Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation
Alexandre Bérard, Olivier Pietquin, Laurent Besacier, Christophe Servan

To cite this version:
Alexandre Bérard, Olivier Pietquin, Laurent Besacier, Christophe Servan. Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation. NIPS Workshop on end-to-end learning for speech and audio processing, Dec 2016, Barcelona, Spain. 2016. <hal-01408086>

HAL Id: hal-01408086
https://hal.archives-ouvertes.fr/hal-01408086
Submitted on 6 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
1 Introduction and background

Current speech translation systems integrate (loosely or closely) two main modules: source language speech recognition (ASR) and source-to-target text translation (MT). In these approaches, source language text transcript (as a sequence or as a graph) appears as mandatory to produce a text hypothesis in the target language. In the meantime, deep neural networks have yielded breakthroughs in different areas including machine translation and speech recognition. Current systems have an “encoder-decoder” architecture where a sequence of input symbols (or an input signal) is projected into a continuous low dimensional space and the output sequence is generated from this representation. This type of architecture has been proposed for machine translation (Sutskever et al., 2014; Bahdanau et al., 2015) and for automatic speech recognition (Chorowski et al., 2015; Chan et al., 2016; Zweig et al., 2016). If we exclude some pioneering attempts to directly translate from a source phone sequence to a target word sequence (Besacier et al., 2006) or to discover a bilingual lexicon from parallel phone-word sequences (Godard et al., 2016), the only attempt to translate directly a source speech signal into target language text is that of Duong et al. (2016). 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 same authors (Anastasopoulos et al., 2016) also propose to jointly use IBM translation models (IBM Model 2), and dynamic time warping (DTW) to align source speech and target text, but again, only the alignment performance is measured in their work.

Consequently, this paper proposes a first attempt to build an end-to-end speech-to-text translation system, which does not use source language text during learning or decoding. Relaxing the need for source language transcription would drastically change the data collection methodology in speech translation, especially in under-resourced scenarios. For instance, in the former project DARPA TRANSTAC (speech translation from spoken Arabic dialects), a large effort was devoted to the collection of speech transcripts (and a prerequisite to obtain transcripts was often a detailed transcription guide for languages with little standardized spelling). Now, if end-to-end approaches for speech-to-text translation are successful, one might consider collecting data by asking bilingual speakers to directly utter speech in the source language from target language text utterances. Such an approach has the advantage to be applicable to any unwritten (source) language.
2 Model

Description We compare two end-to-end models: one for standard machine translation, and one for speech translation. Both models are attention-based encoder-decoder neural networks (Sutskever et al., 2014; Bahdanau et al., 2015).

Given an input sequence \((x_1, \cdots, x_A)\), the encoder (a multi-layered bidirectional LSTM) produces a sequence of outputs \(h = (h_1, \cdots, h_A)\). We initialize the decoder state with \(s_0 = tanh(W_{init}s_A')\) where \(s_A'\) is the last state of the encoder. The next states of the decoder are computed as follows:

\[
s_t, o_t = LSTM(s_{t-1}, Ez_{t-1})
\]

(1)

\(s_t\) is the new state of the LSTM, while \(o_t\) is its output. During training, \(z_{t-1}\) is the previous symbol in the target (reference) sequence. During evaluation, \(z_{t-1}\) is the predicted symbol at the previous time-step. \(E\) is the target embedding matrix.

We compute an attention vector \(d_t\) (over the encoder output \(h\)), which is concatenated to the LSTM’s output \(o_t\):

\[
y_t = W_{proj}(o_t + d_t) + b_{proj}
\]

(2)

This vector is then mapped to a vector of the same size as the target vocabulary, to compute a probability distribution over all target words. At each time-step \(t\), the greedy decoder picks the word \(w_t\) with the highest probability according to this distribution.

\[
p(w_t|y_{t-1}, s_{t-1}) = softmax(W_{out}y_t + b_{out})
\]

(3)

