IberSPEECH 2018
21-23 November 2018, Barcelona, Spain

End-to-End Speech Translation with the Transformer

Laura Cros Vila, Carlos Escolano, José A. R. Fonollosa, Marta R. Costa-jussà

TALP Research Center, Universitat Politècnica de Catalunya, Barcelona

lcrosvila@gmail.com,{carlos.escolano, jose.fonollosa, marta.ruiz}@upc.edu

Abstract

Speech Translation has been traditionally addressed with the concatenation of two tasks: Speech Recognition and Machine Translation. This approach has the main drawback that errors are concatenated. Recently, neural approaches to Speech Recognition and Machine Translation have made possible facing the task by means of an End-to-End Speech Translation architecture.

In this paper, we propose to use the architecture of the Transformer which is based solely on attention-based mechanisms to address the End-to-End Speech Translation system. As a contrastive architecture, we use the same Transformer to built the Speech Recognition and Machine Translation systems to perform Speech Translation through concatenation of systems.

Results on a Spanish-to-English standard task show that the end-to-end architecture is able to outperform the concatenated systems by half point BLEU.

Index Terms: End-to-End Speech Translation, Transformer

1. Introduction

The fields of Machine Translation (MT) and Automatic Speech Recognition (ASR) share many features, including conceptual foundations, sustained interest and attention of researchers in the field, a remarkable progress in the last two decades and the resulting wide popular use. Both ASR and MT have a long way to improve and, as a result, do not give perfect results. Speech Translation (ST) applications are typically created by combining ASR and MT systems [1, 2].

This pipeline implies that each system has to be trained with their own dataset (which are required to be large) creating a big drawback for low resourced languages. In addition, all errors made by the recognizer go to the MT system and then the MT system itself adds its own errors. The errors are combined, and the results are often very poor.

Deep learning architectures have allowed for end-to-end approaches for both machine translation [3] and speech recognition [4]. Both systems are based on an architecture of encoder-decoder with recurrent neural networks and attention mechanisms. This architecture has been successfully extended to end-to-end speech translation [5].

Recently, there has been a new proposed architecture for addressing machine translation [6]. Later, this architecture has also been used for speech recognition[1] [7]. In both cases, the Transformer outperforms previous architectures based on recurrent neural networks. Inspired by these previous works, this paper describes how to adapt this architecture to end-to-end speech translation. The rest of the paper is organised as follows. Section 2 briefly describes the architecture of the Transformer to make this paper self-contained. Section 3 reports the details on the experimental part including databases, training parameters and results. Finally, 4 reports the final conclusions.

2. Transformer

The goal of this work is to build an End-to-End Speech Translation (ST) based on the Transformer [8]. This end-to-end task will be contrasted with the concatenation of the Automatic Speech Recognition (ASR) and Machine Translation (MT) systems, which are also built using the Transformer architecture. This section briefly describes this architecture.

As many neural sequence transduction models, the Transformer has an encoder-decoder structure. The main difference between it and any other models is that Transformer is entirely based on attention mechanisms [9] and point-wise, fully connected layers for both the encoder and the decoder. This makes it computationally cheaper than other architectures with similar test scores. The whole architecture of the Transformer is depicted in (Figure 1).

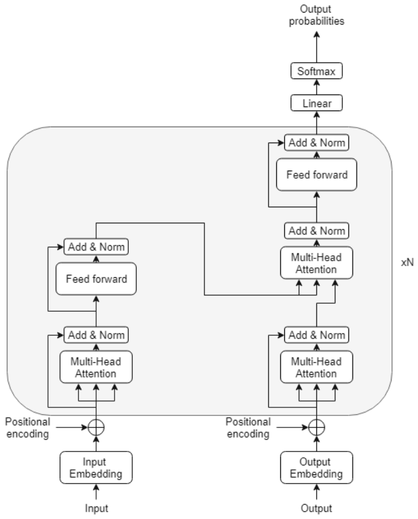


Figure 1: *Model architecture of the Transformer.*

2.1. Input/Output

Originally, the input of the Transformer is a sequence of words divided in sub-units denominated $tokens$. Once the text is turned into a tokenized version of the words, a matrix of real numbers collects the vectors (typically of size $d_{model} = 512$). If taken the raw input sequence as $x = (x_1, x_2, \ldots, x_m)$ and the embedded representation as $w = (w_1, w_2, \ldots, w_m)$ with $w_j \in \mathbb{R}^f$, then each w_j is a column vector of the input matrix belonging to the space $\mathbb{R}^{V \times f}$, with V as the number of embeddings and f the number of features of each embedding.
The decoder generates an output sequence corresponding to the

[1] https://tensorflow.github.io/tensor2tensor/tutorials/asr_with_transformer.html

10.21437/IberSPEECH.2018-13

input sentence.

