

Advances in ASR for schoolchildren

Nicolas Tirel

GreenAI U.P.P.A. x Prof en Poche

10-10-2022



Prof en Poche

Glossary

ASR Automatic Speech Recognition

STT Speech To Text

WER Word Error Rate

CER Character Error Rate

LM Language Model

- ① Previous seminar
- ② Updates
- ③ Results & comparison with Azure
- ④ Improvements & next steps

1 Previous seminar

Litterature

Model architecture

Data & results

2 Updates

3 Results & comparison with Azure

4 Improvements & next steps

Goal and challenges

Be able to recognize and understand children speech with science vocabulary in a classroom with an open-source solution. The model :

- Needs a lot of training data in the good context (children voice in classroom)

Goal and challenges

Be able to recognize and understand children speech with science vocabulary in a classroom with an open-source solution. The model :

- Needs a lot of training data in the good context (children voice in classroom)
- Will run on a smartphone or tablet (cloud or embedded)

Goal and challenges

Be able to recognize and understand children speech with science vocabulary in a classroom with an open-source solution. The model :

- Needs a lot of training data in the good context (children voice in classroom)
- Will run on a smartphone or tablet (cloud or embedded)
- **Must be below a specific size**

1 Previous seminar

Litterature

Model architecture

Data & results

2 Updates

3 Results & comparison with Azure

4 Improvements & next steps

DeepSpeech

Baidu Research Silicon Valley AI Lab

DeepSpeech: Scaling up end-to-end speech recognition

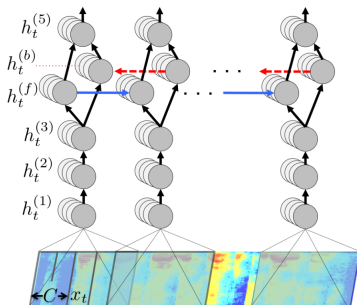


Figure 2: Structure of the RNN model and notation

[Hannun et al., 2014a]

Children SR in particular

Children speech recognition is challenging mainly due to the inherent high variability in childrens physical and articulatory characteristics and expressions. [Shivakumar and Georgiou, 2020]

End-to-end architectures trained on large amounts of adult speech data can help performance on children speech. Addition of large amounts of adult speech is found to benefit more when the acoustic mismatch is large between children and adults. Although, adaptation of acoustic model on children speech helps, the recognition performance remains more than 6 times worse compared to adult ASR. [Shivakumar and Narayanan, 2021]

Energy and carbon footprint E2E ASR

This work investigates for the first time the carbon cost of end-to-end automatic speech recognition (ASR). [...] With this study, we hope to raise awareness on this crucial topic and we provide guidelines, insights, and estimates enabling researchers to better assess the environmental impact of training speech technologies [Parcollet and Ravanelli, 2021]

- 1 Previous seminar
 - Litterature
 - Model architecture**
 - Data & results

- 2 Updates

- 3 Results & comparison with Azure

- 4 Improvements & next steps

Architecture

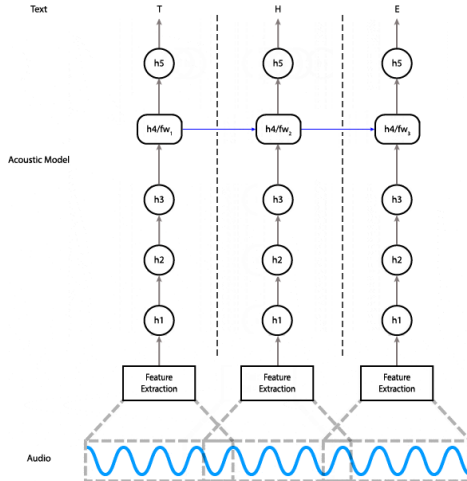


Figure 3: DeepSpeech model by Mozilla's team

Parameters

We can change those parameters to get a different result for each training :

- Alphabet (Character)

Parameters

We can change those parameters to get a different result for each training :

- Alphabet (Character)
- Language Model (Word)

Parameters

We can change those parameters to get a different result for each training :

- Alphabet (Character)
- Language Model (Word)
- Audio with transcription

Parameters

We can change those parameters to get a different result for each training :

- Alphabet (Character)
- Language Model (Word)
- Audio with transcription
- **Hyper-parameters**

