End-to-End Automatic Speech Translation of Audiobooks

Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, Olivier Pietquin

END-TO-END AUTOMATIC SPEECH TRANSLATION OF AUDIOBOOKS

Alexandre Bérard†⋆ *Laurent Besacier*⋆ *Ali Can Kocabiyikoglu*⋆ *Olivier Pietquin*†

⋆LIG - Univ. Grenoble-Alpes (France) †CRIStAL - Univ. Lille (France)

ABSTRACT

We investigate end-to-end speech-to-text translation on a corpus of audiobooks specifically augmented for this task. Previous works investigated the extreme case where source language transcription is not available during learning nor decoding, but we also study a midway case where source language transcription is available at training time only. In this case, a single model is trained to decode source speech into target text in a single pass. Experimental results show that it is possible to train compact and efficient end-to-end speech translation models in this setup. We also distribute the corpus and hope that our speech translation baseline on this corpus will be challenged in the future.

Index Terms— End-to-end models, Speech Translation, LibriSpeech.

1. INTRODUCTION

Most spoken language translation (SLT) systems integrate (loosely or closely) two main modules: source language speech recognition (ASR) and source-to-target text translation (MT). In these approaches, a symbolic sequence of words (or characters) in the source language is used as an intermediary representation during the speech translation process. However, recent works have attempted to build end-to-end speech-to-text translation without using source language transcription during learning or decoding. One attempt to translate directly a source speech signal into target language text is that of [1]. However, the authors focus on the alignment between source speech utterances and their text translation without proposing a complete end-to-end translation system. The first attempt to build an end-to-end speech-to-text translation system (which does not use source language) is our own work [2] but it was applied to a synthetic (TTS) speech corpus. A similar approach was then proposed and evaluated on a real speech corpus by [3].

This paper is a follow-up of our previous work [2]. We now investigate end-to-end speech-to-text translation on a corpus of audiobooks - *LibriSpeech* [4] - specifically augmented to perform end-to-end speech translation [5]. While previous works [2, 3] investigated the extreme case where source language transcription is not available during learning nor decoding (unwritten language scenario defined in [6, 7]), we also investigate, in this paper, a midway case where a certain amount of source language transcription is available during training. In this intermediate scenario, a unique (end-to-end) model is trained to decode source speech into target text through a single pass (which can be interesting if compact speech translation models are needed).

This paper is organized as follows: after presenting our corpus in section 2, we present our end-to-end models in section 3. Section 4 describes our evaluation on two datasets: the synthetic dataset used in [2] and the audiobook dataset described in section 2. Finally, section 5 concludes this work.

2. AUDIOBOOK CORPUS FOR END-TO-END SPEECH TRANSLATION

2.1. Augmented LibriSpeech

Large quantities of parallel texts (e.g. Europarl or OpenSubtitles) are available for training text machine translation systems, but there are no large (>100h) and publicly available parallel corpora that include speech in a source language aligned to text in a target language. The *Fisher/Callhome* Spanish-English corpora [8] are only medium size (38h), contain low-bandwidth recordings, and are not available for free.

We very recently built a large English to French corpus for direct speech translation training and evaluation [5][1], which is much larger than the existing corpora described above. We started from the *LibriSpeech* corpus used for Automatic Speech Recognition (ASR), which has 1000 hours of speech aligned with their transcriptions [4].

The read audiobook recordings derive from a project based on a collaborative effort: LibriVox. The speech recordings are based on public domain books available on *Gutenberg Project*[2] which are distributed in *LibriSpeech* along with the recordings.

Our augmentation of *LibriSpeech* is straightforward: we automatically aligned e-books in a foreign language (French) with English utterances of *LibriSpeech*. This lead to 236 hours of English speech aligned to French translations at utterance level (more details can be found in [5]). Since

[1] The Augmented LibriSpeech corpus is available for download here: `https://persyval-platform.univ-grenoble-alpes.fr/DS91/detaildataset`

[2] `https://www.gutenberg.org/`

English (source) transcriptions are initially available for *LibriSpeech*, we also translated them using *Google Translate*. To summarize, for each utterance of our 236h corpus, the following quadruplet is available: English speech signal, English transcription, French text translation 1 (from alignment of e-books) and translation 2 (from MT of English transcripts).

