Multi-style training for South African call centre audio

W. Heymans, M.H. Davel and C. van Heerden

8 December 2021

Overview

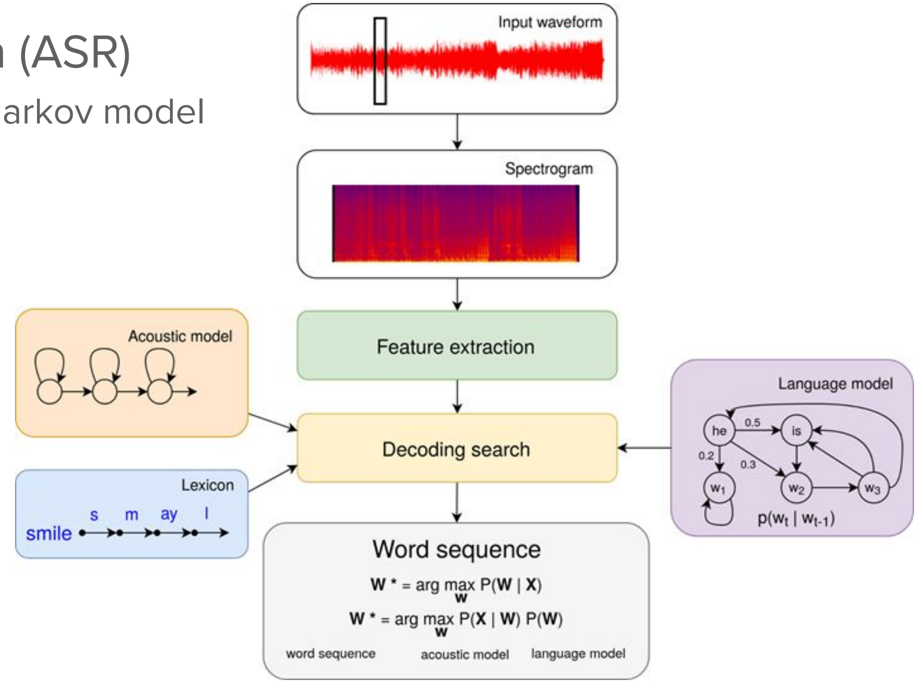
- Introduction to automatic speech recognition (ASR)
- Background
- Datasets
- Experimental setup
- Experiments in a controlled environment
  - Measure isolated effects
  - Speed, volume, noise, sampling rate, encoding
- Experiments on a real-world dataset (South African call centres)
- Conclusion

Introduction

- Automatic speech recognition (ASR) is a technology that enables computer systems to convert spoken language into text using an automated process.
- Actively researched since the 1970s.
- Word error rate (WER) has been significantly improved.
  - Advancements in deep learning
  - Increased computational power of modern computers
  - Large amounts of data
- Important applications
  - Human-to-machine communication
  - Voice search
  - Digital assistants
  - Call centre speech analytics

Introduction

- Automatic speech recognition (ASR)
    - Deep neural network - hidden Markov model
- Input waveform
- Fourier transform
- Feature extraction
- Acoustic model
- Lexicon
- Language model
- Decoding



Background

- Mismatched training and testing data
    - Background noise
    - Microphone distortion
    - Different recording environments
    - Encoding noise
    - Sampling rate mismatch
    - Speaking styles and accents

Background

- Difficult to generalise to new audio if the mismatch is not handled
- Multi-style training
    - Transform training data to better represent test conditions
    - Learn robust representations of the data
    - Add a series of styles:
        - Speed
        - Volume
        - Noise
        - Sampling rate

Datasets: South African Call Centre corpus

- Proprietary call centre corpus
- Contains mostly South African English
- Single channel recordings
- Encoded with WAV49 encoding

Table 1. SACC corpus subsets with sampling rate, encoding and total duration.

Dataset	Sampling rate	Encoding	Hours
train	8 kHz	-	48.8
train-e	8 kHz	WAV49	48.8
dev	8 kHz	-	7.1
dev-e	8 kHz	WAV49	7.1
test	8 kHz	-	6.2
test-e	8 kHz	WAV49	6.2
held-out test	8 kHz	WAV49	1.2

Datasets: LibriSpeech corpus

- Creating a controlled environment
- 1,000 Hours of English audiobook recordings
- We use the 100 hour subset (train-clean-100)
- Public availability - benchmark for many ASR systems
- 16 kHz sampling rate
- Simulate call centre conditions
  - Adding noise using QUT-Noise corpus (for dev and test sets)
  - Reduce sampling rate to 8 kHz and encode with WAV49
  - Create training datasets using Musan Noise corpus (artificial mismatch)

Datasets: LibriSpeech corpus

Table 2. Multi-style training datasets created using the 100 hour clean LibriSpeech subset (*train-clean-100*).

