# Data Augmentation for End-to-End Speech Translation: FBK@IWSLT '19

**Article** · November 2019

DOI: 10.5281/zenodo.3525492

**5 authors**, including:

Mattia Antonino Di Gangi
Fondazione Bruno Kessler
**31** PUBLICATIONS **124** CITATIONS
SEE PROFILE

Amirhossein Tebbifakhr
Fondazione Bruno Kessler
**10** PUBLICATIONS **22** CITATIONS
SEE PROFILE

Marco Turchi
Fondazione Bruno Kessler
**120** PUBLICATIONS **1,868** CITATIONS
SEE PROFILE

# Data Augmentation for End-to-End Speech Translation: FBK@IWSLT '19

*Mattia A. Di Gangi*[1,2], *Matteo Negri*[1], *Viet Nhat Nguyen*[2*], *Amirhossein Tebbifakhr*[1,2], *Marco Turchi*[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Italy
{digangi|negri|vnguyen|atebbifakhr|turchi}@fbk.eu

## Abstract

This paper describes FBK's submission to the end-to-end speech translation (ST) task at IWSLT 2019. The task consists in the "direct" translation (i.e. without intermediate discrete representation) of English speech data derived from TED Talks or lectures into German texts. Our participation had a twofold goal: *i)* testing our latest models, and *ii)* evaluating the contribution to model training of different data augmentation techniques. On the model side, we deployed our recently proposed S-Transformer with logarithmic distance penalty, an ST-oriented adaptation of the Transformer architecture widely used in machine translation (MT). On the training side, we focused on data augmentation techniques recently proposed for ST and automatic speech recognition (ASR). In particular, we exploited augmented data in different ways and at different stages of the process. We first trained an end-to-end ASR system and used the weights of its encoder to initialize the decoder of our ST model (*transfer learning*). Then, we used an English-German MT system trained on large data to translate the English side of the English-French training set into German, and used this newly-created data as additional training material. Finally, we trained our models using SpecAugment, an augmentation technique that randomly masks portions of the spectrograms in order to make them different at every training epoch. Our synthetic corpus and SpecAugment resulted in an improvement of 5 BLEU points over our baseline model on the test set of MuST-C En-De, reaching the score of 22.3 with a single end-to-end system.

## 1. Introduction

End-to-end speech-to-text translation [1, 2] is the task of directly translating a speech presented as an audio signal in a source language into a text in a different target language. The main goals of translating from the audio without intermediate discrete representations are to reduce translation latency and to avoid the cumulative effect of *error propagation*. Cumulative errors typically affect the so-called cascade approaches to ST, which rely on a two-step processing involving separate speech recognition and translation components. Another advantage of the direct approach is the possibility to translate languages without a formal script [3, 4], in particular for emergency situations. However, the high expectations so far have not been matched by the empirical results, which are generally much worse with an end-to-end system than with a cascade approach [5]. First, the main difficulty of this approach comes from the scarce availability of annotated corpora for the end-to-end task. These corpora should come in the form of large collections of (*src audio*, *tgt text*) pairs, which are costly to acquire and still available only in small amount compared to the wealth of data existing for the two related tasks of automatic speech recognition (ASR) and machine translation (MT) [6, 7, 8, 9]. Second, the transfer learning methods proposed so far to re-use models from the two related tasks of ASR and MT resulted to be ineffective [10]. In particular, while pre-training the encoder with an ASR model helps to gain some BLEU points, pre-training the decoder with an MT system gives marginal or no improvement at all [2, 11]. Nonetheless, the end-to-end approach still has potential as it provides a natural way to directly train ST models with parallel audio-translation data. While the availability of this type of resource is still limited, it is expected to grow in future as this task attracts more research. In the meanwhile, data augmentation can represent a viable workaround that has already been used in some works claiming to have bridged the gap with the cascade approach [12, 13].

This is the second time that we participate in this task. Last year, our submissions exploited the ST data released for that edition [14] to train an LSTM-based model similar to the one introduced in [2]. As we found some noise in the training corpus, we performed a data filtering process, which yielded some gain in performance [15]. This year, our submission is improved under several aspects: *i)* we trained our models on larger data with a better segmentation, *ii)* we used our recently-proposed deep learning architecture [16] based on Transformer [17] and leveraging more parameters, and *iii)* we experimented with different techniques for data augmentation. In particular, we still used *transfer learning* of the encoder weights from an ASR system [2, 3]. Additionally, we generated synthetic data by forward-translating transcripts from English into German [12, 5], and we used SpecAugment[18], a technique for online data augmentation that achieved state-of-the-art results on ASR with end-to-end models. Compared to our baseline system, which uses only

[*]Work done during a summer internship at FBK

transfer learning, our best single system obtained a performance improvement of about 5 BLEU points.

