Speechformer: Reducing Information Loss in Direct Speech Translation

Sara Papi [1,2] 🐘**, Marco Gaido**[1,2] 🐘**, Matteo Negri**[1]**, Marco Turchi**[1]
[1]Fondazione Bruno Kessler
[2]University of Trento
{spapi,mgaido,negri,turchi}@fbk.eu

Abstract

Transformer-based models have gained increasing popularity achieving state-of-the-art performance in many research fields including speech translation. However, Transformer's quadratic complexity with respect to the input sequence length prevents its adoption *as is* with audio signals, which are typically represented by long sequences. Current solutions resort to an initial sub-optimal compression based on a fixed sampling of raw audio features. Therefore, potentially useful linguistic information is not accessible to higher-level layers in the architecture. To solve this issue, we propose Speechformer, an architecture that, thanks to a reduced memory usage in the attention layers, avoids the initial lossy compression and aggregates information only at a higher level according to more informed linguistic criteria. Experiments on three language pairs (en→de/es/nl) show the efficacy of our solution, with gains of up to 0.8 BLEU on the standard MuST-C corpus and of up to 4.0 BLEU in a low resource scenario.

1 Introduction

Speech-to-text translation (ST) has been traditionally approached with cascade architectures consisting of a pipeline of two sub-components (Stentiford and Steer, 1988; Waibel et al., 1991): an automatic speech recognition (ASR), which transforms the audio input into a textual representation, and a machine translation (MT) model, which projects the transcript into the target language. A more recent approach consists in directly translating speech into target text using a single model (Bérard et al., 2016; Weiss et al., 2017). This direct solution has interesting advantages (Sperber and Paulik, 2020): *i)* it can better exploit audio information (e.g. prosody) during the translation phase, *ii)* it has lower latency, and *iii)* it is not affected by error propagation.

Thanks to these advantages, the initially huge performance gap with cascade systems has gradually closed (Ansari et al., 2020), motivating research towards further improvements.

Direct ST models are fed with features extracted from the audio with high frequency (usually every 10ms). This, on average, makes the resulting input sequence of vectors ∼10 times longer than the corresponding text, leading to an intrinsically redundant (i.e. long and repetitive) representation. For this reason, it is not possible to process speech data with a vanilla Transformer encoder (Vaswani et al., 2017), whose self-attention layers have quadratic memory complexity with respect to the input length. State-of-the-art architectures tackle the problem by collapsing adjacent vectors in a fixed way, i.e. by mapping a predefined number of vectors (usually 4) into a single one, either using strided convolutional layers (Bérard et al., 2018; Di Gangi et al., 2019; Wang et al., 2020a) or by stacking them (Sak et al., 2015). As a *positive* side effect, these length reduction solutions lower input redundancy. As a *negative* side effect, they disregard the variability over time of the amount of linguistic and phonetic information in audio signals (e.g. due to pauses and speaking rate variations) by giving equal weight to all features. In doing this, relevant features are penalized and considered equally important to the irrelevant ones, resulting in an information loss.

Recently, Salesky et al. (2019) obtained considerable translation quality gains by collapsing consecutive vectors with the same phonetic content instead of compressing them in a fixed way. Zhang et al. (2020) also showed that selecting a small percentage (∼16%) of input time steps based on their informativeness improves ST quality. On the downside, these approaches respectively require adding a model that performs phoneme classification and a pre-trained adaptive feature selection layer on top of an ASR encoder, losing the compactness of direct solutions at the risk of error propagation.

🐘 The authors contributed equally.

In direct ST, Liu et al. (2020) and Gaido et al. (2021) addressed the problem with a transcript/phoneme-based compression leveraging Connectionist Temporal Classification (CTC – Graves et al. 2006). However, since these methods are applied to the representation encoded by Transformer layers, the initial content-unaware downsampling of the input is still required for memory reasons, at the risk of losing important information.