Dataset name	Encoding	Noise corpus	SNR	Speed	Volume
train-clean	-	-	-	-	-
train-clean-8k	-	-	-	-	-
train-clean-e	WAV49	-	-	-	-
train-noisy-e-5	WAV49	QUT	5	-	-
train-clean-e-s	WAV49	-	-	10%	-
train-clean-e-v	WAV49	-	-	-	20%
train-clean-e-sv	WAV49	-	-	10%	20%
train-musan-e-5	WAV49	Musan	5	-	-
train-musan-e-10	WAV49	Musan	10	-	-
train-musan-e-15	WAV49	Musan	15	-	-
train-musan-e-20	WAV49	Musan	20	-	-
train-musan-e-15-s	WAV49	Musan	15	10%	-
train-musan-e-15-v	WAV49	Musan	15	-	20%
train-musan-e-15-sv	WAV49	Musan	15	10%	20%

Datasets: LibriSpeech corpus

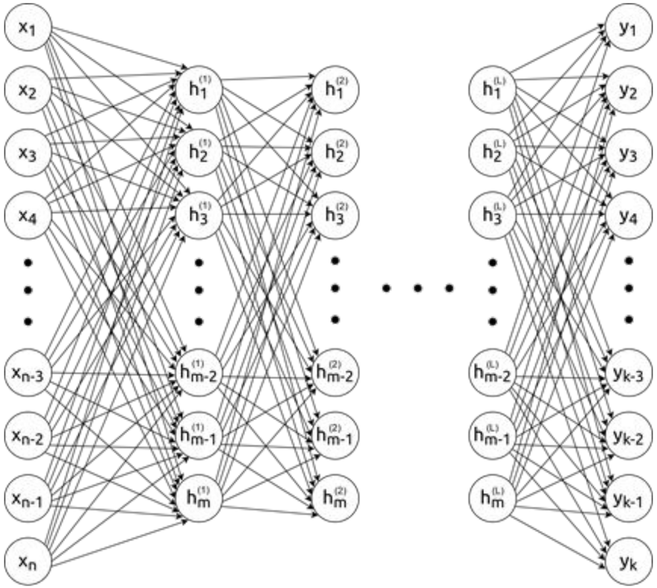
- Development sets
- Test set

Table 3. Development and test datasets created using the LibriSpeech *dev-clean* and *test-clean* sets.

Dataset name	**Source dataset**	**Encoding**	**Noise corpus**	**SNR**	**Hours**
dev-clean-e	dev-clean	WAV49	-	-	5.4
dev-noisy-e-5	dev-clean	WAV49	QUT	5	5.4
test-noisy-e-5	test-clean	WAV49	QUT	5	5.4

Experimental Setup

- Pytorch-Kaldi ASR toolkit
- CD-DNN-HMM ASR system
- Default setup of Pytorch-Kaldi toolkit
- Acoustic model
  - 5 Hidden-layer network (1 024 units)
  - ReLU activation functions
  - Batch normalisation
  - Dropout
- fMLLR input features
- Train all networks until convergence
- Grid search to optimise hyperparameters



Experiments: Controlled environment

- LibriSpeech corpus
- Different sampling rates
- Evaluate matrix of networks (encoding and different sampling rates)

Table 4. WER results of models with different sampling rates on *dev-clean* and *dev-clean-e*. Average WER (%) and standard error is shown over 3 seeds.

Train set	Sampling rate	dev-clean	dev-clean-e
Wide-band			
train-clean	16 kHz	**8.88 ± 0.10**	19.32 ± 0.11
train-clean-e	16 kHz[6]	10.71 ± 0.05	**11.02 ± 0.03**
Narrow-band			
train-clean	8 kHz	10.29 ± 0.03	11.44 ± 0.04
train-clean-e	8 kHz	10.76 ± 0.02	11.21 ± 0.03

Experiments: Controlled environment

- Reduced sampling rate
- Encoding only
- Speed perturbation
- Noise perturbation
- Combination
- Matched training dataset

Table 5. WER on development set (*dev-noisy-e-5*) using training datasets with different styles. Average WER (%) and standard error is shown over 3 seeds.

