fed to the encoder.

To train the encoder, we append 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, similarly to an auto-encoder model [17] (Fig. 4 (a)). The idea is we want the output hidden layer $\mathbf{h}$ to represent for features that are the most useful to predict emphasis levels (we call these an "emphasis representation"). The encoder is trained using minibatches to minimize the mean squared error criterion and adopt RMSprop algorithm [18] for model optimization. After training, the parameters are fixed, and not changed when training the decoder[1].

### 4.2. The decoder

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

The name hard-attentional comes from the way the decoder calculates the emphasis representation vectors used as input. The example in Fig. 3 demonstrates this mechanism. Assume that the word pairs $w_1^{(e)}$-$w_2^{(f)}$ and $w_3^{(e)}$-$w_1^{(f)}$ is aligned according to word alignments described in Section 2.2. To generate the output $o_2^{(f)}$, along with linguistic features $w_2^{(f)}$ and $\mathrm{p}_2^{(f)}$ and the previous output $\lambda_1^{(f)}$, the decoder takes the encoded $\mathbf{h}_1$ from the encoder output, because the word pair $w_1^{(e)}$-$w_2^{(f)}$ are aligned. For unaligned words, we use zero vectors as the emphasis representation vectors.

Depending on how we define the output sequence $\mathbf{o}^{(f)}$, the model structure will be different. Thus, we proposed 2 types of models as follows,

- **LSTM_emph**: The model predicts the target emphasis sequence $\boldsymbol{\lambda}^{(f)}$ directly: $\boldsymbol{\lambda}^{(f)} = \mathbf{o}^{(f)}$.
- **LSTM_diff**: The output of the model is considered as the difference from the input emphasis level, thus the target emphasis level of the $j$-th word is calculated by, $\lambda_j^{(f)} = o_j^{(f)} + \lambda_i^{(e)}$, where the model gets "attention" from the word $w_i^{(e)}$.

Regarding the training process, we use the same squared error loss function as in the encoder.

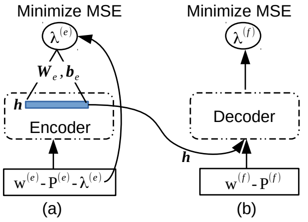

Figure 4: Training procedure of the hard-attentional model.

## 5. Experiments

We conduct emphasis translation experiments from English to Japanese using a bilingual English-Japanese emphasized speech corpus [6]. Details of the corpus and model setup are described in the following sections.

### 5.1. Experimental setup

#### *5.1.1. Corpus*

The corpus consists of 966 parallel utterances of English and Japanese. In each language, at least one of the content words in the sentence is emphasized, and the number of emphasized words is the same 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 the experiment, we divide 966 utterances of each speaker to 2 sets of 866 and 100 samples such that the same sentences are used for all speakers. We then pair 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 consists of 157 emphasized words, in which 30 exist in the training data and 127 do not.

#### *5.1.2. Emphasis translation procedure & measurement*

In this paper, to evaluate the performance of emphasis translation in isolation, we assume that the MT system produces 100% correct translation outputs. Word alignments between the input and output are derived using the pialign toolkit [15].

To measure emphasis translation accuracy, we first perform emphasis translation to derive the target emphasis sequence, then measure the accuracy of emphasis in the target language both objectively or subjectively (as shown in Fig. 5). In the objective evaluation, the target emphasis values are classified into "emphasized" or "not emphasized" using a threshold of 0.5[2] and compared with the true values. In the subjective evaluation, we first synthesize audio from the translated emphasis sequence, and then the output audio is given to 7 Japanese native listeners to predict the emphasized words[3]. In both evaluations, we calculate $F$-measure ranging from 0 to 100 representing how accurately the system can preserve emphasis in the target language.

#### *5.1.3. Encoder-decoder LSTMs*

As has been described in Section 4, to simplify implementation, the encoder and decoder are trained separately.

**The encoder:** The input of the encoder consists of words, PoS tags, and emphasis levels. The input layer has a size of 138 including 100 dimensions of word embedding, 37 dimensions of one-hot PoS, and emphasis level. The hidden layer has a size of 100.

**The decoder:** The input gate consists of 100 dimensions of word embedding and 17 dimensions of one-hot PoS. The attentional vector taken from the encoder is added to the output of the input gate. The input words and PoS are also converted into word-embedding and one-hot vectors, respectively.

The word embeddings for both the encoder and decoder are pre-trained using the BTEC travel conversation corpus [19] using word2vec toolkit [16].

[1] The reason why we adopted this separate training approach is because it is not straightforward to handle errors in back-propagation from the decoder to the encoder due to alignments of source-target in each minibatch being different across sentences. There are some possible solutions to tackle this problem, which we leave for future work.

