# Enhancing Transformer for End-to-end Speech-to-Text Translation

**Mattia A. Di Gangi**[1,2] **Matteo Negri**[1] **Roldano Cattoni**[1] **Roberto Dessì**[2*] **Marco Turchi**[1]

[1]Fondazione Bruno Kessler
via Sommarive, 18, Povo, TN, Italy
surname@fbk.eu

[2]Università degli Studi di Trento
CIMeC, DISI
name.surname@unitn.it

## Abstract

Neural end-to-end architectures have been recently proposed for spoken language translation (SLT), following the state-of-the-art results obtained in machine translation (MT) and speech recognition (ASR). Motivated by this contiguity, we propose an SLT adaptation of Transformer (the state-of-the-art architecture in MT), which exploits the integration of ASR solutions to cope with long input sequences featuring low information density. Long audio representations hinder the training of large models due to Transformer's quadratic memory complexity. Moreover, for the sake of translation quality, handling such sequences requires capturing both short- and long-range dependencies between bidimensional features. Focusing on Transformer's encoder, our adaptation is based on: *i)* downsampling the input with convolutional neural networks, which enables model training on non cutting-edge GPUs, *ii)* modeling the bidimensional nature of the audio spectrogram with 2D components, and *iii)* adding a distance penalty to the attention, which is able to bias it towards short-range dependencies. Our experiments show that our SLT-adapted Transformer outperforms the RNN-based baseline both in translation quality and training time, setting the state-of-the-art performance on six language directions.

## 1 Introduction

Neural encoder-decoder models (Sutskever et al., 2014) with attention (Bahdanau et al., 2015) is a general architecture that, by enabling to tackle sequence-to-sequence problems with a single end-to-end model, achieved state-of-the-art results on machine translation (MT) (Bentivogli et al., 2016; Gehring et al., 2017; Vaswani et al., 2017; Chen et al., 2018) and obtained increasingly good performance in automatic speech recognition (Chan et al., 2016; Chiu et al., 2018; Zhang et al., 2017; Zeyer et al., 2018; Dong et al., 2018). The advantages of end-to-end techniques, besides their conceptual simplicity, reside on the prevention of error propagation, and a reduced inference latency. Error propagation is particularly problematic for the SLT task (Ruiz et al., 2017), in which MT would be significantly penalized by errors resulting from the previous ASR processing step. For this reason, end-to-end solutions have been recently proposed (Bérard et al., 2016; Weiss et al., 2017; Anastasopoulos and Chiang, 2018; Liu et al., 2018; Di Gangi et al., 2018) but, in terms of performance, they are still far behind the pipeline approach. The reason of the worse performance for this task can be found in its intrinsic difficulty, as it inherits and combines the challenges of the two pipelined tasks. Indeed, SLT models map audio features into words, like in ASR, but the input is mapped into text in a different target language, like in MT. Thus, the problems of word reordering and ambiguous meaning typical of translation are combined with the ambiguity of speech signal and speaker variety. One possible approach to deal with this task is to start from an MT solution and adapt it to speech input. Transformer (Vaswani et al., 2017) is an encoder-

[*]Work done during a summer internship at the Machine Translation Research Unit at Fondazione Bruno Kessler.

© 2019 The authors. This article is licensed under a Creative Commons 4.0 licence, no derivative works, attribution, CC-BY-ND.

decoder architecture based on self-attention networks (SAN, (Cheng et al., 2016)) that, because of its strong results, is the most popular architecture in MT, and is now used as a base for many NLP tasks (Devlin et al., 2018). While LSTMs are known to require long trainig time (Lei et al., 2017; Di Gangi and Federico, 2018; Kalchbrenner et al., 2016), Transformer reduces the training time by performing parallel computation along all the time steps, similarly to convolutional neural networks (CNNs). Despite the appealing advantages, the research on end-to-end SLT has focused so far on recurrent architectures, and only big industrial players have been able to train networks with many layers, many parameters, and additional synthetic data (Jia et al., 2018). In fact, for computational and modeling reasons, the application of SANs to speech input has to face additional challenges compared to handling textual data. In particular, these include:

1. SANs have a memory complexity that is quadratic in the sequence length. From a computational perspective, this becomes a problem when the input is an audio signal, which is typically represented as a very long sequence of log-filter-banks. For the same utterance, this type of input is considerably longer than the corresponding textual representation fed to MT encoders.

2. The bidimensional dependencies along the time and frequency dimensions in the spectrogram (Li et al., 2016). This 2-dimensional representation is more difficult to handle compared to the 1-dimensional input representation (i.e. along the time dimension only) processed by MT encoders.

3. The absence of an explicit bias towards the local context. Differently from MT, modeling long-range dependencies between words is logically preceded, as the input is unsegmented, by modeling short-range dependencies between time-frames belonging to the same linguistic constituents (Sperber et al., 2018).

