INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Toward Expressive Speech Translation: A Unified Sequence-to-Sequence LSTMs Approach for Translating Words and Emphasis

Quoc Truong Do, Sakriani Sakti, Satoshi Nakamura

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

Abstract

Emphasis is an important piece of paralinguistic information that is used to express different intentions, attitudes, or convey emotion. Recent works have tried to translate emphasis by developing additional emphasis estimation and translation components apart from an existing speech-to-speech translation (S2ST) system. Although these approaches can preserve emphasis, they introduce more complexity to the translation pipeline. The emphasis translation component has to wait for the target language sentence and word alignments derived from a machine translation system, resulting in a significant translation delay. In this paper, we proposed an approach that jointly trains and predicts words and emphasis in a unified architecture based on sequence-to-sequence models. The proposed model not only speeds up the translation pipeline but also allows us to perform joint training. Our experiments on the emphasis and word translation tasks showed that we could achieve comparable performance for both tasks compared with previous approaches while eliminating complex dependencies.

Index Terms: speech-to-speech translation, paralinguistic translation, machine translation.

1. Introduction

Humans communicate with various degrees of expressions to provide strong clues about their intentions, attitudes, and emotions. Emphasis is one factor of expressiveness deliberately used by speakers to modify linguistic information [1]. A common example of emphasis is in misheard situations where speakers put more focus on particular parts (words or phrases) of utterances to help other interlocutors capture the important information of the utterance. Tsiartas et al. [2] identified importance of emphasis in cross-language communication and argued that it should be considered in S2ST systems to preserve the quality of speech translation.

To leverage conventional S2ST systems to handle emphasis, previous works on emphasis translation [3, 4, 5] proposed additional components to estimate and translate it. The general idea is to first estimate a representation of the source language emphasis. Anumanchipalli et al. [3] used F_0 patterns to represent emphasis and Do et al. [4] used emphasis weights instead[1]. Then an emphasis translation component takes the estimated emphasis and predicts the target emphasis representations. Although these approaches can handle emphasis in S2ST systems, they make the translation pipeline more complex. In particular, Do et al. [4] requires a separate word alignment models before the emphasis translation to map the emphasis weights, and Anumanchipalli et al. [3] also needs phrase alignments to map F_0 patterns. However, the word alignment can only be obtained after word translation, meaning that to translate emphasis, we need to wait for the machine translation system to predict all of the target language sentences, creating a large delay to get the target output speech.

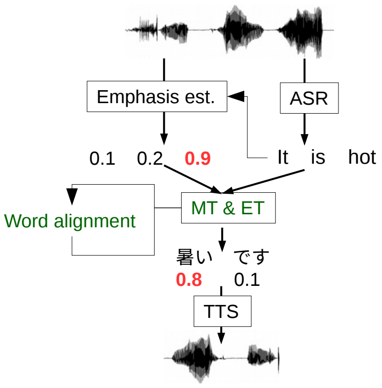

Figure 1: *Illustration of proposed approach that jointly trains and predicts words and emphasis. Machine translation (MT) and emphasis translation (ET) are combined into a unified model. Word alignments are also automatically inferred by attention mechanism of seq2seq approaches.*

On the other hand, sequence-to-sequence (seq2seq) (also known as neural network machine translation-NMT) approaches [6] have recently claimed the state-of-the-art performance in machine translation and are appealing in end-to-end tasks where the minimal domain knowledge is required. This approach can internally infer word alignments using attentional mechanism [7]. A recent work [8] proposed hard-attention sequence-to-sequence models to translate emphasis. This approach, however, still relies on a separate word alignment model to achieve the best performance. If emphasis can be integrated into NMT, the emphasis translation pipeline will become not only simpler, but we can also perform joint optimization of words and emphasis translation.

In this paper, we propose an approach based on seq2seq models that jointly trains and predicts words and emphasis together based on sequence-to-sequence models. The proposed approach eliminates complex dependencies while preserves good translation performance of words and emphasis. It makes the whole translation pipeline much simpler and more compact than previous approaches (Fig. 1). Separated word alignment models are no longer required, and the model can predict emphasis with one-word delay instead of full-sentence delay.