To avoid initial fixed compression, we propose Speechformer: the first Transformer-based architecture that processes full audio content maintaining the original dimensions of the input sequence. Inspired by recent work on reducing the memory complexity of the attention mechanism (Wang et al., 2020b), we introduce a novel attention layer – the ConvAttention – whose memory requirements are reduced by means of convolutional layers. As the benefits of avoiding the initial lossy compression might be outweighed by the increased redundancy of the encoded audio features, we aggregate the high-level representation of the input sequence in a linguistically informed way, as in (Liu et al., 2020; Gaido et al., 2021). In other words, we collapse vectors representing the same linguistic atomic content (words, sub-words, pauses) into a single element, since they express the same linguistic information. The usage of the ConvAttention and of the linguistically motivated compression produces a considerably shorter, yet informative, sequence that fits the memory requirements of vanilla Transformer encoder layers. Experiments on three language directions (en→de/es/nl) show that the proposed architecture outperforms a state-of-the-art ST model by up to 0.8 BLEU points on the standard MuST-C corpus and obtains significantly larger gains (up to 4.0 BLEU) in a low resource setting where the amount of training data is reduced to 100 hours.

2 Model

In this section, we first introduce a novel attention layer that enables to process raw audio features without downsampling (§2.1). Then, we present an architecture that leverages this attention mechanism in the first encoder layers and reduces the redundancy of the more informative but longer resulting sequences with CTC compression (§2.2).

2.1 ConvAttention layer

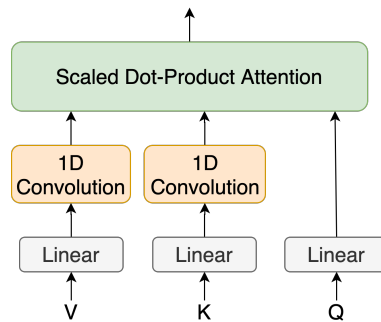State-of-the-art ST models employ convolutional neural networks to sample the feature sequence

Figure 1: Attention mechanism with the proposed convolutional compression of K and V.

to a lower dimension (typically by a factor of 4), enabling the use of Transformer layers otherwise impossible given their memory consumption. Outside ST, the Linformer architecture (Wang et al., 2020b) has been recently proposed to reduce the quadratic complexity of the product between the attention matrix (resulting from the product of the query – Q – and key – K – matrices) and the value (V) matrix by applying a linear projection to K and V. These projections bring the dimension of the sequence length of K and V to a fixed value, yielding a linear memory complexity. However, a direct application of this architecture to ST is problematic due to the high variability in audio lengths. On one side, mapping those sequences to a fixed dimension can cause an excessive information loss, with a consequent performance drop. On the other, it poses technical issues: the linear projection matrix has size $n \times k$, where n is the maximum input length and k is the fixed dimension. If the input has a length n' shorter than n, which is a common case due to the high variability in length of audio sequences, only the first n' weights of the matrix are updated. This results in gradients of different dimensions across GPUs, leading to training failures due to inconsistencies.

To avoid the aforementioned problems, we propose the adoption of ConvAttention (Figure 1), in which the linear projections of the Linformer architecture are substituted, both in K and V, with a single 1D convolutional layer. Hence, the length of the sequences used in the scaled dot-product attention depends on the stride of the convolution, a hyper-parameter we named *compression factor* (χ), which controls the memory complexity of the ConvAttention. Namely, being n the temporal dimension of K and V, the convolution output length is $\frac{n}{\chi}$ and the complexity of the ConvAttention is

$O((\frac{n}{\chi})^2)$, i.e. a $\frac{1}{\chi^2}$ factor lower than a vanilla Transformer self-attention. For instance, setting χ to 4 leads to the same memory consumption as standard ST models with an initial $\times 4$ subsampling (i.e. with two initial convolutional layers with stride 2).

Notice that the output sequence length is still equal to the input sequence length as it depends on the length of Q that is not modified.

2.2 Speechformer

The introduction of ConvAttention layers allows us to avoid sub-optimal fixed compressions that disregard the variability over time in the amount of audio information. However, since an encoder consisting only of ConvAttention layers does not compress the length of the original input sequence, the decoder will be fed with long and redundant sequences that are difficult to attend, leading to potential performance degradation.