2.2. MT and AST tasks

This paper focuses on the speech translation (AST) task of audiobooks from English to French, using the Augmented LibriSpeech corpus. We compare a direct (end-to-end) approach, with a cascaded approach that combines a neural speech transcription (ASR) model with a neural machine translation model (MT). The ASR and MT results are also reported as baselines for future uses of this corpus.

Augmented LibriSpeech contains 236 hours of speech in total, which is split into 4 parts: a test set of 4 hours, a dev set of 2 hours, a clean train set of 100 hours, and an extended train set with the remaining 130 hours. Table 1 gives detailed information about the size of each corpus. All segments in the corpus were sorted according to their alignment confidence scores, as produced by the alignment software used by the authors of the corpus [5]. The test, dev and train sets correspond to the highest rated alignments. The remaining data (extended train) is more noisy, as it contains more incorrect alignments. The test set was manually checked, and incorrect alignments were removed. We perform all our experiments using *train* only (without *extended train*). Furthermore, we double the training size by concatenating the aligned references with the Google Translate references. We also mirror our experiments on the BTEC synthetic speech corpus, as a follow-up to [2].

3. END-TO-END MODELS

For the three tasks, we use encoder-decoder models with attention [9, 10, 11, 2, 3]. Because we want to share some parts of the model between tasks (multi-task training), the ASR and AST models use the same encoder architecture, and the AST and MT models use the same decoder architecture.

3.1. Speech Encoder

The speech encoder is a mix between the convolutional encoder presented in [3] and our previously proposed encoder [2]. It takes as input a sequence of audio features: $\mathbf{x} = (x_1, \ldots, x_{T_x}) \in \mathbb{R}^{T_x \times n}$. Like [2], these features are given as input to two non-linear ($tanh$) layers, which output new features of size n'. Like [3], this new set of features is then passed to a stack of two convolutional layers. Each layer applies 16 convolution filters of shape $(3, 3, depth)$ with a stride of $(2, 2)$ w.r.t. time and feature dimensions; $depth$ is 1 for the first layer, and 16 for the second layer. We get features of shape $(T_x/2, n'/2, 16)$ after the 1st layer, and $(T_x/4, n'/4, 16)$ after the 2nd layer. This latter tensor is flattened with shape $(T'_x = T_x/4, 4n')$ before being passed to a stack of three bidirectional LSTMs. This set of features has $1/4th$ the time length of the initial features, which speeds up training, as the complexity of the model is quadratic with respect to the source length. In our models, we use $n' = 128$, which gives features of size 512.

The last bidirectional LSTM layer computes a sequence of annotations $\mathbf{h} = h_1, \cdots, h_{T'_x}$, where each annotation h_i is a concatenation of the corresponding forward and backward states: $h_i = (\overrightarrow{h_i} \oplus \overleftarrow{h_i}) \in \mathbb{R}^{2m}$, with m the encoder cell size.

This model differs from [2], which did not use convolutions, but time pooling between each LSTM layer, resulting in a shorter sequence (pyramidal encoder).

3.2. Character-level decoder

We use a character-level decoder composed of a conditional LSTM [12], followed by a dense layer.

$$s_t, o_t = update^1(s'_{t-1}, E(y_{t-1})) \quad (1)$$

$$c_t = look(o_t, \mathbf{h}) \quad (2)$$

$$s'_t, o'_t = update^2(s_{t-1}, c_t) \quad (3)$$

$$y_t = generate(o't \oplus c_t \oplus E(y_{t-1})) \quad (4)$$

where $update^1$ and $update^2$ are two LSTMs with cell size m'. $look$ is a vanilla global attention mechanism [9], which uses a feed-forward network with one hidden layer of size m'. $E^{k \times |V|}$ is the target embedding matrix, with k the embedding size and $|V|$ the vocabulary size. $c_t \in \mathbb{R}^{2m}$ is a context vector which summarizes the input states to help the decoder generate a new symbol and update its state. $generate$ uses a nonlinear layer followed by a linear projection to compute a score for each symbol in target vocabulary V. It then picks target symbol z_t with the highest score:

$$generate(x) = \arg\max_{i=1}^{|V|} z_i \quad (5)$$

$$z = W_{proj} \tanh(W_{out}^T x + b_{out}) + b_{proj} \quad (6)$$

with $W_{proj} \in \mathbb{R}^{|V| \times l}, b_{proj} \in \mathbb{R}^{|V|}$, $W_{out} \in \mathbb{R}^{l \times (m' + 2m + k)}$, $b_{out} \in \mathbb{R}^l$, where l is the output layer size.