Focusing on these problems, in this paper we explore different adaptations of Transformer to the end-to-end SLT task. Initially, we show that *as-is* and with a comparable number of parameters, Transformer is not competitive with LSTM models. In order to investigate the reasons of its lower performance, we posit that the problem lies in the inability of the Transformer encoder to properly model long audio input. This hypothesis is checked by switching the encoders and decoders of the Transformer and LSTM architecture, which results in better performance when the Transformer decoder is preceded by the LSTM encoder. These results inform and motivate our enhancements to the Transformer architecture. To this aim, we proceed incrementally showing, through comparative experiments, that:

1. Sequence compression with CNNs and downsampling enables effective audio encoding while allowing to train the system even on single GPUs;

2. Modeling 2D dependencies produces more stable and better results;

3. Biasing the encoder self-attention with a distance penalty improves translation quality.

Our experiments are run on different datasets covering different languages. First, we evaluate our architecture on two relatively small corpora: Augmented Librispeech (Kocabiyikoglu et al., 2018) for English→French and IWSLT 2018 for English→German. Then, we broaden the language coverage through experiments with MuST-C (Di Gangi et al., 2019),[1] a large multilingual SLT dataset recently released. This allows to validate our findings on six language directions (En-De/Es/Fr/Pt/Ro/Ru).

Overall, our evaluation indicates that the proposed SLT-oriented adaptation of Transformer results in a model that significantly outperforms a strong end-to-end system both in translation quality and training speed. For the sake of results' replicability the code developed for the experiments described in this paper can be downloaded at `http://github.com/mattiadg/FBK-Fairseq-ST`.

## 2 Related works

Our work has been influenced by the recent works on end-to-end SLT, as well as the applications of SANs to the task of ASR.

**End-to-end SLT.** The first encoder-decoder architecture based on LSTM was introduced for SLT by Bérard et al. (2016) showing the feasibility of

[1] `http://mustc.fbk.eu`

directly translating from the audio signal. Weiss et al. (2017) enhanced this approach by exploring settings with different numbers of layers in encoder and decoder and testing various multitask learning strategies. Bérard et al. (2018) trained a single model to translate English audiobooks into French and shown that pre-training the encoder on ASR data improves the final result. All these works showed that the input sequence length has to be reduced to work with recurrent models. To cope with the lack of end-to-end data, different directions have been evaluated. For instance, (Anastasopoulos and Chiang, 2018) and (Weiss et al., 2017) performed analyses of different multitask settings to leverage more data. Bansal et al. (2018) shown that the pre-training of the encoder is also helpful when performed on a different language, in particular when the source language is low–resourced. (Jia et al., 2018) increased the training data by using a large quantity of synthetic data that results in an end-to-end system able to outperform the cascade model. Their architecture still relies on LSTMs. Transformer has been applied to this task (Vila et al., 2018) using only a small training set and taking advantage of the computational power of TPUs. Differently from these works, we enhance the Transformer architecture to be trained on GPUs, in shorter time compared to LSTM models, and without using multi-task learning.

**Self-attention for ASR.** Given the results of Transformer in MT, recent works on ASR proposed SANs for both acoustic modeling (Sperber et al., 2018; Povey et al., 2018) and end-to-end ASR (Dong et al., 2018; Zhou et al., 2018a; Zhou et al., 2018b). Some works trained Transformer for (multilingual) ASR with little modification to its architecture (Zhou et al., 2018a; Zhou et al., 2018b), showing the feasibility of this approach in terms of results. Dong et al. (2018) proposed the Speech-Transformer for end-to-end ASR with the goal of encoding efficiently an effectively audio input. They rely on CNNs to reduce the sequence length, and propose 2D self-attention to capture the dependencies in the two dimensions of a spectrogram (Li et al., 2016) that are out of the range of CNNs. In this paper we show that only Speech-Transformer is not enough to outperform an LSTM-based model on end-to-end SLT, because the lack of an explicit bias towards local context seems to be harmful for SANs when applied to audio input. In ASR to address a similar problem, Povey et al. (2018) use hard masking to force the self-attention into a local context, while Sperber et al. (2018) use a Gaussian distance penalty to reduce the attention weights according to the distance between input elements. Though effective, the results of this distance penalty are highly dependent on the initial value of the Gaussian variance. Our work tests, for the first time, the distance penalty in the task of SLT and proposes a penalty function that, without additional hyperparameters, allows the Transformer model to outperform the LSTM architecture.

## 3 Background

Sequence-to-sequence models map a variable-length source sequence into a variable-length target sequence. They are usually composed of three conceptual blocks. An *encoder* maps an input sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ of n time steps into a hidden representation $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n'})$ of contextualized vectors, where $n'$ can be different from $n$. A *decoder* generates a target sequence of tokens $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$ in an autoregressive manner. The connection between encoder and *decoder* is given by one or multiple *attentions* that weight the elements of $\mathbf{H}$ according to their relevance for the current decoder time step. Such a network is trained by minimizing the cross-entropy between the probability distribution of the target tokens estimated by the network, and the gold labels:

$$L(\theta) = \sum_{i=0}^{m} P(\tilde{\mathbf{y}}_i = \mathbf{y}_i | \mathbf{X}, \mathbf{y}_{<i}; \theta) \qquad (1)$$

In this paper, $\mathbf{X}$ is a sequence of audio spectrogram frames, while $\mathbf{Y}$ is a sequence of characters in the target language.

Two encoder-decoder architecture that are relevant for this work are: the recurrent model for end-to-end SLT proposed in (Bérard et al., 2018), which is based on LSTM and CNNs, and the Transformer, as proposed for MT.