To overcome this problem, as in (Liu et al., 2020; Gaido et al., 2021), we apply a content-informed compression to high-level hidden states trained using the CTC loss (Graves et al., 2006) to represent the linguistic content. Specifically, the CTC loss produces a prediction for each input time step and then merges equal predictions for consecutive time steps. The resulting sequence is compared with the reference, which is the sequence of subwords representing the transcript of the input utterance. CTC compression, similarly to the loss computation, collapses consecutive features corresponding to the same predictions, averaging them. After this operation, the sequence is reduced to a representation dimensionally closer to its textual content, which can be processed by the original attention mechanism without the need of approximations.

Speechformer (see Figure 2), is composed of E_L ConvAttention layers up to a CTC compression layer, after which there are E_T Transformer encoder layers. The E_L ConvAttention layers are meant to learn the linguistic content of the input audio while the E_T Transformer encoder layers are in charge of learning higher-level semantic representations, i.e. the encoder outputs, which the decoder has to convert into a text in the target language. We also maintain the two 1D convolutional layers before the ConvAttention layers but without striding, so that no sub-sampling is applied to the input. We make this choice both to keep the number of parameters comparable to the existing architectures, and to let the model learn a better representation of the
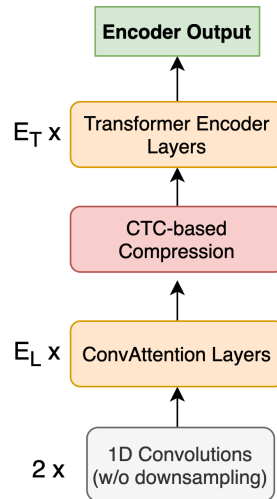

Figure 2: Speechformer architecture with E_L ConvAttention Layers and E_T Transformer Encoder Layers.

kernel	16	16	8	8	4
χ	16	8	8	4	4
BLEU	19.7	20.6	20.5	**21.3**	20.2

Table 1: BLEU on MuST-C en-de dev set varying the compression factor χ and 1D convolutional kernel size. The scores are obtained without label smoothing.

input before feeding it to the attention mechanism.

3 Experimental Settings

We experimented on three languages of MuST-C (Cattoni et al., 2021): English-German, English-Spanish, and English-Dutch. To ensure the reproducibility of our work, all training details are provided in the Appendix and the code – based on fairseq (Ott et al., 2019) – is released open source.[1]

Following (Wang et al., 2020b), we share the convolution parameters of the ConvAttention layers both among K and V and among the attention heads. We select the compression factor and the 1D convolution kernel size with a set of preliminary experiments on the en-de validation set. The compression factor (χ) is chosen among 4, 8, and 16, since 4 is the minimum value that avoids out-of-memory issues. The kernel size is set either equal to or twice as the value of χ. Table 1 shows that the combination of a compression factor of 4 and a kernel size of 8 leads to better performance compared to the other combinations. Consequently, in all our experiments we use this setting.

We initialize the ConvAttention weights of Speechformer with those of a pre-trained ST model

---

[1] https://github.com/sarapapi/ FBK-fairseq/tree/speechformer_emnlp2021.

Model	en-de				en-es				en-nl				Inference Time
	dev		tst-COMMON		dev		tst-COMMON		dev		tst-COMMON		
(Inaguma et al., 2020)	-		22.9		-		28.0		-		27.4		-
(Wang et al., 2020a)	-		22.7		-		27.2		-		27.3		-
Our baseline	22.5		22.8		31.2		27.9		24.2		27.2		1.0x
+ compression	22.3	-0.2	22.8	+0.0	31.1	-0.1	27.9	+0.0	24.2	+0.0	27.0	-0.2	0.9x
Plain ConvAttention	23.1*	+0.6	23.2	+0.4	31.5	+0.3	27.7	-0.2	24.8*	+0.6	26.9	-0.3	1.8x
Speechformer	**23.3***	**+0.8**	**23.6***	**+0.8**	**31.8***	**+0.6**	**28.5***	**+0.6**	**24.9***	**+0.7**	**27.7***	**+0.5**	1.3x

Table 2: BLEU score (average over 3 runs) on English→Dutch (en-nl), English→German (en-de), and English→Spanish (en-es) of MuST-C tst-COMMON (tst) and the dev (validation) set. The * symbol indicates statistically significant improvements over the baseline. Statistical significance is computed with a t-test (Student, 1908), whose null hypothesis is that the mean of the considered experiment is not higher than the mean of the baseline. We consider the result statistically significant if we can reject the null hypothesis with 95% confidence.