4. EXPERIMENTS

4.1. Model Settings

Speech files were preprocessed using Yaafe [13], to extract 40 MFCC features and frame energy for each frame with a step size of 10 ms and window size of 40 ms, following [14, 2]. We tokenize and lowercase all the text, and normalize the punctuation, with the Moses scripts[3]. For BTEC, the same preprocessing as [2] is applied. Character-level vocabularies

[3] http://www.statmt.org/moses/

Corpus		Total		Source (per segment)			Target (per segment)
		segments	hours	frames	chars	(sub)words	chars
LibriSpeech (Real)	train 1 train 2	47271	100:00	762	111	20.7	143 126
	dev	1071	2:00	673	93	17.9	110
	test	2048	3:44	657	95	18.3	112
BTEC (Synthetic)	train	19972	15:51	276	50	10	42
	dev	1512	0:59	236	40	8.1	33
	test	933	0:36	236	41	8.2	34

Table 1: Size of the Augmented LibriSpeech and BTEC corpora, with the average frame, character and word counts (subword count for LibriSpeech) per segment. Character counts take whitespaces into account. The source side of BTEC actually has six times this number of segments and hours, because we concatenate multiple speakers (synthetic voices). LibriSpeech *train 1* (alignments) and *train 2* (automatic translation) share the same source side.

for LibriSpeech are of size 46 for English (transcriptions) and 167 for French (translation). The decoder outputs are always at the character-level (for AST, MT and ASR). For the MT task, the LibriSpeech English (source) side is preprocessed into subword units [15]. We limit the number of merge operations to $30k$, which gives a vocabulary of size $27k$. The MT encoder for BTEC takes entire words as input.

Our BTEC models use an LSTM size of $m = m' = 256$, while the LibriSpeech models use a cell size of 512, except for the speech encoder layers which use a cell size of $m = 256$ in each direction. We use character embeddings of size $k = 64$ for BTEC, and $k = 128$ for LibriSpeech. The MT encoders are more shallow, with a single bidirectional layer. The source embedding sizes for words (BTEC) and subwords (LibriSpeech) are respectively 128 and 256.

The input layers in the speech encoders have a size of 256 for the first layer and $n' = 128$ for the second. The LibriSpeech decoders use an output layer size of $l = 512$. For BTEC, we do not use any non-linear output layer, as we found that this led to overfitting.

4.2. Training settings

We train our models with Adam [16], with a learning rate of 0.001, and a mini-batch size of 64 for BTEC, and 32 for LibriSpeech (because of memory constraints). We use variational dropout [17], i.e., the same dropout mask is applied to all elements in a batch at all time steps, with a rate of 0.2 for LibriSpeech and 0.4 for BTEC. In the MT tasks, we also drop source and target symbols at random, with probability 0.2. Dropout is not applied on recurrent connections [18].

We train all our models on *LibriSpeech train* augmented with the Google Translate references, i.e., the source side of the corpus (speech) is duplicated, and the target side (translations) is a concatenation of the aligned references with the Google Translate references. Because of GPU memory limits, we set the maximum length to 1400 frames for LibriSpeech input, and 300 characters for its output. This covers about 90% of the training corpus. Longer sequences are kept but truncated to the maximum size. We evaluate our models on the dev set every 1000 mini-batch updates using BLEU for AST and MT, and WER for ASR, and keep the best performing checkpoint for final evaluation on the test set.

Our models are implemented with TensorFlow [19] as part of the LIG-CRIStAL NMT toolkit[4].

4.3. Results

Table 2 presents the results for the ASR and MT tasks on BTEC and LibriSpeech. The MT task (and by extension the AST task) on LibriSpeech (translating novels) looks particularly challenging, as we observe BLEU scores around 20%[5].