2. Emphasis Representation

Emphasis is manifested by changing many acoustic features including F_0, power, and duration [9, 10]. Therefore, to effectively convey emphasis in S2ST systems, all these acoustic features must be taken into account. In this paper, we follow the representation of emphasis as emphasis weights [4] based on linear-regression hidden-semi Markov models (LR-HSMMs).

[1] Emphasis weights are real-numbered values representing how greatly a word is emphasized.

http://dx.doi.org/10.21437/Interspeech.2017-896

The emphasis weight is a scalar number that intuitively represents how much a word is emphasized (Fig. 1). The advantage of this representation is that the emphasis weight is estimated using all of the acoustic features that are used to manifest emphasis. With this representation, the emphasis translation task resembles the translation of a sequence of emphasis weights that is similar to machine translation, which translates a sequence of words.

Emphasis weights are used as interpolation parameters between normal HSMMs and emphasized HSMMs to construct LR-HSMMs. Normal HSMMs are constructed from all normal Gaussians, which are trained from normal speech data. The same concept is applied to emphasize HSMMs.

In the emphasis weight estimation process, the weight is estimated using a modified version of cluster adaptive training (CAT) [11]. During the speech synthesis process, given an emphasis sequence, we first construct a sequence of LR-HSMM states and infer the emphasized speech parameters. If the emphasis weights are set to 0, the LR-HSMMs are equivalent to normal HSMMs that synthesize normal speech and vice versa.

3. Sequence-to-Sequence-based for Machine Translation

NMT models translate a source language word sequence that consists of n words $\boldsymbol{W}^{(s)} = \{w_1^{(s)}, \ldots, w_n^{(s)}\}$ to a target language word sequence $\boldsymbol{W}^{(t)} = \{w_1^{(t)}, \ldots, w_m^{(t)}\}$ by directly modeling conditional probability $P(\boldsymbol{W}^{(t)}|\boldsymbol{W}^{(s)})$. A basic form of NMT models consists of two components: an encoder that encodes source word representation $\boldsymbol{s}$ and a decoder that predicts the target words.

A variant of NMT called attentional NMT (Fig. 2) [12, 13] effectively translates long sentences by introducing an attention layer that works as an alignment model. This layer gives the decoder more information about which words of the source sentence are more important to predict the current target word.

The probability of predicting word $w_i^{(t)}$ can be calculated as follows,

$$P(w_i^{(t)}|w_{<i}^{(t)}, \boldsymbol{s}) = \text{softmax}(g(\widetilde{\boldsymbol{h}}_i^{(t)})), \tag{1}$$

where g is a function that maps hidden activation vector $\widetilde{\boldsymbol{h}}_i^{(t)}$ to a vocabulary-sized vector, which is computed by applying linear layer $\boldsymbol{W}_c$ over a concatenation vector of context vector $\boldsymbol{c}_i$ and current hidden vector $\boldsymbol{h}_i$,

$$\widetilde{\boldsymbol{h}}_i^{(t)} = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_i; \boldsymbol{h}_i]) \tag{2}$$

Context vector $\boldsymbol{c}_i$ is, in fact, a weighted average of source hidden vectors $\boldsymbol{H}^{(s)} = [\boldsymbol{h}_1^{(s)}, \ldots, \boldsymbol{h}_n^{(s)}]$, where weighted vector $\boldsymbol{a}_i$ is computed by,

$$\boldsymbol{a}_i = \text{softmax}(\text{dot}(\boldsymbol{H}^{(s)}, \boldsymbol{h}_i^{(t)})) \tag{3}$$

Weighted vector $\boldsymbol{a}_i$ acts as a word alignment score, where the highest value indicates the source language word that is aligned with the current target language word.