having only ConvAttention layers in the encoder, since, in the initial random state, the CTC-based compression might not properly reduce the input sequence, leading to out-of-memory issues in the following Transformer encoder layers. Notice that the pre-training does not improve performance. Indeed, Gaido et al. (2021) already showed that the encoder pre-training improves the baseline performance only without the additional CTC loss and that the results obtained by training without CTC loss and with encoder pre-training are identical to those achieved with the additional CTC loss. These findings have been confirmed in our experiments: *i)* initializing the encoder of the baseline with either an ASR or an ST encoder did not bring any improvement, and *ii)* our results are on par with those obtained with encoder pre-training and no additional CTC loss. We do not include the results with encoder pre-training of the baselines, as they do not bring any additional insight.

4 Results

We compare our proposed model to a strong *baseline* represented by a Transformer-based model with initial fixed sub-sampling (Wang et al., 2020a) and its *baseline+compression* variant that includes the average CTC compression strategy, as per (Gaido et al., 2021). We choose to also develop the second baseline to make the comparison with Speechformer fair since they both use the CTC compression strategy. Table 2 reports the results computed with SacreBLEU[2] (Post, 2018). For each experiment, we report the average over 3 runs to ensure that performance differences do not depend on the fluctuations of particularly good or bad runs.

First, it can be noticed that our baseline is in line with state-of-the-art architectures trained only

on MuST-C (Wang et al., 2020a; Inaguma et al., 2020). Second, the addition of CTC compression to the baseline model does not bring benefits. This confirms the findings of Gaido et al. (2021), who showed that applying CTC compression using transcripts produces differences in score that are not statistically significant. Speechformer, instead, results in statistically significant improvements over the baseline in all language directions, with BLEU gains ranging from 0.5 (for en-nl) to 0.8 (for en-de). As the CTC compression is not helpful for the baseline, we also evaluate a model (*Plain ConvAttention*) whose encoder is a stack of ConvAttention layers, i.e. without vanilla Transformer-encoder layers and any form of compression. The drop in performance with respect to Speechformer varies between 0.4 and 0.8 BLEU on all language pairs, supporting our hypothesis that a non-compressed encoder output is too redundant to be effectively attended by the decoder.

Low-Resource Settings. We suppose that the higher gains on en-de may be related to the size of the training data. Indeed, the en-de section of MuST-C used for training is the smallest one, containing 20% fewer data than the en-es section and 10% less than the en-nl one. Thus, we study Speechformer's performance in different data conditions by progressively reducing the amount of training data. For this analysis, we select the en-es section of MuST-C as it contains the highest number of hours (478h) among the three languages, and we experiment with three subsets, respectively containing 385h (corresponding to the amount of training data for en-de), 200h, and 100h (which can be considered a limited quantity given that the number of hours is respectively less than half and one fourth of the available data). Figure 3 shows that the gains obtained by Speechformer over the baseline do not vary significantly between 385h

---

[2]BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.0

and 478h (0.5 vs 0.6 BLEU). We can then conclude that the gain variation between en-de and en-es does not depend on the smaller size of the en-de training set. However, in the low resource settings (200h and 100h), the gains obtained by the Speechformer are much larger, amounting to 1.1 BLEU with 200h and 4.0 BLEU with 100h. To validate the robustness of these results, we also experimented on the en-de language pair and obtained consistent results: Speechformer outperforms the baseline by 1.5 BLEU (19.6 vs 18.1 BLEU) with 200h of training data and by 1.9 BLEU (9.7 vs 7.8 BLEU) with 100h of training data, achieving a considerable relative improvement of more than 24%. Although it brings consistent and significant gains in higher resource scenarios, these experiments show that Speechformer is particularly fruitful in low-resource settings. We leave to future work the assessment of the behavior of Speechformer in unrestricted data conditions (e.g. when using large ASR corpora to generate pseudo-labelled ST training data).
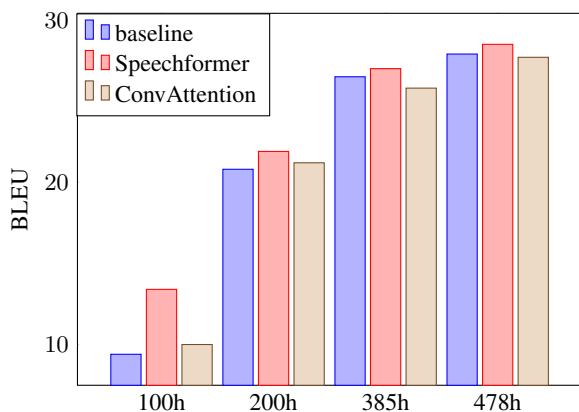