### 3.1 End-to-end SLT

Bérard et al. (2018) proposed a recurrent sequence-to-sequence architecture for SLT based on LSTMs. The encoder receives an input in the form of sequences of Mel-filterbanks. The input is first projected to a larger space with two affine transformations, each followed by ReLU activation. The expanded input is then reduced by a factor of 4 with two following strided 2D convolutions. Finally, the resulting tensor is linearized and processed as a sequence by three stacked bi-directional LSTMs.

The average of the encoder outputs along the time dimension is used to initialize the first of two LSTMs in the decoder. The output of the first LSTM is used by an attention network to compute a context vector of the source, which is fed as input to the second LSTM. The output of the second LSTM is used to compute the target probabilities and also as a hidden state for the first LSTM (deep transition LSTMS (Pascanu et al., 2014)). Hencefort, we will refer to this approach as **CNN+LSTM**.

### 3.2 Transformer

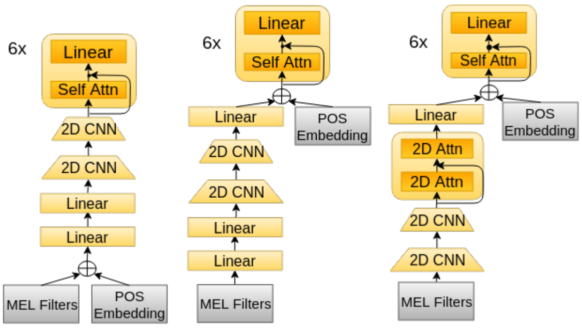
Transformer (Vaswani et al., 2017) is an encoder-decoder architecture entirely based on attention networks. Given three sequences $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ the attention computes a context vector $d_i$ for each query time step $i$ ($\mathbf{Q}_i$) that is a weighted average of the values $\mathbf{V}$, where the weights are computed as a normalized score of the similarity between $\mathbf{Q}_i$ and all the key values $\mathbf{K}$:

$$d_i = \text{softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_{\text{model}}}) \cdot \mathbf{V} \qquad (2)$$

where $\sqrt{d_{\text{model}}}$ is a constant scaling factor based on the layer size $d_{\text{model}}$. The core component of Transformer is the multi-head attention (MHA), a network that, given two input sequences $\mathbf{a}, \mathbf{b}$ computes attention between $\mathbf{a}$ and $\mathbf{b}$ in multiple, parallel branches. MHA is used to model dependencies both between encoder and decoder ($\mathbf{K}, \mathbf{V} = \mathbf{a}$ and $\mathbf{Q} = \mathbf{b}$), and within the two networks (self-attention, $\mathbf{K}, \mathbf{V}, \mathbf{Q} = \mathbf{a}$). As it is shown in Equation 2, MHA is fully content-based and, as such, it is position invariant. The positional information within the sequence is conveyed by summing the vector content with a fixed positional encoding based on trigonometric functions. Another relevant property of the MHA is the possibility to compute it in parallel for all the time steps in both $\mathbf{Q}$ and $\mathbf{K}$, as well as for all the multiple heads, but this comes at the cost of a quadratic memory complexity.

## 4 SLT Transformer

The application of Transformer to speech input is not trivial because of *i)* computational issues that hinder its use; and *ii)* modeling limitations that harms its performance. The first issue to overcome is the quadratic GPU memory occupation of Transformer, which is particularly relevant on speech because the sequences are order of magnitudes larger than in text. On the modeling side, Transformer's performance is limited by the absence of a bias to capture short-range dependencies along time (Sperber et al., 2018; Povey et al., 2018), as well as the 2D joint dependencies over the time and frequency dimensions that characterize a spectrogram (Li et al., 2016). Strided 2D CNNs can compress the input sequence while also modeling 2D dependencies. However, the resulting sequences are still much longer than an equivalent text sequence, and thus we propose a distance penalty to enforce the modeling of short-range dependencies.



(a) B-Transformer (b) R-Transformer (c) S-Transformer

**Figure 1:** Three Transformer encoders for SLT. Components in grey are non-learnable.

### 4.1 Encoding with 2D CNNs

In this section, we propose three variants of Transformer. B- and R-Transformer replace the LSTM layers in CNN+LSTM with Transformer encoder layers and differ in their use of the positional encoding. S-Transformer is a further improvement of R-Transformer that adds to the encoder the capability of modeling 2D dependencies in the input data. In all the three variants, the adaptations regard only the layers preceding the Transformer encoder. The following Transformer encoder and decoder stacks are left unchanged.

**B-Transformer** (Figure 1a). Our baseline model uses the same encoder as CNN+LSTM (Bérard et al., 2018) but replaces the LSTM layers with Transformer encoder layers. The replacement of LSTMs makes the encoder position invariant, and thus the sequential order is conveyed by summing the positional encoding directly to the input fea-

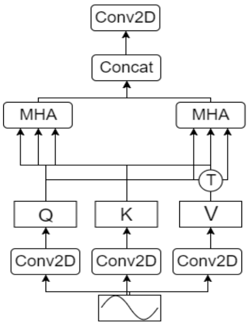


**Figure 2:** Schematic representation of 2D self-attention.

tures.[2]

**R-Transformer** (Figure 1b). As the positional encoding and the input are both fixed vectors, we propose to sum the positional encoding right before the Transformer encoder (the part of the network that requires a positional information). The sum is preceded by a linear transformation of the CNN output followed by ReLU non-linearity, whose goal is to transform its input into a space where the fixed positional encoding can be more effective.