4. Unified Framework for Translating Words and Emphasis

In this section, we propose a unified framework based on attentional NMT that can translate words and emphasis. We chose NMT-based approaches for two main reasons. First, they can capture long-distance dependencies and handle continuous values. This is particularly important for emphasis translation because emphasis weights are continuous. Second, a recent work [8] applied a hard-attention NMT for emphasis translation and showed a significant improvement. We expect that integrating emphasis directly into NMT will eliminate complex dependencies while preserving the high performance of emphasis translation.

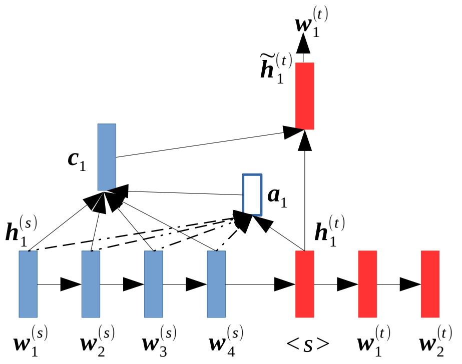

Figure 2: *Neural network machine translation with attention layer.*

The major difficulty when integrating emphasis with word translation is that the amount of text data usually dominates the amount of emphasis data. This is because emphasis data are derived from parallel emphasized speech that is much harder to collect than parallel text data, which can be massively collected by crawling websites [14].

The unified translation model can be defined as follows. Given a source language word and an emphasis sequence 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.

4.1. Encoder with emphasis weights

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

$$\boldsymbol{h}_i^{(s)} = enc([w_i^{(s)}, e_i^{(s)}]). \tag{4}$$

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

$$\boldsymbol{h}_i^{(s)} = [enc(w_i^{(s)}), e_i^{(s)}] \tag{5}$$

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

4.2. Decoder with emphasis weights

As illustrated in Fig. 3, the decoder has two components. A word prediction layer follows the standard NMT Eq. 1, 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:

$$e_i^{(t)} = \boldsymbol{W}_e([\widetilde{\boldsymbol{h}}_i^{(t)}, w_i^{(t)}]). \tag{6}$$

However, as stated above, the lack of emphasis data compared with the text data might lead to the problem where the effect of the source emphasis might be saturated when going through many hidden layers. To overcome this problem, we

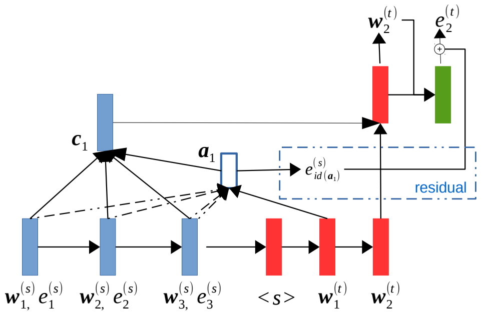

Figure 3: *Unified word-emphasis translation framework with word dependencies and residual connection.*

utilize residual connection in the way that the source emphasis weight is also used when predicting target emphasis weights (Fig. 3),

$$e_i^{(t)} = \boldsymbol{W}_e([\widetilde{\boldsymbol{h}}_i^{(t)}, w_i^{(t)}]) + e^{(s)}_{id(\boldsymbol{a}_i)}, \qquad (7)$$

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

4.3. Training procedure

To train the model, we utilize two objective functions, cross entropy (CE) for word prediction and mean square error (MSE) for emphasis prediction, because the CE function performs much better than MSE with discrete labels, which is the case of word prediction. Since emphasis weights are continuous, CE function cannot be utilized as the objective function for emphasis prediction.

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

5. Experiments

5.1. Experimental setup

5.1.1. Corpus

The corpus consists of two parts, parallel texts for the machine translation task and parallel emphasized speech for the emphasis translation task. The parallel text were derived from the BTEC corpus [15] that consists of ~466,000 parallel sentences. The parallel emphasized speech data are bilingual English-Japanese emphasized speech that consists of 966 parallel emphasized speech utterances spoken by three native English and five native Japanese speakers [10].