Figure 3: Architecture comparison varying the amount of en-es training data (478h, 385h, 200h, and 100h).

Inference Time. The ConvAttention layers process the whole input sequences, which are 4 times larger than those elaborated by the baseline attention mechanism. Thereby, a slow-down at inference time is expected, especially for the *Plain ConvAttention*, whose encoder layers are all ConvAttention layers. The last column of Table 2 confirms that the *Plain ConvAttention* architecture is 1.8 times slower than the baseline, i.e. the inference time is nearly twice. Speechformer is also slower than the baseline, but the overhead amounts to only 30% instead of 80%. Moreover, it can be noticed that the size of the attention matrix – and therefore the corresponding computational cost – can be

controlled in the Speechformer with the *compression factor* (χ) hyper-parameter. We leave to future studies the analysis of the trade-off between overall translation quality and inference time, which is usually irrelevant in offline ST, but becomes critical in simultaneous scenarios.

Manual Analysis. Lastly, we inspected the baseline and Speechformer outputs to better understand the reason behind the improvements brought by our architecture. This qualitative analysis was conducted on a sample of 200 sentences of the en-de test set – the language direction showing the largest gap between the systems (+0.8, see Table 2) – by a professional linguist with C2 German level. It emerged (see the Appendix for examples) that Speechformer tends to have better word-ordering, a typical problem arising when translating from an SVO language like English to an SOV language like German. Furthermore, Speechformer outputs display a better punctuation positioning – attributable to an improved handling of pauses and prosody – and a reduction of the number of audio misunderstandings and omissions. Together with the overall BLEU gains, these findings provide us with interesting hints about the potential of Speechformer.

5 Conclusion

In the wake of previous works showing the benefits of a content-informed compression over fixed downsampling of the audio features, we proposed Speechformer: the first ST Transformer-based model able to encode the whole raw audio features without any sub-optimal initial subsampling typical of current state-of-the-art models. Our solution is made possible by the introduction of a modified attention mechanism – the ConvAttention – that reduces the memory complexity to $O((\frac{n}{\chi})^2)$. As the plain application of ConvAttention layers leads to redundant sequences, high-level hidden states are compressed with a CTC-based strategy to obtain a compact, yet informative representation that can be processed by vanilla Transformer encoder layers. Experiments on three language pairs show that Speechformer significantly outperforms a state-of-the-art ST model by 0.5-0.8 BLEU, reaching a peak of 4 BLEU points in a low resource scenario.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019. A Comparative Study on End-to-end Speech to Text Translation. In *Proceedings of International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 792–799, Sentosa, Singapore.

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *Proceedings of ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Alberta, Canada.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. 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*, Barcelona, Spain.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Mattia A. Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. 2019. Enhancing Transformer for End-to-end Speech-to-Text Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 21–31, Dublin, Ireland. European Association for Machine Translation.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021. CTC-based Compression for Direct Speech Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 690–696, Online. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning (ICML)*, pages 369–376, Pittsburgh, Pennsylvania.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. ESPnet-ST: All-in-One Speech Translation Toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311, Online. Association for Computational Linguistics.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, New Orleans, Louisiana.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the Modality Gap for Speech-to-Text Translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proceedings of Interspeech 2019*, pages 2613–2617, Graz, Austria.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. 2015. Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition. In *Proceedings of Interspeech 2015*, Dresden, Germany.