2.2. Positional encoding

The lack of recurrence and convolution in the model entails that no recurrence nor temporal information is available. A good way to keep the order of the sentence is adding positional encoding to the input embeddings at the bottoms of the encoder and decoder stacks.

What is used in Transformer is an element-wise vector $p = (p_1, p_2, \ldots, p_m)$, with $p_j \in \mathbb{R}^f$, which is added to the original matrix.

2.3. Encoder

The encoder consists of a stack of N layers, each of them composed of two sub-layers: a multi-head attention mechanism and a fully-connected feed forward net, plus residual connections [2] (referenced in Figure 1 as ”Add”) on both stages, followed by a layer of normalization.

The multi-head attention has several parallel attention layers, or heads, which concatenate attention functions with different linearly projected queries, keys and values.

2.4. Decoder

The decoder resembles the encoder but it is not completely equal. Although it is also a stack of N layers layers with sub-layers within them, a masked multi-head attention layer is added (apart from the common residual connections and layer normalization). The singular fact of the decoder is that at each step the model is auto-regressive, meaning that it uses the previously generated symbols as additional input when generating the next ones.

3. Experimental work

Our implementation of the Transformer is based on Tensor2Tensor [8], or T2T for short, which is a library of deep learning models and datasets actively used and maintained by researchers and engineers within the Google Brain team and a community of users. As follows, we report the database used, the parameters to train the system and the results.

3.1. Database

The database used in this experiment is the Fisher Spanish and Callhome Spanish Corpus. The Fisher Spanish Corpus provides a set of speech and transcripts developed by the Linguistic Data Consortium (LDC) which consists of audio files covering roughly 163 hours of telephone speech from 136 native Caribbean Spanish and non-Caribbean Spanish speakers. The speech recordings consist of 819 telephone conversations of 10 to 12 minutes in duration. Full orthographic transcripts of these audio files are available in [3]LDC2010T04. The audio files are available in [4]LDC2010S01.

The CALLHOME Spanish Corpus consists of 120 unscripted telephone conversations between native speakers of Spanish. All calls, which lasted up to 30 minutes, originated in North America and were placed to international locations. Most participants called family members or close friends. The audio files of the CALLHOME Corpus are available at [5]LDC96S35. The transcripts of these audio files are available at [6]LDC96T17. The transcript files are in plain-text, tab-delimited format (tdf) with UTF-8 character encoding. In order to adapt the transcript files for the Transformer, all the text was turned into capital letters as well as a reference number at the beginning of each sentence was added, consisting of six digits starting from ”000000” to the last sentence and separated with a tabulator such as follows:

000000 HELLO
000001 ALO.
000002 ALO, BUENAS NOCHES. QUIéN ES?
000003 QUé TAL, EH, YO SOY GUILLERMO, CóMO ESTáS?
000004 AH GUILLERMO.
...
003637 OH MY GOD.
003638 MHM. Y NO LE PODíAN HACER NADA, NO.
003639 MM.

The conversations were recorded as 2-channel mu-law sample data with 8000 samples per second (as captured from the public telephone network).

Table 1 shows the corpus statistics of the text dataset for Spanish-English.

L	Set	S	W	V
Spanish	Train	138819	1503003	57587
	Dev1	3979	41271	6251
	Dev2	3960	40072	5793
	Test	3641	40141	5888
English	Train	138819	1441090	37817
	Dev1	3979	40015	5053
		3979	39977	5059
		3979	39799	5163
	Dev2	3960	39152	4739
		3960	39513	4734
		3960	39004	4840
	Test	3641	39617	4929
		3641	39011	4901
		3641	38578	4875

Table 1: *Corpus Statistics. Language (L), number of sentences (S),words (W), vocabulary (V).*

For the MT experiment, we used the parallel text from LDC2014T23 available from LDC[7] and in github [8].

3.2. Parameters

For the experiments, we used three different architectures. These three architectures correspond to the ASR, MT and ST models.