Model	Datasets	Size	Dev WER
Variations of clean set			
train-clean	train-clean	1	36.46 ± 0.14
train-clean-8k[7]	train-clean @ 8 kHz	1	33.23 ± 0.21
train-clean-e	train-clean-e	1	$\mathbf{28.06 \pm 0.03}$
Speed and volume			
train-clean-e-s	train-clean-e + s	2	$\mathbf{27.83 \pm 0.02}$
train-clean-e-v	train-clean-e + v	2	28.21 ± 0.07
train-clean-e-sv	train-clean-e + sv	2	28.25 ± 0.10
train-clean-e-s-v	train-clean-e + s + v	3	28.04 ± 0.13
Noise			
train-musan-e-5	train-musan-e-5	1	29.29 ± 0.34
train-musan-e-10	train-musan-e-10	1	27.29 ± 0.12
train-musan-e-15	train-musan-e-15	1	$\mathbf{23.30 \pm 0.06}$
train-musan-e-20	train-musan-e-20	1	26.64 ± 0.09
Speed, volume and noise			
train-musan-e-15-s	train-musan-e-15 + s	2	24.09 ± 0.05
train-musan-e-15-v	train-musan-e-15 + v	2	23.97 ± 0.11
train-musan-e-15-sv	train-musan-e-15 + sv	2	24.34 ± 0.02
train-musan-e-15-s-v	train-musan-e-15 + s + v	3	$\mathbf{23.80 \pm 0.09}$
train-musan-e-15-s-v-sv	train-musan-e-15 + s + v + sv	4	23.89 ± 0.08
Matched noise			
train-noisy-e-5	train-noisy-e-5	1	$\mathbf{19.75 \pm 0.04}$

Experiments: Controlled environment

- Test set

Table 6. WER on test set (*test-noisy-e-5*) using training datasets with different styles. Average WER (%) and standard error is shown over 3 seeds.

Model	Datasets	Size	Test WER
Variations of clean set			
train-clean	train-clean	1	36.92 ± 0.18
train-clean-e	train-clean-e	1	29.16 ± 0.12
Speed and volume			
train-clean-e-s	train-clean-e + s	3	28.61 ± 0.11
Noise			
train-musan-e-15	train-musan-e-15	1	$\mathbf{24.54 \pm 0.22}$
Speed, volume and noise			
train-musan-e-15-s-v	train-musan-e-15 + s + v	3	24.87 ± 0.08
Matched noise			
train-noisy-e-5	train-noisy-e-5	1	$\mathbf{20.48 \pm 0.03}$

Experiments: Controlled environment

- Increased network size

Table 7. WER on development (*dev-noisy-e-5*) and test set (*test-noise-e-5*) using MLP acoustic models with 2 048 hidden units per layer. Average WER (%) and standard error is shown over 3 seeds.

Model	**Size**	**Dev WER**	**Test WER**
Encoded			
train-clean-e (1 024x5)	1	28.06 ± 0.03	29.16 ± 0.12
train-clean-e (2 048x5)	1	26.82 ± 0.08	27.74 ± 0.14
Noise			
train-musan-e-15 (1 024x5)	1	23.30 ± 0.06	24.54 ± 0.22
train-musan-e-15 (2 048x5)	1	23.15 ± 0.10	24.55 ± 0.04
Speed, volume and noise			
train-musan-e-15-s-v (1 024x5)	3	23.80 ± 0.09	24.87 ± 0.08
train-musan-e-15-s-v (2 048x5)	3	$\mathbf{22.81 \pm 0.06}$	$\mathbf{24.34 \pm 0.08}$

Experiments: Real-world dataset

- Call centre data
- Baseline trained using unencoded data
- Encoded model
- MTR model (unencoded, encoded and speed perturbed)

Table 8. WER results on dev/test sets for the SACC corpus. Average WER (%) is shown over 3 seeds.

Model	dev	dev-e	test	test-e	held-out test
train	28.41	28.91	33.14	33.43	41.90
train-e	28.40	28.63	33.36	33.04	41.80
MTR	**27.98**	**28.19**	**32.77**	**32.46**	**41.42**

Conclusion

- MTR is very useful if the styles are chosen appropriately.
    - Speed and volume may improve performance, given enough capacity.
    - Noise and encoding was the most effective.
- Matched training data is still much better than MTR.
- With proper network capacity MTR does not hurt performance.
- Small consistent improvements are observed in matched datasets.
- Very large improvements achieved on mismatched datasets.

Questions?