**S-Transformer** (Figure 1c). Our second improvement follows the idea of modeling 2D joint dependencies in the input signal by applying a stack of 2D components to the input (Dong et al., 2018). The first two CNNs capture local 2D-invariant features (Amodei et al., 2016) in the input, while the following two 2D self-attention layers (Figure 2) model long-range context (Dong et al., 2018). The 2D self-attention computes the three tensors $\mathbf{K}, \mathbf{Q}, \mathbf{V}$ with three parallel 2D CNNs of its input with $c$ output channels. Each of the $c$ channels is used as an attention head in an MHA network. $\mathbf{K}, \mathbf{Q}$ and $\mathbf{V}$ are used to compute the attention over the temporal dimension as in Equation 2. Then, the three matrices are transposed and another MHA is computed over the frequency dimension. Finally, the $2c$ channels from the two MHAs are concatenated and processed by an additional $2D$ CNN with $n$ output channels. The 2D attentions enrich the encoder representation by modeling 2D dependencies that cannot be captured by CNNs.

### 4.2 Distance Penalty

To further improve the encoder capability of modeling short-range dependencies, we introduce, besides CNNs, a distance penalty mechanism in the encoder self-attention. This mechanism biases the network towards the local context without imposing hard constraints that would prevent it from finding long-range dependencies. The attention computation (Equation 2) is modified as follows:

$$\mathbf{c} = \text{softmax}(\mathbf{QK^T}/\sqrt{d_{\text{model}}} - \pi(\mathbf{D}))\mathbf{V} \quad (3)$$

where $\mathbf{D}$ is a matrix containing, in each cell $d_{i,j}$, the position distance $|i - j|$, and $\pi$ is a distance penalty function.

In this paper, we experiment with distance penalty computed with two different functions. The Gaussian penalty introduced in (Sperber et al., 2018) computes a Gaussian-shaped penalty distribution with a distinct learnable variance $\sigma$ for each head in the MHA as follows:

$$\pi_G(d) = \frac{(d)^2}{2\sigma^2} \quad (4)$$

This function gives to a network the flexibility to shrink or extend the attention span in each attention head. In this way, the network can extract different features from different heads in a layer, but also in different layers. Indeed, in (Sperber et al., 2018) only the first layer restricts its attention span in the best setting. The downside of this approach is that the initial value of the variance is an additional hyperparameter that highly affects the performance. In order to eliminate this additional hyperparameter, we propose to use a logarithmic function as a distance penalty:

$$\pi_{\log}(d) = \begin{cases} 0, & \text{if d = 0} \\ \log_e(d), & \text{else} \end{cases} \quad (5)$$

The logarithm biases the network towards the local context but the penalty grows slowly with distance, and thus it does not impede the modeling of global dependencies.

## 5 Experiments

We run our experiments on three SLT datasets, of which two comprise a single language direction and one comprises 6 language directions. In all cases, English is the source language.

**Monolingual datasets.** The first one monolingual corpus is built from material released for the IWSLT evaluation campaigns, namely the En→De training data from IWSLT 2018 (Niehues et al.,

[2] Due to its high GPU memory occupation, we could not train a baseline Transformer (comparable in size to the other models used for experiments) without input compression.

| Corpus | Hours | Train | Valid | Test |
|---|---|---|---|---|
| **IWSLT** (En-De) | 273 | 171K | 1000 | 1000 |
| **Librispeech** (En-Fr) | 236 | 95K | 1071 | 2048 |
| **Multilingual** | | | | |
| En-De | 408 | 234K | 1423 | 2641 |
| En-Es | 504 | 270K | 1316 | 2502 |
| En-Fr | 492 | 280K | 1412 | 2632 |
| En-Pt | 385 | 210K | 1367 | 2502 |
| En-Ro | 432 | 240K | 1370 | 2556 |
| En-Ru | 489 | 270K | 1317 | 2513 |

**Table 1:** Data statistics for IWSLT, Librispeech and our multilingual corpus. Train, Valid and Test are numbers of sentence pairs.

2018) and the test data from IWSLT 2014 (Cettolo et al., 2014).[3] The second dataset is the Augmented Librispeech corpus (Kocabiyikoglu et al., 2018) that is produced using English audiobooks of novels, and their translations into French.

**Multilingual dataset.** We have recently developed a large corpus from English TED talks, called MuST-C (Di Gangi et al., 2019). Unlike IWSLT and Librispeech, MuST-C covers multiple language directions (En→De/Es/Fr/Pt/Ro/Ru). We built it following the alignment-based approach proposed in (Kocabiyikoglu et al., 2018) and using English speech recordings and their translations available on the TED talks website.[4] For each target language, we aligned text in English and in the target language using the Gargantua toolkit (Braune and Fraser, 2010), then we aligned the resulting English sentences with the corresponding audio using Gentle,[5] a forced-aligner based on the Kaldi toolkit (Povey et al., 2011). In order to improve the alignment quality we performed two successive steps of filtering. In the first step, we removed all the talks where at least 15% of the words have not been recognized by Gentle. In the second step, we removed from the remaining talks all the sentences with no recognized words. For replicability of results, the corpus is released with a predefined train, validation and test split. The corpora statistics are listed in Table 1 and show that each language direction of MuST-C is considerably larger than the other 2 corpora.