The main hyperparameters used are detailed in the Table 2. For the ASR/ST models the learning rate had a decay each

[2] Assuming that the function to be modeled (weights and bias of the net) is closer to an identity mapping than to a zero mapping, a good way to optimize the learning process is to add residual connections, which provides the input without any transformation to the output of the layer.

[3] https://catalog.ldc.upenn.edu/LDC2010T04

[4] https://catalog.ldc.upenn.edu/LDC2010S01

[5] https://catalog.ldc.upenn.edu/LDC96S35

[6] https://catalog.ldc.upenn.edu/LDC96T17

[7] https://catalog.ldc.upenn.edu/ldc2014t23

[8] https://github.com/joshua-decoder/fisher-callhome-corpus

5000 steps and the learning rate warm-up steps set to 8000[9]. To adapt the speech features in the ASR encoder, we used the *conv_relu_conv* from tensor2tensor. As parameters, we used *mel_filterbank* of 80 coefficients every 10ms with a window of 25 ms. As preprocessing for the ASR inputs, we used the tensor2tensor options as follows *conv1d(inputs, filter_size=1536, kernel_size=9) + relu + conv1d(inputs, filter_size=384, kernel_size=1). The Transformer gets a vector of dimension 384 every 10ms.* Also for the speech part, a clarification of the input and target maximum sequence length is that to have an input maximum sequence length of 1550 means that only examples of transcriptions whose audio has less than 1550 frames are used, which implies that with frames of 10 ms the maximum size of the input audio frame is approximately 15.5 seconds in length. On the other hand, to have a target maximum sequence length of 350 means that the train transcripts are limited to a maximum size of 350 characters.

The speech models were trained on TPUs [10] following the suggested parameters for the librispeech task of the tensor2tensor library [11].

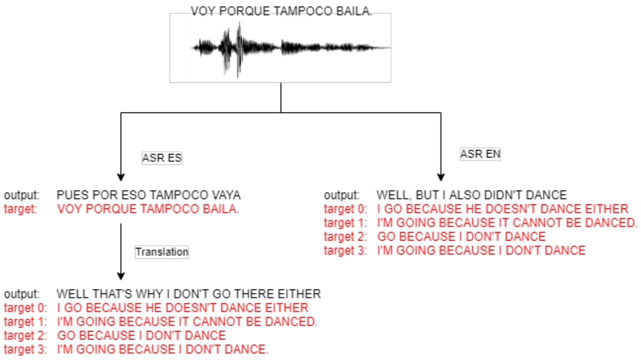
3.3. Training

When training, as there are several GPUs or TPUs, the parameters are applied to each one. So the effective batch size is the numbers of GPUs (in this case 4 or 8 in a TPU) multiplied by the batch size. In each batch the parameters are updated using the stochastic gradient descend and the Adam optimizer [12].

Both the ASR/ST and MT systems use a character-based tokenization. This implies that the models look for the correlation of input and output sentences character by character.

3.4. Results

The evaluation of each model involving translation was done by computing the Bilingual Evaluation Understudy score (BLEU) [13]. The BLEU score is the most used for the field of MT and it compares the decoded sentence with the target sentence of the test set by looking into the *modified n-gram precision*. As for ASR evaluation, a commonly used metric is Word Error Rate (WER) [14], which is defined as the ratio of word errors (substitutions, deletions and insertions) to words processed. For the evaluation of ASR systems, punctuation marks were not taken into account.



output: PUES POR ESO TAMPOCO VAYA
target: VOY PORQUE TAMPOCO BAILA.

output: WELL THAT'S WHY I DON'T GO THERE EITHER
target 0: I GO BECAUSE HE DOESN'T DANCE EITHER
target 1: I'M GOING BECAUSE IT CANNOT BE DANCED.
target 2: GO BECAUSE I DON'T DANCE
target 3: I'M GOING BECAUSE I DON'T DANCE.

output: WELL, BUT I ALSO DIDN'T DANCE
target 0: I GO BECAUSE HE DOESN'T DANCE EITHER
target 1: I'M GOING BECAUSE IT CANNOT BE DANCED.
target 2: GO BECAUSE I DON'T DANCE
target 3: I'M GOING BECAUSE I DON'T DANCE

Figure 2: *Speech Translation Example.*

Comparing ASR+MT concatenation and End-to-End Speech Translation, the results show that in terms of BLEU, the latter is slightly better than the former gaining 0.5 points of BLEU.

