ANALYTICS CONSULTING
AIRBUS – ATC CHALLENGE
CIB ANALYTICS CONSULTING – OUR MISSIONS

- test: [https://analytics-cib.staging.echonet/search/](https://analytics-cib.staging.echonet/search/)
- contact us: Hugues EVEN [hugues.even@bnpparibas.com](mailto:hugues.even@bnpparibas.com)
GENERAL PROCESSING:

- 38 hours of data used for training and around 2 hours for dev
- Text processing with:
  - Mapping of ‘@’ (foreign words) and ‘_’ (not intelligible words) to ‘<unk>’ symbol
  - Form modification for anagrams (i.e. ILS → i-l-s) to be spelled correctly by the grapheme to phoneme (Seq-2-Seq model)
  - Merging words used as bigrams to be better integrated within the language model (i.e. air france → air_france)
First approach: Transfer Learning

Basis model:
- Hybrid model (HMM-DNN / Kaldi ‘chain’ model).
- 7 layers: 6 Tdnn layers and 2 final branches (chain and cross-entropy branch, which is not used at test time).
- Trained on about 1000 hours of data (mix of internal noisy data with public datasets such as CommonVoice etc...)

Transfer Learning:
- New ivector extractor
- Add new final branches
- Bigger LR for new layers than transferred layers (1.0 vs 0.25)
- Training for 2 epochs with only the challenge data (without augmentation and converted to 8kHz)
- Lexicon and LM (3-gram and 4-gram for rescoring) restricted to the challenge data.

Specialized LM // 10.93%
First approach: Transfer Learning

Basis model:
- Hybrid model (HMM-DNN / Kaldi ‘chain’ model).
- 7 layers: 6 Tdnn layers (with dim=1024) and 2 final branches (chain and cross-entropy branch, which is not used at test time).
- Trained on about 1000 hours of data (mix of BNPP noisy data with public datasets such as CommonVoice etc...)

Transfer Learning:
- New i-vector extractor
- Add new final branches
- Bigger LR for new layers than transferred layers (1.0 vs 0.25)
- Training for 2 epochs with only the challenge data (without augmentation and converted to 8kHz)
- Lexicon and LM (3-gram and 4-gram for rescoring) restricted to the challenge data.

Second approach: Training from scratch

- Trained on the challenge dataset augmented with: reverberation, speed perturbation and volume perturbation.
- Trained for 4 epochs (decrease of results with more epochs)
- Smaller model: 6 layers: 5 Tdnn layers (with dim=448) and 2 final branches.

WER Transfer
3-gram decoding // 11.27%
4-gram rescoring // 10.93%

WER model
3-gram decoding // 10.49%
4-gram rescoring // 10.22%
SEQ2SEQ MODEL

- The encoder first reads the transcript
- The decoder then outputs the callsign
- OpenNMT framework

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>lufthansa five three uniform turn right five degrees</td>
<td>lufthansa five three uniform</td>
</tr>
<tr>
<td>Source</td>
<td>Target</td>
</tr>
<tr>
<td>go ahead sir</td>
<td>nan</td>
</tr>
</tbody>
</table>

- Preprocessing:
  - Get rid characters not present in the callsign (‘<unk>’, ‘huh’ …)
  - Correct typos with the provided dictionary
- Train also on the predictions of the STT model if the callsign is in the prediction : extended dataset
- Embedding size and hidden size : 50
- Better results when retraining embeddings
- Bidirectional LSTM (1 layer)
- 15 Epochs

<table>
<thead>
<tr>
<th></th>
<th>WER</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provided dataset</td>
<td>0.1022</td>
<td>87.69%</td>
</tr>
<tr>
<td>Extended dataset</td>
<td>0.1022</td>
<td>86.3%</td>
</tr>
</tbody>
</table>
BI-LSTM + CRF MODEL

- Concatenation of final states of a bi-lstm on character embeddings (with dim=100) to get a character-based representation (of size 200) of each word
- Concatenation of this representation to a standard word vector representation (GloVe with dim=200 here) following [1]
- Embed the sentence with a bi-lstm and decode with a linear chain CRF

<table>
<thead>
<tr>
<th>Source</th>
<th>lufthansa five three uniform turn right five degrees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target</td>
<td>B-CS I-CS I-CS I-CS O O O O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WER</th>
<th>F1 (true transcript)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1022</td>
<td>87.62%</td>
</tr>
</tbody>
</table>

- No increase of the F1 score when trying ensembling methods for this task.

IF WE HAD MORE TIME...

- Add a branch for callsign extraction to the speech recognition network.
- Input phonemes probability distribution instead of transcription in the callsign extraction network.
- Train simultaneously both networks.

THANK YOU!