1 Previous seminar

Litterature

Model architecture

Data & results

2 Updates

3 Results & comparison with Azure

4 Improvements & next steps

Main corpus

CommonVoice : a crowdsourcing project from Mozilla with the motivation to build a high quality, publicly open dataset. It has been started in early 2019, and get updated half a year

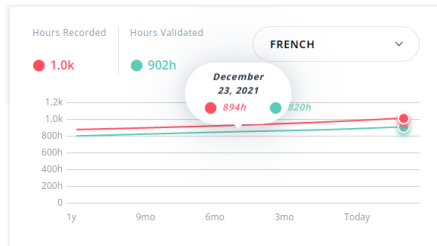


Figure 4: Evolution of the audio recorded and validated in French

Best model

Trained in three steps decreasing learning rate each time and for 40 epochs :

- CommonVoice 8 only with a learning rate of 0.001
- CommonVoice and **mathia** with a learning rate of 0.0001
- **mathia** only with a learning rate of 0.00005

Score (for a total of 28.25 kWh consumed)

WER: 0.187479, CER: 0.123425, loss: 12.353087

1 Previous seminar

2 Updates

From DeepSpeech to Coqui STT
Dataset
Specific Language Model

3 Results & comparison with Azure

4 Improvements & next steps

1 Previous seminar

2 Updates

From DeepSpeech to Coqui STT

Dataset

Specific Language Model

3 Results & comparison with Azure

4 Improvements & next steps

Same team

On the code owners file of DeepSpeech and Coqui STT, we find the same name of Alexandre Lissy (@lissyx) and Reuben Morais (@reuben), they only change the name of the structure



OUR STORY

In 2016 while at Mozilla the founders of Coqui noticed that speech technology was siloed in large corporations, leaving the open source world out in the cold. To remedy the situation we decided to take action!

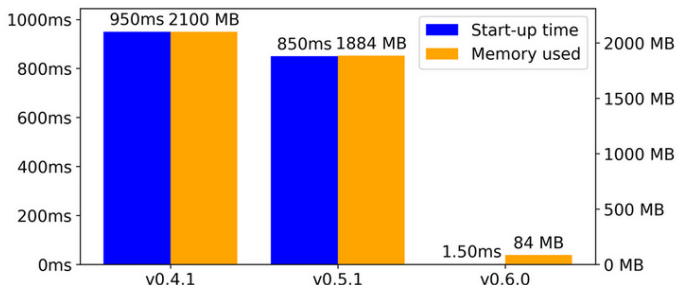
Over the intervening years we built open sourced STT and TTS engines which have been used by hundreds of thousands of people. Also, we kicked off projects open sourcing thousands of hours of speech training data. A vital, knowledgeable, and supportive community joined the cause and accelerated progress exponentially.

Now we're building these projects at Coqui, an organisation dedicated to continued support of these open source efforts and the community gathered around them.

Figure 5: Story of Coqui

But li(gh)te(r) and faster with TFLite

Coqui supports TensorFlow Lite allowing a transcription faster than real time on a Raspberry Pi 4 thanks to post-training quantization. The size of a model is now 47 MB instead of 188.



We now use **22 times less memory** and start up over **500 times faster**. Together with the optimizations we've applied to our language model, a complete Coqui STT package including the inference code and a trained English model is now more than **50% smaller**.

Figure 6: Smaller, faster, smarter

1 Previous seminar

2 Updates

From DeepSpeech to Coqui STT

Dataset

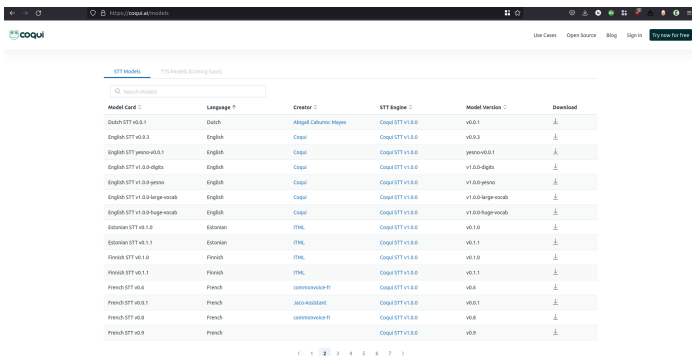
Specific Language Model

3 Results & comparison with Azure

4 Improvements & next steps

Coqui zoo

On the coqui website, we can find models trained by anyone, and shared with statistics like WER/CER/number of epochs/LM etc...