To create training and testing data for the experiment, we utilize previously described emphasis estimation process based on CAT [16], resulting in 966 emphasis weight sequences for each speaker. Then we divide the 966 utterances of each speaker into two sets of 866 and 100 samples. After that, we pair the 866 utterances of each English speaker with those of all five Japanese speakers, resulting in 4330 training and 100 testing samples for emphasis translation experiments. As for the BTEC data, we first divide them into 460,000 training, 1000 developing, and 5000 testing sentences. Then, because the parallel text data are very large and it is impractical to collect parallel emphasized speech for it, we create fake emphasis data for all of the sentences so that content words have emphasis weights of 0.99 and other words have emphasis weights of 0.01[2].

5.1.2. Training details

Our encoder and decoder models have 1 layer (unless stated otherwise), with 512 cells, and 512-dimensional word embeddings. We train for a maximum of 20 epochs using the RMSprop algorithm [17]. Emphasis prediction layer $\boldsymbol{W}_e$ is frozen when training with fake emphasis data to avoid learning from unrealistic emphasis weights.

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

5.1.3. Measurement metric

In this paper, we separately evaluate the performances of the machine and emphasis translations by BLEU and F-scores, respectively. We chose F-score for emphasis evaluation due to the fact that it objectively models human performance on emphasis detection task on the target language side. While other measurement metrics taken into account continuous numbers such as RMSE could be used, it is not easy for human to detect the continuous emphasis level for words.

To calculate the F-score for the emphasis evaluation, the target emphasis values are classified as “emphasized” or “not emphasized” using a threshold of 0.5[3] and compared with the true values. Because the system is speaker dependent, we train individual models for each speaker and average the scores over all models.

5.2. Effect of using emphasis as additional features on machine translation

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

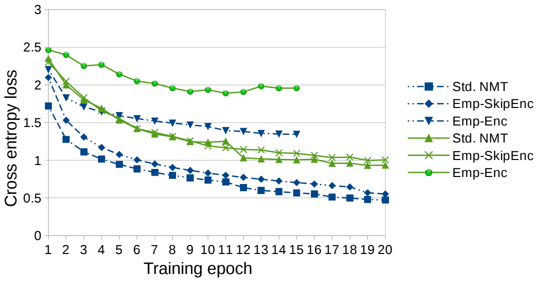

Figure 4: *Effect of emphasis on machine translation. The solid and dash lines denote the MT performance on the development set and the training set, respectively.*

We first keep the same decoder structure like standard NMT systems so that no emphasis prediction is performed. Then we evaluate two encoders with emphasis weights added in different positions as described in Section 4.1. The baseline is the standard NMT system without emphasis weights (*Std. NMT*). Fig. 4 shows the result of the cross entropy loss of word prediction performance on the training and development sets. The loss is higher in both approaches (*SkipEnc* and *Emp-Enc*) than the *Std. NMT*, indicating that emphasis does not help to improve NMT

[2] The fake emphasis levels do not have significant effect on emphasis translation because the emphasis decoder does not take them into account.

[3] This was reported in previous work [16] as having the best performance to classify emphasized and normal words.

performance. We hypothesize that such a negative effect is due to the fact that emphasis weights are paralinguistic while NMT translating linguistic information. Using emphasis weights only as an additional feature without translating is insufficient for the model to learn anything useful from emphasis.

Although the NMT performance is degraded when using 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 experiments use the *SkipEnc* model.

5.3. Emphasis translation performance

In this experiment, we fully train the model for both emphasis and word prediction. Fig. 5 shows the F-score, precision, and recall for emphasis prediction using the *SkipEnc* encoder with *baseline* and *residual* decoders.

Looking at the F-score, the *residual* decoder outperforms the *baseline* decoder by a 2.7% F-score. The *baseline* decoder's precision is, however, higher than of the *residual* one, indicating that the *residual* connection makes more mistakes that predict high emphasis weights for normal words. Similarly, the high score for *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 can be gained by combining them. In the next section, we describe our combination technique and compare its result with previous works.