Elizabeth Salesky, Matthias Sperber, and Alan W. Black. 2019. Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Florence, Italy. Association for Computational Linguistics.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online. Association for Computational Linguistics.

Frederick W. M. Stentiford and Martin G. Steer. 1988. Machine Translation of Speech. *British Telecom Technology Journal*, 6(2):116–122.

Student. 1908. The probable error of a mean. *Biometrika*, 6(1):1–25.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Las Vegas, Nevada, United States.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5998–6008, Long Beach, California.

Alex Waibel, Ajay N. Jain, Arthur E. McNair, Hiroaki Saito, Alexander G. Hauptmann, and Joe Tebelskis. 1991. JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, ICASSP 1991*, pages 793–796, Toronto, Canada.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. Fairseq S2T: Fast Speech-to-Text Modeling with Fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-Attention with Linear Complexity.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. In *Proceedings of Interspeech 2017*, pages 2625–2629, Stockholm, Sweden.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. Adaptive Feature Selection for End-to-End Speech Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online. Association for Computational Linguistics.

Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. 2021. Delving Deep Into Label Smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996.

A   Training Details

All our models are composed by 12 encoder layers and 6 decoder layers with 8 attention heads and are trained using label smoothed cross entropy (Szegedy et al., 2016) with the auxiliary CTC loss (Kim et al., 2017; Bahar et al., 2019) and Adam optimizer (Kingma and Ba, 2015). The number of parameters is ∼77M for the baseline and ∼79M for the Speechformer. The CTC is computed at the 8th encoder layer and its role is to predict the source transcription (lowercased and without punctuation), as in (Liu et al., 2020). The learning rate is set to 1e-3 with an inverse square-root scheduler and 10,000 warm-up updates. Mini-batches contain up to 5,000 tokens and we update gradients every 16 mini-batches. We apply SpecAugment (Park et al., 2019) and utterance-level cepstral mean and variance normalization. We filter out samples with duration exceeding 30s. The text is segmented in sub-word units with transcript and target Sentencepiece (Kudo and Richardson, 2018) unigram language models (Kudo, 2018) with size 5,000 and 8,000 respectively. We average 7 checkpoints around the best on the validation loss. Trainings were performed with 4 GPUs NVIDIA Tesla K-80 with 12GB of RAM and lasted about 3 days.

B   Output examples

Table 3 provides examples of the German translations generated by the baseline and by Speechformer for four utterances of the MuST-C test set, chosen among 200 sentences manually inspected by a C2 German speaker. These sentences have

(a) Word ordering	
Audio	It was a way that parents could figure out which were the right public **schools** for their **kids**.
Reference	Es ging um eine Methode, mit der Eltern herausfinden können, welche die richtigen öffentlichen **Schulen** für ihre **Kinder** sind.
Baseline	Es war eine Möglichkeit, dass Eltern herausfinden konnten, welche für ihre **Kinder** die richtige öffentliche **Schule** war.
	*It was an opportunity for the parents to find out which were for their **children** the right public **schools**.*
Speechformer	Es war eine Methode, mit der Eltern herausfinden konnten, welche die richtigen öffentlichen **Schulen** für ihre **Kinder** waren.
	*It was a method with which the parents could find out which were the right public **schools** for their **children**.*
(b) Punctuation handling	
Audio	**So, sir, can you help me?** I need help.
Reference	**Also, mein Herr, können Sie mir helfen?** Ich brauche Hilfe.
Baseline	Es ist also möglich, mir zu helfen.
	So it is possible to help me.
Speechformer	**Also, können Sie mir helfen?** Ich habe keine Hilfe.
	***So can you help me?** I have no help.*
(c) Audio misunderstanding	
Audio	You see Aluminum was the most valuable metal on the Planet, worth more than Gold and **Platinum**.
Reference	Aluminium war zu dieser Zeit das wertvollste Metall auf dem Planeten, wertvoller als Gold und **Platin**.
Baseline	Aluminium war die wertvollste Metallart auf dem Planeten, mehr als Gold und **Pflanzen**.
	*Aluminum was the most valuable type of metal on the planet, more than gold and **plants**.*
Speechformer	Aluminium war das wertvollste Metall auf dem Planeten, mehr als Gold und **Platin**.
	*Aluminum was the most valuable metal on the planet, more than gold and **platinum**.*
(d) Omission	
Audio	But the amazing thing about cities is they're worth so much more than **it costs** to build them.
Reference	Aber das Erstaunliche an Städten ist, dass sie so viel mehr wert sind, als **es kostet** sie zu bauen.
Baseline	Aber das Faszinierende an Städten ist, dass es viel mehr wert ist, als es zu bauen.
	But the fascinating thing about cities is that it's worth a lot more than building it.
Speechformer	Aber das Erstaunliche an Städten ist, dass sie viel mehr wert sind als sie **es kostet**, sie zu bauen.
	*But the amazing thing about cities is that they are worth a lot more than **it costs** to build them.*