The screenshot shows the Coqui website's 'STT Models' page. The page has a search bar and a table of models. The table is sorted by language, with 'English' models appearing first. The table columns are: Model Card, Language, Creator, STT Engine, Model Version, and Download. The 'English' models include 'English STT v0.8.1', 'English STT v0.9.3', 'English STT yersno v0.0.1', 'English STT v1.0.0-digits', 'English STT v1.0.0-yersno', 'English STT v1.0.0-large-vocab', and 'English STT v1.0.0-huge-vocab'. The 'Dutch' models include 'Dutch STT v0.8.1'. The 'Estonian' models include 'Estonian STT v0.1.0' and 'Estonian STT v0.1.1'. The 'Finnish' models include 'Finnish STT v0.1.0' and 'Finnish STT v0.1.1'. The 'French' models include 'French STT v0.0', 'French STT v0.0.1', 'French STT v0.0', and 'French STT v0.9'. The table is paginated, showing 1 to 7 models.

Model Card	Language	Creator	STT Engine	Model Version	Download
Dutch STT v0.8.1	Dutch	Abigail Cabunoc Hayes	Coqui STT v1.0.0	v0.8.1	Download
English STT v0.9.3	English	Coqui	Coqui STT v1.0.0	v0.9.3	Download
English STT yersno v0.0.1	English	Coqui	Coqui STT v1.0.0	yersno-v0.0.1	Download
English STT v1.0.0-digits	English	Coqui	Coqui STT v1.0.0	v1.0.0-digits	Download
English STT v1.0.0-yersno	English	Coqui	Coqui STT v1.0.0	v1.0.0-yersno	Download
English STT v1.0.0-large-vocab	English	Coqui	Coqui STT v1.0.0	v1.0.0-large-vocab	Download
English STT v1.0.0-huge-vocab	English	Coqui	Coqui STT v1.0.0	v1.0.0-huge-vocab	Download
Estonian STT v0.1.0	Estonian	ITML	Coqui STT v1.0.0	v0.1.0	Download
Estonian STT v0.1.1	Estonian	ITML	Coqui STT v1.0.0	v0.1.1	Download
Finnish STT v0.1.0	Finnish	ITML	Coqui STT v1.0.0	v0.1.0	Download
Finnish STT v0.1.1	Finnish	ITML	Coqui STT v1.0.0	v0.1.1	Download
French STT v0.0	French	comnvoice-fr	Coqui STT v1.0.0	v0.0	Download
French STT v0.0.1	French	Jaco-Auslissant	Coqui STT v1.0.0	v0.0.1	Download
French STT v0.0	French	comnvoice-fr	Coqui STT v1.0.0	v0.0	Download
French STT v0.9	French		Coqui STT v1.0.0	v0.9	Download

Figure 7: Models sorted by language

Commonvoice & others

Even with a change in dataset and with the inclusion of new versions of CommonVoice, we don't always get better result

Test Corpus	WER	CER
African_Accented_French_test.csv	47.7%	6.6%
Att-HACK	12.9%	7.1%
M-AILABS	9.9%	3.3%
trainingspeech	10.9%	4.1%
Common Voice	31.5%	15.2%
LinguaLibre	67.6%	21.6%
MLS	22.6%	9.7%

Corpus tested	WER evolution	CER evolution	Conclusion
AAF	- 0.7 / +3.9	+ 0.6 / -18.2	+
Att-HACK	New / + 0.1	New / + 1.1%	+
M-AILABS	+ 2.5 / - 2.3	+ 1 / - 0.3	+
trainingspeech	- 7.9 / - 1.2	- 2 / + 0.1	✓
Common Voice	+ 6.9 / -6.5	+ 5.1 / -4.2	+
LinguaLibre	+ 53.4 / + 8.3	+ 19.5 / +3	×××
MLS	New / - 4.2	New / - 2.5	✓
CCPMF	Abandoned	Abandoned	

Figure 8: Result of the best french model and comparison with previous ones

Spontaneous dataset

To go further, we need to get a dataset as close as possible to the use case. We decided to validate unlabeled audios from the people using the app with transcription from Microsoft Azure

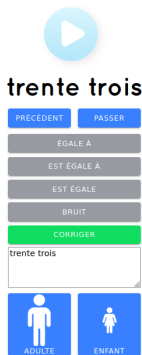


Figure 9: Validate audios and precise if there's noise ("bruit" in french)

Validation and sort

We listen more than 7000 audios :

- 5464 were validated

Validation and sort

We listen more than 7000 audios :

- 5464 were validated
- 3688 were children voices, 2h35

Validation and sort

We listen more than 7000 audios :

- 5464 were validated
- 3688 were children voices, 2h35
- 1476 with noise

1 Previous seminar

2 Updates

From DeepSpeech to Coqui STT
Dataset
Specific Language Model

3 Results & comparison with Azure

4 Improvements & next steps

What is a LM ?