## 2. Data and augmentation

We trained our model using a combination of clean and synthetic data, which are all derived from TED talks. In particular, we only exploited the MuST-C [6] corpus because it is the most recent and the largest one among the usable corpora. Indeed, the talks included in the other resources available to participants (i.e. WIT$^3$ – [19] and the Speech-Translation TED corpus downloadable from the task web page[1]) are either contained in MuST-C or they were filtered out of it because too noisy.

### 2.1. Clean Data

To train our end-to-end ST model, we used only the English-German portion of MuST-C [6], which consists of 408 hours of speech in English and 234K sentence pairs. The MT system used to generate the synthetic data was trained on all the allowed En-De MT data from WMT 2019 [20], which amount to 42M sentence pairs and over 900M words for both languages. The ASR model used to initialize the encoder weights of our ST model has been trained on the speech-transcript data from the En-De portion of MuST-C.

### 2.2. Synthetic Data

Forward-translating the transcripts of parallel data for ASR is a technique that showed important improvements [12]. Last year's best submission [5] used this approach to generate synthetic data starting from the TED LIUM corpus [21], the only dataset allowed for ASR, which is comparable in content and slightly smaller in size than the IWSLT speech translation dataset. Similarly to that work, we decided to translate the English transcripts of the En-Fr portion of MuST-C, which accounts for 492 hours of speech and 280K parallel segments. In a final experiment, we also translated the transcripts from the En-Es portion of MuST-C, containing slightly more hours than En-Fr. However, the inclusion of this additional, highly overlapping content did not result in further performance gains.

### 2.3. SpecAugment

SpecAugment [18] is a data augmentation technique that works by modifying the input spectrograms. It is an online technique as it is applied during training, and it is applied differently at each training iteration. In practice, SpecAugment is similar to dropout [22], as it zeroes out portions of the input. Unlike dropout, however, it does not operate point-wise but on entire rows and/or columns, *de facto* canceling sound events during time and/or entire frequency bands. Each mask, either horizontal or vertical, takes two parameters: starting position $s$ and length $l$, both sampled from a distribution. Once the mask orientation is chosen (i.e. either horizontal, vertical or both), all the values in the lines (rows or columns) ranging from $s$ to $s + l$ are set to zero, which is the mean value of the spectrograms after mean and variance normalization. [18] proposes also *time warping* as a third component for SpecAugment, but it also shows that it is more computationally expensive than masking and the improvement is marginal. For these reasons, we decided not to use it.

[1]http://sites.google.com/view/iwslt-evaluation-2019/speech-translation

## 3. Methods

### 3.1. Model architecture

We trained one model for MT and one for ST En-De. Since both models are based on Transformer, here we first introduce its general architecture. Then, we will move to our ST-oriented adaptation.

Transformer [17] is an attentional encoder-decoder architecture that consists of multiple layers of the same type stacked together. The layers are slightly different in the encoder and in the decoder. Encoder layers consist of two sub-layers: self-attention and feed-forward. The self-attention sublayer uses the multi-head attention mechanism to compute multiple parallel self-attentions of the input sequence. The feed-forward layer is a stack of two affine transformations, where the first one is larger in size than the second one, and it is also followed by a ReLU nonlinearity. Residual connections sum the input of every sub-layer to its output, and the result of residual connections [23] is normalized with layer normalization [24]. Transformer's decoder layers are similar to the encoder layers but they have an additional cross-attention sub-layer after the self-attention. Cross-attention computes multi-head attention of its input with the output of the encoder.

Our ST model is the S-Transformer described in [16]. The difference with the original Transformer model is in the part that precedes the first Transformer encoder layer. While in MT this part is normally preceded by (sub-)word embeddings, S-Transformer processes the input spectrograms with additional layers. First, the input is processed by two stacked 2D CNNs with kernel (3, 3) and stride (2, 2). The output of the second CNN is then fed to a stack of two 2D Self-Attention [25], which process the input with 2D CNNs and compute attention along both matrix directions. All CNNs are followed by batch normalization [26] and ReLU nonlinearity. Additionally, we use a logarithmic distance penalty [16] in every self-attention layer in the encoder. Given a position $i$ in the query vector, and a position $j$ in the key vector, with $i \neq j$ we compute $pen = \log(|i - j|)$ and subtract $pen$ from the attention scores before softmax normalization.

The whole architecture is depicted in Figure 1, in which the audio processing (on the left-hand side) is visually divided by the Transformer's encoder and decoder.