**Experimental setup.** For a fair comparison of the different architectures, we first set the parameters of the recurrent baseline (CNN+LSTM, §3.1) similar to what reported in (Bérard et al., 2018). Then, we adjust the Transformer to have a number of parameters similar to the recurrent one (∼9.5M). The CNNs have a $3 \times 3$ kernel and $16$ output filters. The LSTMs in the baseline have a hidden size of $512$, with $3$ layers in the encoder and $2$ in the decoder. The initial encoder states are learnable parameters, while the initial decoder state is computed as the mean of the encoder states. We found the learnable encoder states to be critical to reach convergence. The Transformer models have $6$ layers in both encoder and decoder, with layer size of $256$, hidden size of $768$ and $4$ heads in multi-head attention. To further asses the performance of our models, we also experiment with a BIG version with more parameters, featuring layer size $512$, hidden size $1024$, and $8$ heads. set dropout to $0.2$ for CNN+LSTM and $0.1$ for Transformer. No dropout is applied in the recurrent connections. Training is performed using the Adam optimizer (Kingma and Ba, 2015) with learning rate $0.001$ for LSTM and $0.0002$ for Transformer. The learning rate is kept fixed for Transformer for the sake of a fair comparison with the baseline. B-Transformer serves as a baseline to evaluate the impact of the proposed adaptations. We train our R- and S-Transformer models with and without distance penalty, either Gaussian or logarithmic. We test all these configurations on the IWSLT and Librispeech corpora. Then, due to the higher number of directions in the multilingual corpus, we only run experiments on it with the best-performing system. Following (Bansal et al., 2018; Bérard et al., 2018), we first train a model with the ASR part of each corpus and then we use it to initialize the weights of the SLT encoder. All the experiments are run on a single GPU Nvidia 1080 Ti with 12G of RAM, and the code used for all the experiments is based on Pytorch (Paszke et al., ).

**Data processing and evaluation.** 40-dimensional MFCC filter-banks were extracted from the audio signals of each dataset using window size of 25 ms and step size 10 ms. The frame energy feature was additionally extracted from the LibriSpeech audio, similarly to (Bérard et al., 2018). All texts were tokenized and split into characters. Performance is evaluated with BLEU (Papineni et al., 2002) at token level after aggregating the output characters into words.

[3] We could not use the IWSLT 2018 test data, because the gold standard has not been released.

[4] http://www.ted.com – dump of April 2018

[5] `github.com/lowerquality/gentle`

| **Librispeech** | Enc | Dec | BLEU ↑ |
|---|---|---|---|
| *CNN+LSTM* | - | - | 10.7 |
| | ✓ | - | **13.2** |
| | ✓ | ✓ | 13.0 |
| *B-Transformer* | - | - | 6.3 |
| | ✓ | - | 9.0 |
| | ✓ | ✓ | **9.5** |
| **IWSLT** | | | |
| *LSTM* | - | - | 8.5 |
| | ✓ | - | **9.2** |
| | ✓ | ✓ | 7.5 |
| *B-Transformer* | - | - | **7.9** |
| | ✓ | - | 7.3 |
| | ✓ | ✓ | 5.9 |

**Table 2:** Speech translation results for the Librispeech and IWSLT corpora wuth our two baseline models. A checkmark on Enc (Dec) means that the encoder (decoder) has been pre-trained.

| **Enc / Dec** | LSTM | Transformer |
|---|---|---|
| LSTM | 13.2 | 11.9 |
| Transformer | 8.2 | 9.0 |

**Table 3:** Mixed-architecture experiments on Librispeech.

## 6 Results and Discussion

### 6.1 Baseline.

As a first step, we want to evaluate our baseline B-Transformer against CNN+LSTM to understand the effectiveness 2D convolutional compression with Transformer. We ran the experiments with no pre-training, by pre-training only the encoder or both encoder and decoder. As can be seen in Table 2, the best results with CNN+LSTM are obtained by pre-training only the encoder, while for B-Transformer the training is more unstable and this is reflected also in the results. Considering the results of CNN+LSTM and the relatively good results of B-Transformer when pre-training only the encoders, we decided to follow this practice in all the following experiments. When considering only the results with the pre-trained encoder, CNN+LSTM outperforms B-Transformer by 4 BLEU points on Librispeech and 2.1 BLEU points on IWSLT. To better understand the source of degradation for the B-Transformer, we performed an experiment switching encoder and decoder between the two architectures with pre-trained encoder (table 3) and evaluated them on Librispeech. When using CNN+LSTM encoder, the Transformer decoder causes a degradation of 1.3 BLEU points, while having Transformer encoder and LSTM decoder causes a degradation of 5 points over CNN+LSTM. Given these results, the following experiments all focus on enhancing the B-Transformer encoder. Despite the poor translation quality, exploring the Transformer is still interesting because of its reduced training time (listed on Table 4), which is reduced by a factor of 2 on IWSLT (67K vs 112K seconds) and even more on Librispeech (101K vs 248K seconds). These results show that input compression makes the training of Transformer feasible for SLT, but it does not result in immediate improvements over LSTMs.