Table 3: Examples of translation problems – *(a)*, *(b)*, *(c)* – and omissions – *(d)* – that Speechformer does not suffer from while baseline does.

been selected to highlight the specific aspects that are better handled by Speechformer.

Example *(a)* exhibits a wrong word ordering present in the baseline output, i.e. it anticipates "für ihre Kinder" (*for their kids*) with respect to "die richtigen öffentlichen Schulen" (*the right public schools*). Our proposed architecture, instead, translates the sentence in the correct order, making the translation easier to be read and understood.

Example *(b)* displays that Speechformer shows better punctuation handling, which – we hypothesize – is the result of an improved representation of prosody and pauses. In this example, for instance, our architecture is capable of detecting a question (i.e. *So can you help me?*) and translating it, while the baseline does not translate the input in question form and omits the last part of the audio content. Listening to the audio, we noticed a long pause after the question. We suppose that this pause led the baseline to conclude the sentence, while Speechformer managed to translate the remaining part of the utterance by going beyond that pause.

Our architecture shows an improved encoding of audio features that is reflected in its superior understanding of audio content. This emerges, indeed, from example *(c)*, where the word *Platinum* is correctly recognized and translated by our system, while the baseline misunderstands and translates it in another word, "Pflanzen" (*plants*), with a completely different meaning. The better audio understanding of Speechformer is present in example *(d)* as well. On the contrary, the baseline omits part of the original sentence (i.e. *it costs*), with a huge impact on the meaning of the resulting sentence, while Speechformer does not lose audio details and produces a complete translation. In this example, we can also notice that our system better solves pronominal resolution as it chooses *sie*, which follows the grammatical gender and number of *Staedten* (i.e. plural feminine), while the baseline uses *es*, which wrongly agrees with *das Faszinierende* (i.e. singular neuter).

C Effect of Label Smoothing

Label smoothing (Szegedy et al., 2016) is a widely adopted regularization factor (Zhang et al., 2021). As such, a more complex architecture that pro-

Model	en-de		en-es		en-nl	
baseline	21.2		26.2		25.5	
+ compression	21.2	+0.0	26.0	-0.2	25.1	-0.4
Plain ConvAttention	21.6	+0.4	25.6*	-0.6	25.6	+0.1
Speechformer	**22.3***	**+1.1**	**26.7***	**+0.5**	**26.2***	**+0.7**

Table 4: BLEU score on three language pairs of MuST-C tst-COMMON. The * symbol indicates statistically significant improvements over the baseline computed with bootstrap resampling (Koehn, 2004) with 10,000 samples, 1,000 sample size and 95% significance level.

cesses longer and potentially more redundant inputs – like our proposed Speechformer – can benefit more from its adoption. Hence, to validate that our gains are not due to a better regularization of the models and to assess the effect of label smoothing, we run experiments using the cross entropy loss without smoothing factor. The results are reported in Table 4. Compared with the scores reported in Section 4 of the paper, we can see that label smoothing brings significant gains for all the systems (ranging from 1.5 to 2.0 BLEU points). Most importantly, the improvements of the Speechformer architecture (0.5-1.1 BLEU) are similar to those achieved with label smoothing (0.5-0.8 BLEU). The minimal difference can be explained by statistical variations of the results, considering that those obtained without label smoothing are computed on a single run. We can conclude that these results confirm the efficacy of our architecture and the validity of our experiments, showing that they are not biased by a higher regularization that might favor our solution over the baseline.