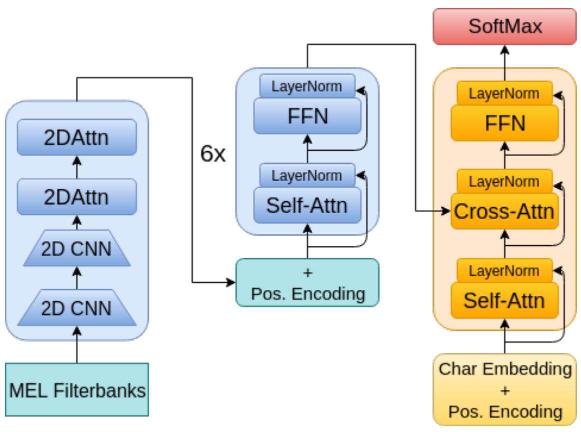

Figure 1: S-Transformer architecture for end-to-end speech translation. On the left-hand side, we have the 2D processing of the spectrogram. In the center, we find the usual Transformer encoder architecture with positional encoding and the 2 sub-layers of self-attention and FFN. On the right-hand side we have the Transformer decoder consisting of the 3 sub-layers of self-attention, cross-attention and FFN.

### 3.2. 2D self-attention

The output of the CNN layers is processed by a second bidimensional, long-ranged mechanism called 2D self-attention network (2DSAN) [25]. The input and the ouptut of the 2DSAN respectively have shape $(c_i, h, w)$ and $(c_o, h, w)$, where $c_i$ is the number of input channels and $c_o$ is the number of output channels. The input tensor is processed by three 2D CNNs, each with $m$ output channels, which generate the three tensors $Q$, $K$, $V$, respectively the query, the key, and the value of the attention [17]. Each of the $m$ output channels is used to compute a separate head of a multi-head attention. Then, the three tensors $Q$, $K$, $V$ are transposed and attention is also computed using the frequency dimension to compute similarity. Finally, the context vectors resulting from the attention along the two dimensions are concatenated along the channels dimension and the result is processed by a last 2D CNN with $c_o$ output channels. The data flow is depicted in Figure 2.

### 3.3. Data Tagging

The most common way to use synthetic data is to concatenate it with the clean data before starting the training procedure, as it is done with back-translations in MT [27]. However, recent research efforts [28] have shown that prepending a tag to all the source sentences from the synthetic dataset leads to better results as it makes the model able to recognize and better handle noisy and clean data. In the tagging experiments, we use one tag for the clean data and one for the synthetic data, learning an embedding for each. Instead of prepending the tag to the input sequence as in [28], its embedding is summed to all the elements of the sequence.

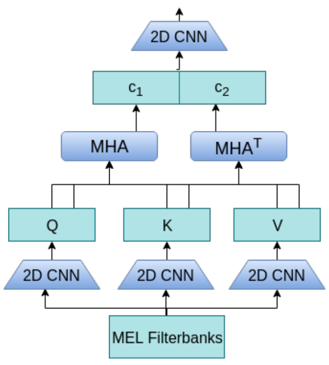

Figure 2: 2D attention. Green rectangles represent data, while gradient bleu blocks represent operations.

## 4. Experiments

We performed our experiments on the MuST-C train, validation and test splits. This way, we could compare the results of the present work with our published results on MuST-C [16]. For our final submission, we segmented the test set using a VAD trained with the LIUM diarization tool [29], which is the same used for last year's test set release.

In the encoder, our end-to-end SLT models use $3 \times 3$ 2D CNNs with stride $(2, 2)$ and 64 output channels, 4 heads in the 2D self-attention and output size of 64 units. The size of the Transformer layers is 512, while the hidden size of the feed-forward sub-layers is 1024. The Transformer layers in the decoder have the same size of the ones in the encoder. Both the encoder and the decoder have 6 Transformer layers. Translation is performed at character level, with a vocabulary for German of 176 entries. The final parameter count is about 33M parameters. It is a small model, in particular if we consider that the MT system has more than 250M parameters. All our models have been trained with the Adam optimizer [30] using an initial learning rate of $3 \times 10^{-4}$, which increases linearly to $1 \times 10^{-3}$ during 4000 warm-up steps and then decreases with inverse square root decay [17]. Dropout is set to 0.1 after every layer. As a further regularization, we train all our models with labeled-smoothed cross-entropy [31] with smoothing factor 0.1. The batch size is 4 segments for each GPU, but we delay the updates for 16 iterations in order to have large batch sizes. In the experiments with data tagging, we build batches by taking 4 segments from the pool of clean data, and 4 from the pool of synthetic data. As the two datasets are not of the same size, we oversample the smallest dataset (clean) in order to keep the training schedule consistent.