	Model	ASR (WER ↓)	MT (BLEU ↑)
BTEC	greedy	14.9	47.4
	beam-search	13.8	49.2
	ensemble	**11.3**	**50.7**
LibriSpeech	greedy	19.9	19.2
	beam-search	17.9	18.8
	ensemble	**15.1**	19.3
	Google Translate		**22.2**

Table 2: MT and ASR results on test set for *BTEC* and *Augmented LibriSpeech*. We use a beam size of 8, and ensembles of 2 models trained from scratch.

For Automatic Speech Translation (AST), we try four settings. The *cascaded* model combines both the ASR and MT models (as a pipeline). The *end-to-end* model (described in section 3) does not make any use of source language transcripts. The *pre-trained* model is identical to *end-to-end*, but its encoder and decoder are initialized with our ASR and MT

[4] The toolkit and configuration files are available at: `https://github.com/eske/seq2seq`

[5] Google Translate is also scored as a topline (22.2%).

	greedy	beam	ensemble	params
	Test BLEU			(million)
Baseline [2]	29.1	31.3	37.9†	10.4
Cascaded	38.9	40.7	**43.8**	7.9 + 3.4
End-to-End	31.3	33.7		6.7
Pre-trained	33.7	36.3	**40.4**	
Multi-task	35.1	37.6		

Table 3: Results of the AST task on *BTEC test*. † was obtained with an ensemble of 5 models, while we use ensembles of 2 models. The non-cascaded ensemble combines the *pre-trained* and *multi-task* models. Contrary to [2], we only present mono-reference results.

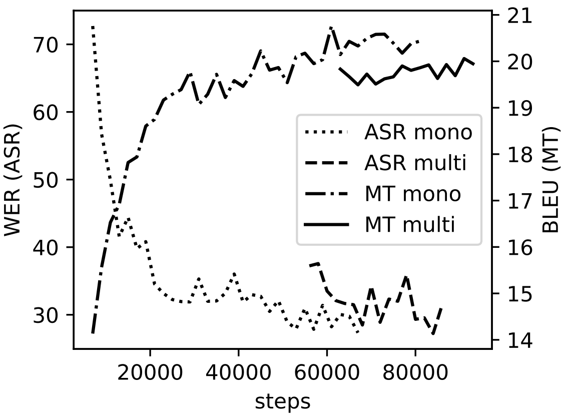

Fig. 1: Augmented LibriSpeech Dev BLEU scores for the MT task, and WER scores for the ASR task, with the initial (mono-task) models, and when multi-task training picks up.

models. The *multi-task* model is also pre-trained, but continues training for all tasks, by alternating updates like [20], with 60% of updates for AST and 20% for ASR and MT.

Table 3 and 4 present the results for the end-to-end AST task on BTEC and LibriSpeech. On both corpora, we show that: (1) it is possible to train compact end-to-end AST models with a performance close to cascaded models; (2) pre-training and multi-task learning[6] improve AST performance; (3) contrary to [3], in both BTEC and LibriSpeech settings, best AST performance is observed when a symbolic sequence of symbols in the source language is used as an intermediary representation during the speech translation process (cascaded system); (4) finally, the AST results presented on LibriSpeech demonstrate that our augmented corpus is useful, although challenging, to benchmark end-to-end AST systems on real speech at a large scale. We hope that our baseline on Augmented LibriSpeech will be challenged in the future.

The large improvements on MT and AST on the BTEC corpus, compared to [2] are mostly due to our use of a better decoder, which outputs characters instead of words.

	greedy	beam	ensemble	params
	Test BLEU			(million)
Cascaded	14.6	14.6	**15.8**	6.3 + 15.9
End-to-End	12.3	12.9	**15.5**†	9.4
Pre-trained	12.6	13.3		
Multi-task	12.6	13.4		

Table 4: AST results on *Augmented LibriSpeech test*. † combines the end-to-end, pre-trained and multi-task models.