A Language Model is created using a corpus of text, gets a sentence as input and returns the probability of the last word given all the previous words. It was used in 2014 for decoding CTC output with an important improve : an acoustic model could go from a WER of 35.8% to 14.1% [Hannun et al., 2014b] Really good explanation can be found here

Number LM

Once we know the specific vocabulary, i.e. be able to recognize numbers, yes, no, and some geometric shapes, we can write all of them in a file, and convert them using KenLM toolkit.

```
 Nicolas > training_lm > LM / $ LM_validate_withoutTime.txt
1  soixante trois
2  soixante quatre
3  soixante cinq
4  vingt quatre
5  soixante seize
6  quarante huit
7  quatre vingt treize
8  huit
9  neuf
10 huit
11 deux
12 trois
13 trois
14 soixante et un
15 quatre vingt dix neuf
16 soixante dix huit
17 vingt six
18 soixante deux
19 cinquante six
20 quarante neuf
21 trente quatre
22 cinquante deux
23 dix sept
24 huit cent onze
25 huit cent douze
26 huit cent douze
27 huit cent treize
28 huit cent quatorze
29 huit cent quatorze
30 huit cent quinze
31 huit cent dix huit
32 huit cent dix huit
33 six
34 six
35 un un
36 quinze
37 treize
38 neuf
39 treize
40 dix sept
41 quatorze
42 treize
```

Figure 10: LM from all the validated transcription

Alpha & Beta optimization

Two hyper parameters can be optimized with grid-search tries :
alpha, the weight of the language model and **beta** a compensation term.

Algorithm 1 Prefix Beam Search. The algorithm initializes the previous set of prefixes A_{prev} to the empty string. For each time step and every prefix ℓ currently in A_{prev} , we propose adding a character from the alphabet Σ to the prefix. If the character is a blank, we do not extend the prefix. If the character is a space, we incorporate the language model constraint. Otherwise we extend the prefix and incorporate the output of the network. All new active prefixes are added to A_{next} . We then set A_{prev} to include only the k most probable prefixes of A_{next} . The output is the 1 most probable transcript, although this can easily be extended to return an n -best list.

```

 $p_0(\emptyset; x_{1:0}) \leftarrow 1, p_{nb}(\emptyset; x_{1:0}) \leftarrow 0$ 
 $A_{prev} \leftarrow \{\emptyset\}$ 
for  $t = 1, \dots, T$  do
   $A_{next} \leftarrow \{\}$ 
  for  $\ell$  in  $A_{prev}$  do
    for  $c$  in  $\Sigma$  do
      if  $c = \text{blank}$  then
         $p_0(\ell; x_{1:t}) \leftarrow p(\text{blank}; x_t)(p_0(\ell; x_{1:t-1}) + p_{nb}(\ell; x_{1:t-1}))$ 
        add  $\ell$  to  $A_{next}$ 
      else
         $\ell^+ \leftarrow \text{concatenate } \ell \text{ and } c$ 
        if  $c = \ell_{eos}$  then
           $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)p_0(\ell; x_{1:t-1})$ 
           $p_{nb}(\ell; x_{1:t}) \leftarrow p(c; x_t)p_0(\ell; x_{1:t-1})$ 
        else if  $c = \text{space}$  then
           $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(W(\ell^+)|W(\ell))^{\alpha} p(c; x_t)(p_0(\ell; x_{1:t-1}) + p_{nb}(\ell; x_{1:t-1}))$ 
        else
           $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)(p_0(\ell; x_{1:t-1}) + p_{nb}(\ell; x_{1:t-1}))$ 
        end if
        if  $\ell^+$  not in  $A_{prev}$  then
           $p_0(\ell^+; x_{1:t}) \leftarrow p(\text{blank}; x_t)(p_0(\ell^+; x_{1:t-1}) + p_{nb}(\ell^+; x_{1:t-1}))$ 
           $p_{nb}(\ell^+; x_{1:t}) \leftarrow p(c; x_t)p_{nb}(\ell^+; x_{1:t-1})$ 
        end if
        add  $\ell^+$  to  $A_{next}$ 
      end if
    end for
  end for
   $A_{prev} \leftarrow k \text{ most probable prefixes in } A_{next}$ 
end for
return 1 most probable prefix in  $A_{prev}$ 