To prevent out-of-memory errors, we discarded all the training samples longer than 2000 steps in the source side Unless specified differently, we trained our models on 4 GPUs Nvidia 1080 with 12GB of RAM each. The codebase used for our experiments is FBK-Fairseq-ST[2], our adapta-

[2]The code is made available at the following link: https://github.com/mattiadg/FBK-Fairseq-ST

| Model | Test BLEU |
|---|---|
| (Di Gangi et al. 2019) | 17.3 |
| + Synth Data | 20.1 |
| + Tagged | **21.5** |

Table 1: Training with synthetic data vs. baseline using only MuST-C En-De.

| Model | Test Single | Test AVG |
|---|---|---|
| (Di Gangi et al. 2019) | 17.3 | 18.0 |
| + SpecAugment | 18.9 | 19.3 |
| + Synth Tagged | 22.2 | 22.7 |
| + Es Synth data | 22.3 | 22.9 |
| + FT NO SpecAugment | 22.5 | 22.3 |
| + FT long segments | 22.1 | 23.0 |

Table 2: Results with SpecAugment only, SpecAugment and synthetic data, and final fine-tuning without SpecAugment. The models used for checkpoint averaging are then used in our final ensemble decoding.

tion of fairseq [32] for end-to-end speech translation, which is written in PyTorch [33].

## 5. Results

In this section, we first evaluate the effect of data augmentation with synthetic data, then we add also SpecAugment. As during training we remove many training segments because of GPU memory issues, we also run a fine-tuning with a single GPU with larger memory in order to fit segments up to 8000 frames. Finally, we show the improvement obtaiend with checkpoint averaging and ensemble decoding.

### 5.1. Synthetic data

In Table 1, we compare our baseline (proposed in [16]) with identical systems trained also on synthetic data. The first system (`Synth Data`) simply concatenates the two sets of clean and synthetic data, while the second system (`Tagged`) tags clean and synthetic data differently. Surprisingly, although the two training data portions are extracted from the same pool of data (TED talks), and thus have a strong overlap, their simple concatenation obtains 2.8 BLEU points of improvement over the baseline. Tagging them differently leads to a further improvement of 1.4 points. These results indicate that end-to-end models can benefit significantly from the addition of new data, even if the reference is automatically generated. This observation paves the way for interesting approaches to data augmentation that we leave for future work.

### 5.2. SpecAugment

Table 2 shows the effect of training with `SpecAugment`. When comparing the baseline with a system that differs only for this aspect, we can observe an improvement of 1.6 BLEU points. When the training data is larger we observe a smaller improvement, but we still get $+0.7$ points over `Synth Tagged`. Finally, we have also experimented by translating the Spanish portion of MuST-C into German and adding it to the `Synth` data. However, with the further addition of this highly overlapping material, the improvement is marginal. Although the improvement with SpecAugment is limited, the results in ASR suggest that larger models can benefit more from the augmentation effect of this technique. In future work, we plan to train significantly larger models to explore in this direction

### 5.3. Translation of long sentences

In our experiments, we noticed that our systems struggle to translate long audio segments, for which the translation is often suddenly truncated. Trying to overcome this issue, we fine-tuned our models in a single GPU NVIDIA V100 with 16GB of memory and increasing the segment limit from 2,000 to 8,000. We did not perform training on such GPUs from scratch because we could access to only one GPU of this kind at a time, and training on 4 GPUs 1080 is faster. Unfortunately, although this procedure brought in 11,000 new audio segments that were discarded before, there were no visible BLEU score improvements (22.05) and we keep observing the truncation phenomenon. This is a problem that we consider worth investigating further.

### 5.4. Checkpoint averaging and Ensemble decoding

At the end of the training phase, for each training run we averaged the 5 checkpoints around the best one according to the validation loss. The results, which are reported in the last column of Table 2, show improvements for all models except for the one obtained by fine-tuning without SpecAugment. Finally, we performed ensemble decoding using all the models with the averaged checkpoints, and we achieved a score of 23.4 on the test set of MuST-C. This is the final configuration that we used for our IWSLT submission, which scored 15.8 BLEU points.

## 6. Conclusions

FBK's submission to IWSLT 2019 focused on the English-German end-to-end speech translation task. Our goal was to test the technology (S-Transformer) and data (MuST-C) built at FBK, in combination with state-of-the-art techniques for data augmentation. Our results show that data augmentation yields a significant improvement in translation quality, at the price of training an additional MT system. Since our models were constrained by GPU memory limitations, in future work we plan to exploit methods that allow training larger models, making a better use of larger datasets.