| **Librispeech** | BLEU ↑ | Time (s) | Time/Ep. |
|---|---|---|---|
| CNN+LSTM | 13.2 | 248K | ∼ 2.8K |
| B-Transformer | 9.0 | 101K | ∼ 0.69K |
| R-Transformer | 11.5 | 72K | ∼ 0.73K |
| - Gauss penalty | 12.5 | 82K | ∼ 0.75K |
| - log penalty | 12.3 | 64K | ∼ 0.75K |
| S-Transformer | 12.5 | 76K | ∼ 0.79K |
| - Gauss penalty | 13.8 | 88K | ∼ 0.86K |
| - log penalty | 13.5 | 76K | ∼ 0.86K |
| **IWSLT** | BLEU ↑ | Time (s) | Time/Ep. |
| CNN+LSTM | 9.2 | 112K | ∼ 2.9K |
| B-Transformer | 7.1 | 67K | ∼ 1.0K |
| R-Transformer | 9.8 | 92K | ∼ 1.0K |
| - Gauss penalty | 10.8 | 101 K | ∼ 1.1K |
| - log penalty | 10.5 | 93K | ∼ 1.1K |
| S-Transformer | 9.8 | 89K | ∼ 1.1K |
| - Gauss penalty | 10.8 | 90K | ∼ 1.2K |
| - log penalty | 10.6 | 81K | ∼ 1.2K |

**Table 4:** Results on the Librispeech and IWSLT 2014 test set. Differences wrt the baseline (CNN+LSTM) are statistically significant (randomization test, p=0.05).

### 6.2 Encoder Enhancements

In the following, we discuss the results obtained with our enhancements to the Transformer encoder, i.e. modify the use of position encoding, model 2D dependencies with CNNs and 2D self-attention, and insert a distance penalty to the encoder self-attention.

R-Transformer differs from B-Transformer in the layer where the position encoding is summed to the input. As can be seen in Table 4, this detail is very relevant as R-Transformer improves over B-Transformer by more than 2.5 BLEU points in both datasets with less training time. However, it is significantly worse than CNN+LSTM on Librispeech ($-1.7$ BLEU points) and slightly better on IWSLT ($+0.6$).

The next step is to evaluate the enhancements in modeling 2D input proposed in S-Transformer. Its results are 1.0 BLEU point better than R-Transformer in Librispeech, and equal on IWSLT, while having a similar parameter count and con-

| **Initial variance** | Librispeech | IWSLT |
|---|---|---|
| 5.0 | 13.8 | 10.8 |
| 100.0 | 13.1 | 10.9 |

**Table 5:** Results with different values of initial variance for Gaussian penalty and S-Transformer.

vergence time. Despite the improvement, S-Transformer is 0.7 points less than CNN+LSTM on Librispeech.

In Table 4 we show the results obtained using the distance penalties introduced in §4.2 to model short-range dependencies in the Transformer encoder. Distance penalties produce performance improvements for R- and S-Transformer that range from 0.7 to 1.3 BLEU points, with the Gaussian penalty (initial variance = 5.0) being 0.2 $\sim$ 0.3 BLEU points better than the logarithmic one. S-Transformer with Gaussian penalty obtains the best results in both corpora, with improvements of +0.6 and +1.6 BLEU points over CNN+LSTM on, respectively, Librispeech and IWSLT. The results with Gaussian penalty are computed using initial variance (for the ASR training) of 5.0. Using an initial variance of 100.0 (the value recommended in the work by Sperber et al. (2018)) we obtained a significant degradation on Librispeech with a BLEU of 13.1 and a comparable result on IWSLT with 10.9 (Table 5). These results show that biasing the self-attention with a distance penalty is critical to obtain competitive translation quality with Transformer and also outperform the strong CNN+LSTM baseline.

### 6.3 Gaussian variances

Sperber et al. (2018) have shown that the variances of the Gaussian penalty are smaller in the first layer and larger in the second layer of their 2-layered self-attentional acoustic model. Based on this observation, they suggest that it is better for the first layer to have a restricted range, while a global range is desirable for the upper layer. We performed a similar analysis for our models, but obtained quite different results. First of all, Table 5 shows that, in our experiments, the initial value of variance plays a role but it appears to be less critical. An inspection of the final variance values is shown in Figure 3, in which we do not observe any relation between the layers and the variance. On the contrary, we observe that different heads in the same layer can differ significantly. Additionally, the initial weight makes a big difference for the final values but, as shown in table 5, this does not affect the performance significantly. To understand whether results' differences from the work of Sperber et al. (2018) are related to the task (SLT instead of ASR), we checked the weights of our ASR models and find that they do not differ significantly from the ones showed in Figure 3. The absence of a pattern in the distribution of the variance is a further justification to use a logarithmic distance penalty in all the layers.