```

Figure 11: Beam search using Language Model [Hannun et al., 2014b]

- 1 Previous seminar
- 2 Updates
- 3 Results & comparison with Azure
- 4 Improvements & next steps

Azure precision

Once we validated enough audio (20% of all our dataset, +7000 audios), we can evaluate the score of Azure, and try to get same precision with our models.

	Children (WER)	Children (CER)	Adult (WER)	Adult (CER)
Without noise	6.9 %	5.09 %	4.2 %	3.1 %
With noise	20.8 %	17.4 %	17.2 %	15.0 %

Figure 12: Results of validated audio with Microsoft Azure

Previous model

Trained in three steps decreasing learning rate each time and for 40 epochs :

- CommonVoice 8 only with a learning rate of 0.001
- CommonVoice and **mathia** with a learning rate of 0.0001
- **mathia** only with a learning rate of 0.00005

Score (for a total of 28.25 kWh consumed)

WER: 0.187479, CER: 0.123425, loss: 12.353087

Different tries

In order to get the best model in WER, I tried different trainings...

- Fine-tuning of last model with specific Language Model
- Evaluation on different dataset, azure children, adult and mathia
- Optimization from other models
- Grid search for LM alpha and beta parameters
- Data augmentation with overlay, reverb, pitch, tempo, volume...

Best model

Firstly trained with CommonVoice 8, then with both **validated audios**, and the **mathia** corpus, using a specific Language Model and the best alpha and beta hyper-parameters

Score

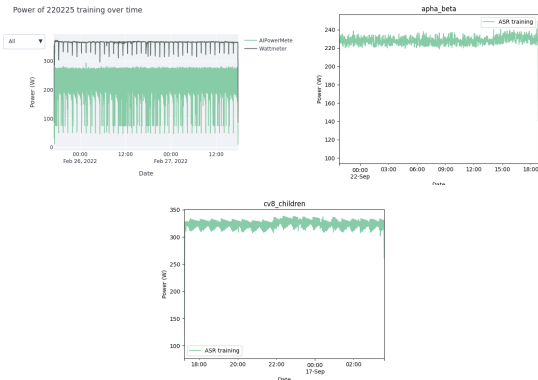
WER: 10.98%, CER: 07.19%, loss: 08.99 - **Mathia**

WER: 14.75%, CER: 11.49%, loss: 07.31 - **New audios w/o noise**

WER: 33.95%, CER: 29.55%, loss: 16.20 - **New audios w/ noise**

Consumption

Two thirds of the total consumption is due to the training part, 2 days 00:43:26 and 17.81 kWh, then fine-tuning (10:21:00 3.34 kWh) and optimization (21:00:00 4.78 kWh), in total, the best model has consumed **25,93 kWh in 80 hours**



Demonstration

Let's see how it looks like with a streamlit dashboard !

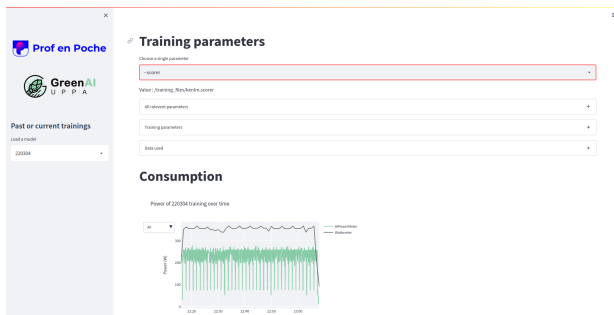


Figure 13: Screenshot of the dashboard, for those only reading the slides

- 1 Previous seminar
- 2 Updates
- 3 Results & comparison with Azure
- 4 Improvements & next steps
Energy and emission

Ideas to improve our results

- Annotate more audio

Ideas to improve our results

- Annotate more audio
- Increase data with data adaptation

Ideas to improve our results

- Annotate more audio
- Increase data with data adaptation
- Try different model (transformers like wav2vec [Schneider et al., 2019])

Future work

- Measure the consumption in inference

Future work

- Measure the consumption in inference
- Compare different models on the WER and energy consumption during training and inference

Future work

- Measure the consumption in inference
- Compare different models on the WER and energy consumption during training and inference
- Develop an embedded solution and compare consumption

- 1 Previous seminar
- 2 Updates
- 3 Results & comparison with Azure
- 4 Improvements & next steps
Energy and emission

Wattmeter

In addition, the machine used for all my work at Prof en Poche is plugged to a watt-meter which measures the power used by the whole machine instead of only the CPU/GPU. We just have to integrate over time to get the energy consumption in Joules or Watt-hours.