## 7. Acknowledgements

This work is part of a project financially supported by an Amazon AWS ML Grant. We thank Mauro Cettolo for the useful technical conversations.

## 8. References

[1] A. Bérard, O. Pietquin, L. Besacier, and C. Servan, "Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation," in *NIPS Workshop on end-to-end learning for speech and audio processing*, 2016.

[2] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-End Automatic Speech Translation of Audiobooks," in *Proceedings of ICASSP 2018*, Calgary, Alberta, Canada, April 2018.

[3] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on High-resource Speech Recognition Improves Low-resource Speech-to-text Translation," *Proceedings of NAACL 2019*, 2018.

[4] S. Bansal, H. Kamper, A. Lopez, and S. Goldwater, "Towards Speech-to-text Translation without Speech Recognition," in *Proc. of EACL*, 2017.

[5] D. Liu, J. Liu, W. Guo, S. Xiong, Z. Ma, R. Song, C. Wu, and Q. Liu, "The ustc-nel speech translation system at iwslt 2018," in *Proocedings of IWSLT*, 2018.

[6] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," in *Proc. of NAACL*, 2019.

[7] A. C. Kocabiyikoglu, L. Besacier, and O. Kraif, "Augmenting Librispeech with French Translations: A Multimodal Corpus for Direct Speech Translation Evaluation," in *Proceedings of LREC 2018*, Miyazaki, Japan, May 2018.

[8] M. Z. Boito, W. N. Havard, M. Garnerin, É. L. Ferrand, and L. Besacier, "Mass: A large and clean multilingual corpus of sentence-aligned spoken utterances extracted from the bible," *arXiv preprint arXiv:1907.12895*, 2019.

[9] R. Sanabria, O. Caglayan, S. Palaskar, D. Elliott, L. Barrault, L. Specia, and F. Metze, "How2: a large-scale dataset for multimodal language understanding," in *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018. [Online]. Available: http://arxiv.org/abs/1811.00347

[10] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 313–325, 2019.

[11] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-Resource Speech-to-Text Translation," *Proceedings of Interspeech 2018*, 2018.

[12] Y. Jia, M. Johnson, W. Macherey, R.-J. Weiss, Y. Cao, C.-C. Chiu, S.-L. Ari, and Y. Wu, "Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation," *ArXiv e-prints arXiv:1811.02050*, 2018.

[13] J. Pino, L. Puzon, J. Gu, X. Ma, A. D. McCarthy, and D. Gopinath, "Leveraging out-of-task data for end-to-end automatic speech translation," *arXiv preprint arXiv:1909.06515*, 2019.

[14] J. Niehues, R. Cattoni, S. Stüker, M. Cettolo, M. Turchi, and M. Federico, "The IWSLT 2018 Evaluation Campaign," in *Proceedings of IWSLT 2018*, Bruges, Belgium, October 2018.

[15] M. A. Di Gangi, R. Dessì, R. Cattoni, M. Negri, and M. Turchi, "Fine-tuning on clean data for end-to-end speech translation: FBK@ IWSLT 2018," *Proceedings of the 15th International Workshop on Spoken Language Translation (IWSLT 2018).*, 2018.

[16] M. A. Di Gangi, M. Negri, and M. Turchi, "Adapting Transformer to End-to-end Spoken Language Translation," in *INTERSPEECH*, 2019.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of NIPS 2017*, 2017.

[18] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[19] M. Cettolo, C. Girardi, and M. Federico, "Wit$^3$: Web inventory of transcribed and translated talks," in *Proceedings of the 16$^{th}$ Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[20] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri, "Findings of the 2019 conference on machine translation (WMT19)," in *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, 2019, pp. 1–61. [Online]. Available: https://www.aclweb.org/anthology/W19-5301/

[21] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus." in *Proc of LREC*, 2012.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[25] L. Dong, S. Xu, and B. Xu, “Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[26] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[27] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” in *Proc. of ACL 2016*, 2016.

[28] I. Caswell, C. Chelba, and D. Grangier, “Tagged Back-Translation,” in *Proceedings of the fourth Conference on Machine Translation (WMT 2019)*.

[29] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, “An open-source state-of-the-art toolbox for broadcast news diarization,” *Proceedings of Interspeech 2013*.

[30] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. of ICLR*, 2015.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. of CVPR*, 2016.

[32] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional Sequence to Sequence Learning,” in *Proceedings of ICML 2017*, Sydney, Australia, August 2017.

[33] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS 2017 Workshop Autodiff*.