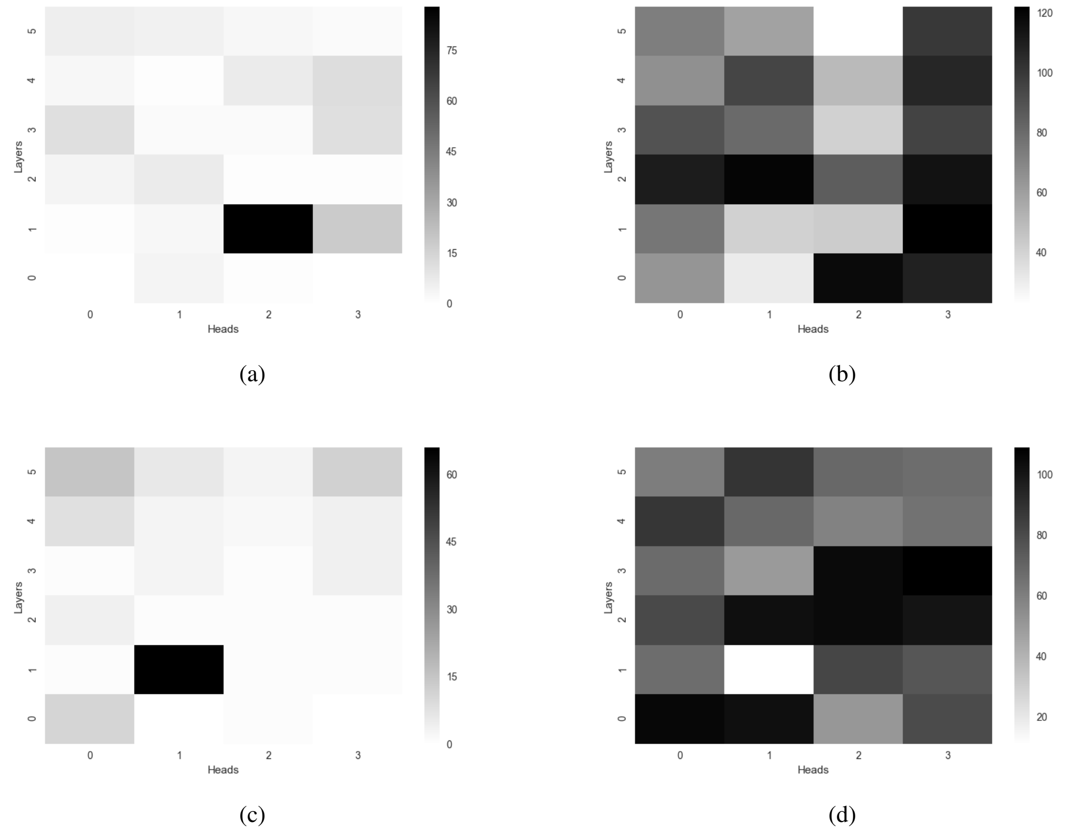
|  | LSTM | log | Gauss | BIG+log | BIG+Gauss |
|---|---|---|---|---|---|
| De | 12.9 | **14.5** | 14.4 | **17.3** | 16.2 |
| Es | 17.9 | 18.4 | **18.6** | **20.8** | 20.1 |
| Fr | 22.3 | 23.1 | **24.0** | **26.9** | 24.7 |
| Pt | 17.1 | 18.6 | **19.7** | **20.1** | 19.3 |
| Ro | 13.4 | 14.7 | **15.0** | **16.5** | 16.1 |
| Ru | 7.2 | 8.8 | **9.1** | **10.5** | 8.5 |

**Table 6:** Results on six language pairs covered by the multilingual corpus. LSTM is the CNN+LSTM model. Results in columns 3-6 are computed with S-Transformer with logarithmic (log) or Gaussian (Gauss) distance penalty. Improvements over CNN+LSTM are statistically significant.

### 6.4 Additional experiments

The previous experiments have shown that S-Transformer performs better than the other variants, and as such we report experiments on the larger MuST-C corpus only with S-Transformer and the two distance penalties. S-Transformer outperforms CNN+LSTM on all the 6 language directions with gains from +0.5 to +1.6 BLEU points with log penalty and from +0.7 to +2.6 with Gaussian penalty. Gaussian penalty generally achieves results only slightly better than the logarithmic one, except for the top improvements of +0.9 and +1.1 respectively on En→Fr and En→Pt. To explain this difference, it is useful to recall that the parameters of the encoders of SLT models (including their Gaussian variances) are initialized from a model pre-trained on English ASR. In particular, for the multilingual corpus we use the same model trained on the larger dataset. The inherited variance from this model may affect differently the different target languages.

Experiments with a larger model (S-Transformer BIG) show further improvements from a minimum of 1.5 points for En→Pt to a maximum of 3.8 points for En→Fr with log penalty, while the poor results with Gaussian penalty confirm that it is less stable than the logarithmic one. The number of training iterations is also reduced to less than half of the previous



(a) (b) (c) (d)

**Figure 3:** Final values of the variances for the SLT task in Librispeech (top) and IWSLT (bottom) with initial variance of 5.0 (left) and 100.0 (right).

experiments. The improvements obtained in this experiment, up to 4.6 BLEU point in En→Fr over the baseline, represent a step forward towards a translation quality that allows real-world applications for end-to-end SLT.

To conclude, our experiments show that: *i)* our task-specific adaptations make the Transformer trainable for the SLT task, also on a single GPU; *ii)* when both short-range and 2D dependencies are explicitly addressed in the model, they allow it to outperform a strong baseline based on LSTMs; *iii)* the logarithmic distance penalty can be preferable over the Gaussian one because it does not require additional hyperparameter tuning and results in competitive performance.

## 7 Conclusion

We have shown that the application of Transformer to end-to-end SLT is problematic in the encoder side. Consequently, we have proposed to enhance the Transformer encoder by taking into account the characteristics of a speech spectrogram. Our solution consists of: *i)* 2D processing of the input to compress it effectively before the self-attentional stack; and *ii)* a distance penalty in the encoder self-attention layers that forces the network to give more attention to neighboring time steps.We have shown that, although using a distance penalty is always beneficial, a simple logarithmic function can result in equal or better improvements than a learnable Gaussian penalty. Experimental results performed on three different corpora, for a total of 6 language directions, show that our approach outperforms a strong recurrent baseline in both translation quality and training time.