[2] This has been reported in the previous work [7] as having the best performance to classify emphasized and normal words.

[3] 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.

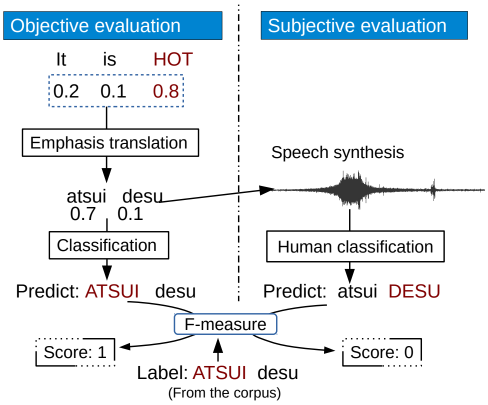

Figure 5: An example of the emphasis translation procedure and measurement methods.

### 5.1.4. CRFs

We keep the configuration of CRF models the same with the previous work in [7]. The word-level emphasis is quantized to the closest of $\{0, 0.3, 0.6, 0.9\}$. The input features are words, PoS, and PoS context in the target language side. The model predicts the target side emphasis sequence directly. This setting has shown to have the best performance compared to other features combinations.

## 5.2. Objective evaluation

First, we compared objective accuracy on the same corpus as in [7] with 916 training samples and 50 testing samples. The results showed that the proposed method achieved 92.6% $F$-measure, which is higher than the previous work by 1%. Although the dataset is 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 conduct larger scale experiments with the dataset that has been described in the Section 5.1.1.

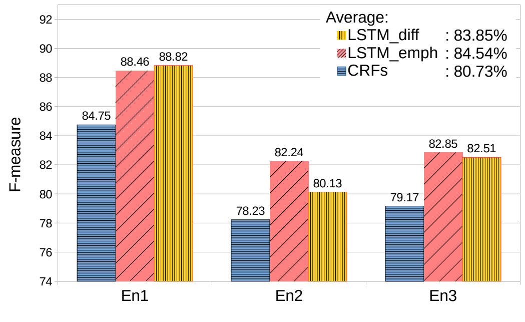

Figure 6: Objective emphasis prediction $F$-measure.

Fig. 6 shows the objective $F$-measure for emphasis prediction on this larger data. As we can see, in all 3 test sets and in the average, the proposed methods perform better than the CRFs. According to the bootstrap resampling significance test [20], both results are significant at the $p < 0.01$ level. On the other hand, the difference between *LSTM_diff* and *LSTM_emph* was not found to be significant, demonstrating that the LSTM model can learn emphasis level differences between aligned words without defining them explicitly in the equations.

Furthermore, we perform a detailed analysis into the advantage of the proposed model with respect to the use of continuous variables. If continuous variable are useful, we can expect that emphasis values in the middle of the range will be better modeled 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 calculate $F$-measure for the *CRFs* and *LSTM_emph* on individual sets[4]. The result is shown in Table 1, indicating that both systems have equivalent performance when the word is certainly considered as normal or emphasized (emphasis levels fall below 0.3 or higher than 0.6), but when emphasis levels fall between 0.3-0.6, *LSTM_emph* outperform *CRFs*. This demonstrates the limitation of *CRFs*, which require emphasis level quantization to handle continuous variables while LSTMs do not.

Table 1: $F$-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 |

## 5.3. Subjective evaluation on emphasis translation

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

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

# 6. Conclusion

In this paper, we explored encoder-decoder neural net approaches and proposed "hard"-attentional LSTMs for emphasis translation tasks. Compared to previous works, the proposed model has achieved significantly better performance. This is a result of the fact that the model does not require any emphasis quantization and takes into account emphasis label relationships in the loss function. We also found out that the model can learn emphasis level differences between aligned words across languages without defining them explicitly in the equation.

However, in subjective evaluation the improvement of the proposed method over CRFs become smaller. Future works will improve the training algorithm for "hard"-attentional mechanism and also adopt neural net approaches for the emphasis estimation and synthesis components, making a unified neural-net-based system.

[4] Because the accuracies of *LSTM_diff* and *LSTM_emph* are similar, we only show the result of *CRFs* and *LSTM_emph* from here on.

## 7. References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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 *Proceedings of INTERSPEECH*, 2015.

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

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

[10] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of ICML*, 2014, pp. 1764–1772.

[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, H. Pham, and C. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[13] 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.

[14] M. Gales, “Cluster adaptive training of hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000.

[15] G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, “An unsupervised model for joint phrase alignment and extraction,” in *Proceedings of ACL*, 2011, pp. 632–641.

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

[17] 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.

[18] A. Graves, “Generating sequences with recurrent neural networks,” *CoRR*, vol. abs/1308.0850, 2013.

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

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