Figure 2 shows an example that when concatenating ASR and MT, the errors are also concatenated. The Spanish target word *BAILA* (*DANCE* in English), when recognized with the model ASR ES, is misspelled and transcribed to the word *VAYA*, which has a very similar sound but totally different meaning. As a consequence, the final translation output can not reproduce the word *DANCE*, which gives a strong meaning of context to the sentence. In this case, the end-to-end system is able to produce a better translation

4. Conclusions

This paper proposes to use of the Transformer as main architecture for Speech Recognition, Machine Translation and Speech Translation. To the best of our knowledge, this is the first time that this promising architecture is used to reproduce an End-to-End Speech Translation system. BLEU results show that the End-to-End Speech Translation architecture provides slightly better results than the standard ASR and MT concatenation. Examples show that these better results are achieved by avoiding the concatenation of errors.

In future work, it would be interesting to train a system capable of doing multi-task learning [15]. This system would build several models and not only the one learning to translate from Spanish speech to English text. The new multi-task model would learn in addition Spanish Recognition and/or Spanish-to-English text translation.

5. Acknowledgements

This work is supported in part by the Spanish Ministerio de Economía y Competitividad, the European Regional Development Fund and the Agencia Estatal de Investigación, through the postdoctoral senior grant Ramón y Cajal, the contract TEC2015-69266-P (MINECO/FEDER,EU) and the contract PCIN-2017-079 (AEI/MINECO).

6. References

[1] A. Waibel and C. Fugen, “Spoken language translation,” *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 70–79, May 2008.

[2] M. Dureja and S. Gautam, “Article: Speech-to-speech translation: A review,” *International Journal of Computer Applications*, vol. 129, no. 13, pp. 28–30, November 2015, published by Foundation of Computer Science (FCS), NY, USA.

[3] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell,” *CoRR*, vol. abs/1508.01211, 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[5] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, “Sequence-to-sequence models can directly translate foreign speech,” 2017. [Online]. Available: https://arxiv.org/abs/1703.08581

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[9] To set the learning rate warm-up steps to 8000 means that the first 8k steps the learning rate grows linearly and then follows an inverse square root decay.

Hparam	**Text-to-Text (GPU)**	**ASR/ST (TPU)**
Number of encoder layers	6	6
Number of decoder layers	6	4
Gradient clipping	No	No
Learning rate	0.2	0.15
Momentum	0.9	0.9
Audio sampling rate	-	8000
Batch size	4096	16
Maximum length	256	125550
Input sequence maximum length	0	1550
Target sequence maximum length	0	350
Adam optimizer	$\beta_1 = 0.9$ $\beta_2 = 0.997$ $\epsilon = 10^{-9}$	$\beta_1 = 0.9$ $\beta_2 = 0.997$ $\epsilon = 10^{-9}$
Attention layers	8	2
Initializer	uniform unit scaling	uniform unit scaling
Initializer gain	1.0	1.0
Training steps	250000	210000

Table 2: *Training parameters.*

System	Results
ASR ES (TPU)	38.02 (WER)
MT (GPU)	55.05 (BLEU)
ASR + MT (TPU/GPU)	19.97 (BLEU)
ASR EN (TPU)	20.47 (BLEU)

Table 3: *Results of the model evaluation. ASR ES stands for the speech recognition with Spanish transcriptions as target. ASR EN stands for the speech recognition with English transcriptions as target.*

[7] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-Based Sequence-to-Sequence Speech Recognition with the Transformer in Mandarin Chinese," *ArXiv e-prints*, 2018.

[8] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar *et al.*, "Tensor2tensor for neural machine translation," *arXiv preprint arXiv:1803.07416*, 2018.

[9] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of attention in sequence-to-sequence models,," in *Proc. of Interspeech*, 2017.

[10] N. Jouppi, "Google supercharges machine learning tasks with tpu custom chip," *Google Blog, May*, vol. 18, 2016.

[11] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," *CoRR*, vol. abs/1803.07416, 2018. [Online]. Available: http://arxiv.org/abs/1803.07416

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[14] A. C. Morris, V. Maier, and P. Green, "From wer and ril to mer and wil: improved evaluation measures for connected speech recognition," in *Eighth International Conference on Spoken Language Processing*, 2004.

[15] M. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *CoRR*, vol. abs/1511.06114, 2015. [Online]. Available: http://arxiv.org/abs/1511.06114