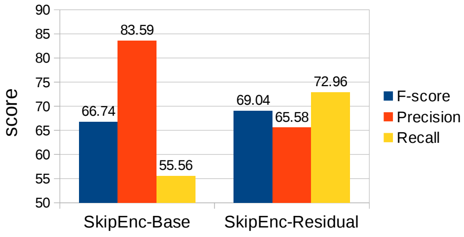

Figure 5: *Emphasis translation performance in unified translation framework.*

5.4. Model combination for emphasis translation

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

We also perform emphasis translation using previous approaches based on conditional random fields (CRFs) [16] and LSTMs hard attention models [8]. The input features for these approaches are words and emphasis weights that resemble the proposed approach. The result is shown in Fig. 6. Compared with CRFs, our proposed approaches perform better with ~5% F-score and have a closed performance with the LSTM hard-attention approach with a ~2% lower F-score.

The result matches our expectation because both 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.

5.5. Machine translation performance

We evaluate the machine translation performance with various depths of hidden layers. The baseline system is the standard NMT without emphasis weights used in both the encoder and decoder. As shown in Table 1, with hidden layer depths of 1 and 2, the performance different of the proposed approach and the baseline is negligible, indicating that optimizing the model with emphasis weights can compensate for the negative effect of emphasis found in Section 5.2.

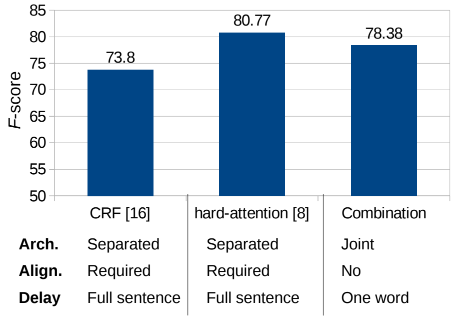

Figure 6: *Comparison of the emphasis translation performance of the proposed and previous approaches. The graph also showed the differences in terms of translation architecture (Arch.), word alignment requirement (Align.), and the translation delay (Delay).*

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

Table 1: *Machine translation performance in unified translation framework. Various depth of hidden layers denoted as* d(1,2,3) *were evaluated.*

System	BLEU
Baseline (d1)	27.67
SkipEnc-Base (d1)	27.25
SkipEnc-Residual (d1)	27.19
Baseline (d2)	27.44
SkipEnc-Base (d2)	27.70
SkipEnc-Residual (d2)	27.72
Baseline (d3)	23.68
SkipEnc-Base (d3)	**25.41**
SkipEnc-Residual (d3)	**26.36**

6. Conclusions

This paper presents the first attempt that analyzed the effect of emphasis on machine translation and jointly translated emphasis and words in a unified model. Compared to previous works on emphasis translation, our proposed models achieved comparable performance while eliminating complex dependencies. In a machine translation task, the proposed models demonstrated that emphasis weights reduce over-fitting issues. Future work will integrate emphasis estimation and synthesis components into the model, making a completely end-to-end expressive speech translation system.

7. Acknowledgments

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101, JP17H00747, and JP17K00237.

8. References

[1] H. Fujisaki, “Information, prosody, and modeling-with emphasis on tonal features of speech,” in *Speech Prosody*, 2004.

[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, pp. 8396–8400.

[3] 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.

[4] Q. T. Do, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, “Preserving word-level emphasis in speech-to-speech translation,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 25, pp. 544–556, 2016.

[5] 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.

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

[7] M. T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *EMNLP*, 2015.

[8] Q. T. Do, S. Sakti, G. Neubig, and S. Nakamura, “Transferring emphasis in speech translation using hard-attentional neural network models,” in *InterSpeech*, September 2016.

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

[10] Q. T. Do, 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.

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

[12] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proceedings of ICLR*, 2015.

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

[14] J. Tiedemann, “Parallel data, tools and interfaces in OPUS,” in *Proceedings of LREC*, 2012.

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

[16] 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.

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