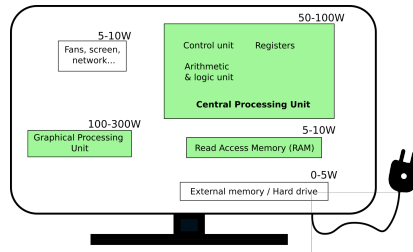


Figure 14: Sources of energy consumption in a computer

Training has a huge impact

When using our three machines, we can see a huge increase during training, and one of them consumes around 100 kWh for a single training. The emission related is highly dependant of the country of production, in France with 60 grams per kWh we get 6 kg of CO₂e emissions, but if we did this in Poland it rises to 73 kg !
[Ritchie et al., 2020]

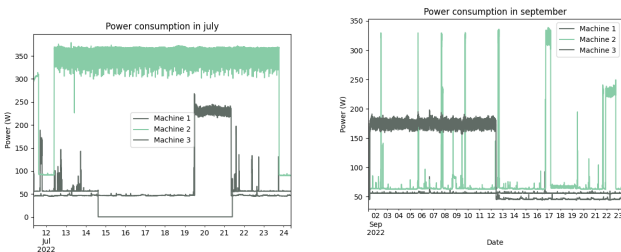


Figure 15: Power consumption of three machines in July and September

Electricity is not the only impact

When talking about numeric emission, we always think about the electricity or the data centers, but we need to think also about the fabrication process, which is responsible of 80% of the footprint in the life cycle assessment [Déragne and Mouneu, 2020]

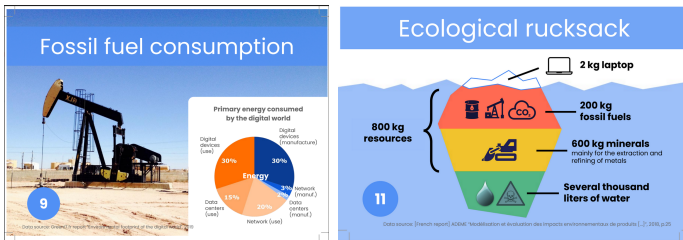


Figure 16: Two cards from The Digital Collage

To conclude

If you want to go further and take concrete actions :

- Measure your carbon footprint
- Become a player of the change : participate in The Digital Collage, keep your numeric equipment 10 years at least, avoid buying new equipment as possible
- Read the IPCC reports, "L'âge des low tech" Philippe Bihouix, watch "Ruée minière au XXI^e siècle : jusqu'où les limites seront-elles repoussées ?" - Aurore Stephant at USI...

Technology will NOT save us

Thanks!

References I

[Déragne and Mouneu, 2020] Déragne, A. and Mouneu, Y. (2020).

The digital collage.

[Source here.](#)

[Hannun et al., 2014a] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. (2014a).

Deep speech: Scaling up end-to-end speech recognition.

[Hannun et al., 2014b] Hannun, A. Y., Maas, A. L., Jurafsky, D., and Ng, A. Y. (2014b).

First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns.

References II

[Parcollet and Ravanelli, 2021] Parcollet, T. and Ravanelli, M. (2021).

The energy and carbon footprint of training end-to-end speech recognizers.

[Ritchie et al., 2020] Ritchie, H., Roser, M., and Rosado, P. (2020).

Energy.

Our World in Data.

<https://ourworldindata.org/energy>.

[Schneider et al., 2019] Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019).

wav2vec: Unsupervised pre-training for speech recognition.

References III

[Shivakumar and Georgiou, 2020] Shivakumar, P. G. and Georgiou, P. (2020).

Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations.

[Shivakumar and Narayanan, 2021] Shivakumar, P. G. and Narayanan, S. (2021).

End-to-end neural systems for automatic children speech recognition: An empirical study.