Attention mechanism For text input, we use the same attention model as Vinyals et al. (2015):

\[
attention(h, s_t) = \sum_{i=1}^{A} a_t^i h_i
\]

(4)

\[
a_t^i = softmax(u_t^i)
\]

(5)

Convolutional attention For speech input, we use the same attention model as Chorowski et al. (2015), which uses a convolution filter \(F\) to take into account the attention weights at the previous time-step. This allows the attention model to “remember” and not translate the same part of the signal twice. This is especially useful for speech as the input sequence can be quite long.

\[
u_t^i = v^T \tanh(W_1 h_i + W_2 s_i + b_2 + f_{t,i})
\]

(6)

\[W_{init} \in \mathbb{R}^{2m \times 2m}, E \in \mathbb{R}^{n \times V}, W_{proj} \in \mathbb{R}^{m \times 2m}, W_{out} \in \mathbb{R}^{V \times m}, W_1 \in \mathbb{R}^{m \times m}, W_2 \in \mathbb{R}^{m \times 2m},
\]

\(\mu \in \mathbb{R}^{m}, F \in \mathbb{R}^{k}\), where \(n\) is the embedding size, \(m\) the size of the LSTM layers (the state size is 2m), \(k\) the size of the convolution filter, and \(V\) the target vocabulary size.

Implementation Our implementation is based on the seq2seq model implemented by TensorFlow (Abadi et al., 2015). It reuses some of its components, while adding a number of features, like a bidirectional encoder (Bahdanau et al., 2015), a beam-search decoder, a convolutional attention model and a hierarchical encoder (Chorowski et al., 2015). Our implementation also allows pre-training, multiple encoders (Zoph and Knight, 2016) and multi-task training (Luong et al., 2016). Contrary to the reference implementation by TensorFlow, which uses a bucket-based method with static (unrolled) RNNs, we use symbolic loops. This makes model creation much faster, and removes the need to manually configure bucket sizes.

Model settings We describe two models: one for text translation, and one for speech translation. Both models use a decoder with 2 layers of 256 LSTM units, with word embeddings of size 256. The speech model does not have source embeddings, as the input is already a sequence of vectors (of 40 MFCCs + frame energy). Both encoders are multi-layer bidirectional LSTMs, with 256 units. The text model’s encoder has 2 layers. The speech model has a hierarchical encoder with 3 layers, i.e., like Chorowski et al. (2015), the 2nd and 3rd layers read every other output from the previous layer, resulting in a sequence of \(\frac{1}{4}\)th the size of the input. For the text model, we use a standard attention model, while the speech model uses convolutional attention (with a single filter of size 25).

For training, we use the Adam algorithm with an initial learning rate of 0.001 (Kingma and Ba, 2014), and a mini-batch size of 64. We apply dropout (with a rate of 0.5) during training on the connections between LSTM layers in the encoder and decoder (Zaremba et al., 2014).

Our source code is available online at github.com/eske/seq2seq
Table 1: On the full corpus, there is on average 9.8 words/sent on the French side, and 9.3 on the English side. On the French audio side (with Agnes as a speaker), there is on average 281 frames/sent (2.8 sec/sent).

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Lines</th>
<th>French</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>Chars</td>
<td>Frames</td>
</tr>
<tr>
<td>BTEC train</td>
<td>19972</td>
<td>201k</td>
<td>1020k</td>
</tr>
<tr>
<td>BTEC dev</td>
<td>1512</td>
<td>12.2k</td>
<td>61.9k</td>
</tr>
<tr>
<td>BTEC test1</td>
<td>469</td>
<td>3.9k</td>
<td>19.5k</td>
</tr>
<tr>
<td>BTEC test2</td>
<td>464</td>
<td>3.8k</td>
<td>19.2k</td>
</tr>
</tbody>
</table>