## References

Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep Speech 2: End-to-end Speech Recognition in English and Mandarin. In *International conference on machine learning*, pages 173–182.

Anastasopoulos, Antonios and David Chiang. 2018. Tied Multitask Learning for Neural Speech Translation. In *Proceedings of NAACL 2018*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR 2015*.

Bansal, Sameer, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Pre-training on High-resource Speech Recognition Improves Low-resource Speech-to-text Translation. *Proceedings of NAACL 2019*.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of EMNLP 2016)*.

Bérard, Alexandre, Olivier Pietquin, Laurent Besacier, and Christophe Servan. 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, December.

Bérard, Alexandre, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *Proceedings of ICASSP 2018*, Calgary, Alberta, Canada, April.

Braune, Fabienne and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of ACL 2010*, pages 81–89.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014. In *Proceedings of IWSLT 2014*.

Chan, William, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Chen, Mia Xu, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Cheng, Jianpeng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Chiu, Chung-Cheng, Tara Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. 2018. State-of-the-art Speech Recognition with Sequence-to-sequence Models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Di Gangi, Mattia A and Marcello Federico. 2018. Deep Neural Machine Translation with Weakly-Recurrent Units. In *Proc. of EAMT*.

Di Gangi, Mattia Antonino, Dessì Roberto, Roldano Cattoni, Matteo Negri, and Marco Turchi. 2018. Fine-tuning on clean data for end-to-end speech translation: Fbk@ iwslt 2018. In *International Workshop on Spoken Language Translation (IWSLT 2018)*, pages 147–152.

Di Gangi, Mattia A., Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Dong, Linhao, Shuang Xu, and Bo Xu. 2018. 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.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proceedings of ICML 2017*, Sydney, Australia, August.

Jia, Ye, Melvin Johnson, Wolfgang Macherey, Ron-J. Weiss, Yuan Cao, Chung-Cheng Chiu, Stella-Laurenzo Ari, and Yonghui Wu. 2018. Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation. *ArXiv e-prints arXiv:1811.02050*.

Kalchbrenner, Nal, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Kingma, Diederik and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Kocabiyikoglu, Ali Can, Laurent Besacier, and Olivier Kraif. 2018. Augmenting Librispeech with French Translations: A Multimodal Corpus for Direct Speech Translation Evaluation. In *Proceedings of LREC 2018*, Miyazaki, Japan, May.

Lei, Tao, Yu Zhang, and Yoav Artzi. 2017. Training RNNs as Fast as CNNs. *arXiv preprint arXiv:1709.02755*.

Li, Jinyu, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong. 2016. Exploring Multidimensional LSTMs for Large Vocabulary ASR. In *Proceedings of ICASSP 2016*, pages 4940–4944. IEEE.

Liu, Dan, Junhua Liu, Wu Guo, Shifu Xiong, Zhiqiang Ma, Rui Song, Chongliang Wu, and Quan Liu. 2018. The ustc-nel speech translation system at iwslt 2018. In *Proocedings of IWSLT*.

Niehues, Jan, Roldano Cattoni, Sebastian Stüker, Mauro Cettolo, Marco Turchi, and Marcello Federico. 2018. The IWSLT 2018 Evaluation Campaign. In *Proceedings of IWSLT 2018*, Bruges, Belgium, October.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL 2002*.

Pascanu, Razvan, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. In *Proceedings of ICLR 2014*.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop Autodiff*.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. Technical report, IEEE Signal Processing Society.

Povey, Daniel, Hossein Hadian, Pegah Ghahremani, Ke Li, and Sanjeev Khudanpur. 2018. A time-restricted self-attention layer for asr. In *Proceedings of ICASSP 2018*, pages 5874–5878. IEEE.

Ruiz, Nicholas, Mattia Antonino Di Gangi, Nicola Bertoldi, and Marcello Federico. 2017. Assessing the tolerance of neural machine translation systems against speech recognition errors. *Proc. Interspeech 2017*, pages 2635–2639.

Sperber, Matthias, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. Self-Attentional Acoustic Models. *Proceedings of Interspeech 2018*, pages 3723–3727.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS 2014*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of NIPS 2017*.

Vila, Laura-Cross, Carlos Escolano, José AR Fonollosa, and Marta-R Costa-Jussà. 2018. End-to-End Speech Translation with the Transformer. *Proceedings of IberSPEECH 2018*, pages 60–63.

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

Zeyer, Albert, Kazuki Irie, Ralf Schlüter, and Hermann Ney. 2018. Improved Training of End-to-end Attention Models for Speech Recognition. In *Proceedings of Interspeech 2018*, pages 7–11.

Zhang, Yu, William Chan, and Navdeep Jaitly. 2017. Very deep convolutional networks for end-to-end speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4845–4849. IEEE.

Zhou, Shiyu, Linhao Dong, Shuang Xu, and Bo Xu. 2018a. Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese. *Proc. Interspeech 2018*, pages 791–795.

Zhou, Shiyu, Shuang Xu, and Bo Xu. 2018b. Multilingual end-to-end speech recognition with a single transformer on low-resource languages. *arXiv preprint arXiv:1806.05059*.