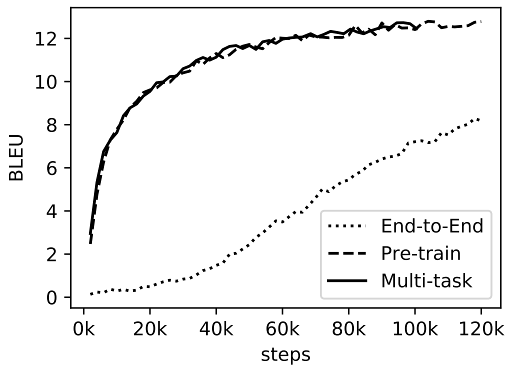

Fig. 2: Dev BLEU scores on 3 models for end-to-end AST of audiobooks. Best scores on the dev set for the end-to-end (mono-task), pre-train and multi-task models were achieved at steps 369k, 129k and 95k.

4.4. Analysis

Figure 1 shows the evolution of BLEU and WER scores for MT and ASR tasks with single models, and when we continue training them as part of a multi-task model. The multi-task procedure does more updates on AST, which explains the degraded results, but we observe that the speech encoder and text decoder are still able to generalize well to other tasks.

Figure 2 shows the evolution of dev BLEU scores for our three AST models on LibriSpeech. We see that pre-training helps the model converge much faster. Eventually, the End-to-End system reaches a similarly good solution, but after three times as many updates. Multi-Task training does not seem to be helpful when combined with pre-training.

5. CONCLUSION

We present baseline results on End-to-End Automatic Speech Translation on a new speech translation corpus of audiobooks, and on a synthetic corpus extracted from BTEC (follow-up to [2]). We show that, while cascading two neural models for ASR and MT gives the best results, end-to-end methods that incorporate the source language transcript come close in performance.

[6] if source transcriptions are available at training time

6. REFERENCES

[1] Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn, "An Attentional Model for Speech Translation Without Transcription," in *NAACL-HLT*, 2016.

[2] Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan, "Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation," *NIPS 2016 End-to-end Learning for Speech and Audio Processing Workshop*, 2016.

[3] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-Sequence Models Can Directly Transcribe Foreign Speech," in *Interspeech*, 2017.

[4] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[5] Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif, "Augmenting Librispeech with French Translations: A Multimodal Corpus for Direct Speech Translation Evaluation," in *LREC*, 2018.

[6] Gilles Adda, Sebastian Stücker, Martine Adda-Decker, Odette Ambouroue, Laurent Besacier, David Blachon, Hélène Bonneau-Maynard, Pierre Godard, Fatima Hamlaoui, Dmitri Idiatov, Guy-Noël Kouarata, Lori Lamel, Emmanuel-Moselly Makasso, Annie Rialland, Mark Van de Velde, François Yvon, and Sabine Zerbian, "Breaking the Unwritten Language Barrier: The Bulb Project," in *Proceedings of SLTU (Spoken Language Technologies for Under-Resourced Languages)*, 2016.

[7] Antonios Anastasopoulos and David Chiang, "A case study on using speech-to-translation alignments for language documentation," in *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 2017.

[8] Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur, "Improved Speech-to-Text Translation with the Fisher and Callhome Spanish-English Speech Translation Corpus," in *IWSLT*, 2013.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *ICLR*, 2015.

[10] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-Based Models for Speech Recognition," in *NIPS*, 2015.

[11] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," in *ICASSP*, 2016.

[12] Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Laeubli, Antonio Valerio, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde, "Nematus: a Toolkit for Neural Machine Translation," in *EACL*, 2017.

[13] Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gaël Richard, "YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software," in *ISMIR (International Society of Music Information Retrieval)*, 2010.

[14] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, Attend and Spell," in *ICASSP*, 2016.

[15] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural Machine Translation of Rare Words with Subword Units," in *ACL*, 2016.

[16] Diederik Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.

[17] Diederik P Kingma, Tim Salimans, and Max Welling, "Variational dropout and the local reparameterization trick," in *NIPS*, 2015.

[18] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, "Recurrent Neural Network Regularization," in *ICLR*, 2014.

[19] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Jon Shlens, Benoit Steiner, Ilya Sutskever, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Oriol Vinyals, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv*, 2015.

[20] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser, "Multi-task Sequence to Sequence Learning," in *ICLR*, 2016.