Table 2: Results of the Text Translation task on BTEC, under different settings of the decoder. “Greedy” uses a greedy decoder. “BeamSearch” adds a beam-search decoder with a beam size of 8. “LM” adds an external trigram language model estimated on the target side of BTEC train. “Ensemble” uses a log-linear model with 5 NMT models trained independently (+ language model). The train corpus used in evaluation is a subset of 1000 sentences from BTEC train. For all the non-ensemble configurations, the best model out of 5 is used (according to its score on dev). The SMT baseline is a phrase-based model (Moses) trained on BTEC train, and tuned on BTEC dev.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>BLEU score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Greedy</td>
</tr>
<tr>
<td>dev</td>
<td>42.5</td>
</tr>
<tr>
<td>test1</td>
<td>39.8</td>
</tr>
<tr>
<td>test2</td>
<td>41.0</td>
</tr>
<tr>
<td>test1 (7 refs)</td>
<td>50.7</td>
</tr>
<tr>
<td>test2 (7 refs)</td>
<td>47.0</td>
</tr>
<tr>
<td>train.1000</td>
<td>78.4</td>
</tr>
</tbody>
</table>

3 Experimental Settings

As we do not have any good-sized speech translation corpus, we generated one with speech synthesis. We decided to use the French→English BTEC corpus, which is a small parallel corpus (see Table 1). BTEC (Basic Travel Expression Corpus) is very clean and contains short sentences (10 words/sentence on average), with a small vocabulary (9218 unique tokens on the French side, 7186 on the English side). This corpus also has multiple references (7 references per source sentence for test1 and test2), which we use for BLEU evaluation. We lowercase and tokenize all the data with the Moses tokenizer.

Synthetic speech was generated using Voxygen®, a commercial speech synthesis system, for 4 different female voices (Agnes, Marion, Helene, Fabienne) and 3 different male voices (Michel, Loic, Philippe). It is important to note that this is corpus-based concatenative speech synthesis and not parametric synthesis. So, for each speaker’s voice, speech utterances are generated by concatenation of units mined from a large speech corpus (generally around 3000 sentences/speaker). This means that despite having very little intra-speaker variability in our speech data, there is a realistic level of inter-speaker variability. To challenge the robustness of our system to inter-speaker variability, we concatenate the training data for 6 speakers, and leave the last speaker (Agnes) for evaluation only.

Like Chan et al. (2016), as input for our speech model, we segment the speech raw data into frames of 40 ms, with a step-size of 10 ms, and extract 40 MFCCs for each frame, along with the frame energy.

4 Experiments

Table 2 shows the results of our Machine Translation experiments. The main reason why we perform these experiments, is to assess whether it makes sense to train NMT models on such a small and specialized corpus. Our best system (an ensemble of 5 models with a language model) gives similar results in terms of BLEU score to a baseline SMT system. This is rather surprising, considering the small amount of data used. Also, even though the models seem to overfit quite a lot (see train.1000 scores), this does not seem to be a problem during evaluation.

[www.voxygen.fr](http://www.voxygen.fr)
Table 3: Results of the speech translation experiments on BTEC. The models are trained with 6 different speakers (including Michel). Agnes is not used for training. The Ensemble configuration uses 5 models trained independently. The baseline system uses a pipeline Google Speech ASR + SMT system trained on BTEC (we trained a new system on French without punctuation, as Google Speech API does not output any punctuation symbol, except when actually pronounced). The WER scores obtained by the baseline ASR system range between 23% and 26%. The last column from Table 2 provides a baseline assuming a perfect ASR.

5 Conclusion

We proposed a model for end-to-end speech translation, which gives promising results on a small French-English synthetic corpus. These results, however, are to be mitigated due to the synthetic nature of the data: there is not much variability (except across speakers), which makes it easier for the network to learn. In any case, the experiments show that our architecture is able to generalize quite well to a new speaker.

Future work includes the evaluation of this model on non-synthetic data. Some datasets that could be used are: TED talks (there exists a parallel corpus and a speech corpus, but their alignment is not straightforward); or books from the Gutenberg project (public-domain books in many languages, along with some human-read audiobooks). We also wish to evaluate this model in a multi-source setting: does it help when translating text to have access to a human-read version of it, or other kinds of information such as part-of-speech or images? 
References


