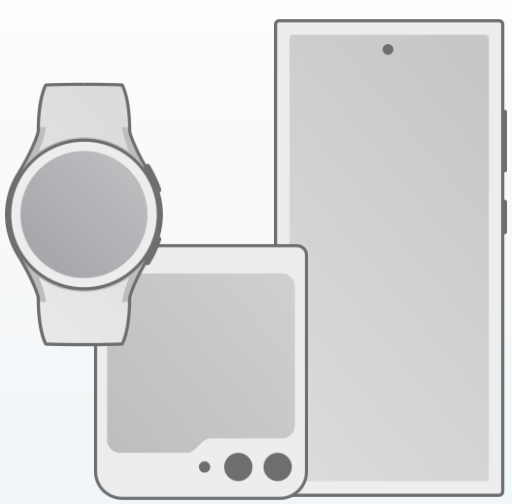
Samsung Research

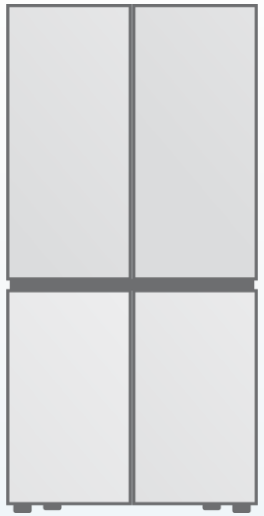
Unsupervised On-device Adaptation of a Speech Recogniser and the Pitfalls of "SpeechLLM" Evaluation

AI Center Cambridge

April 17th, 2025

Speech Recognition





Can happen in the cloud or on-device.

Contents

Challenges and opportunities.

Hardware constraints.

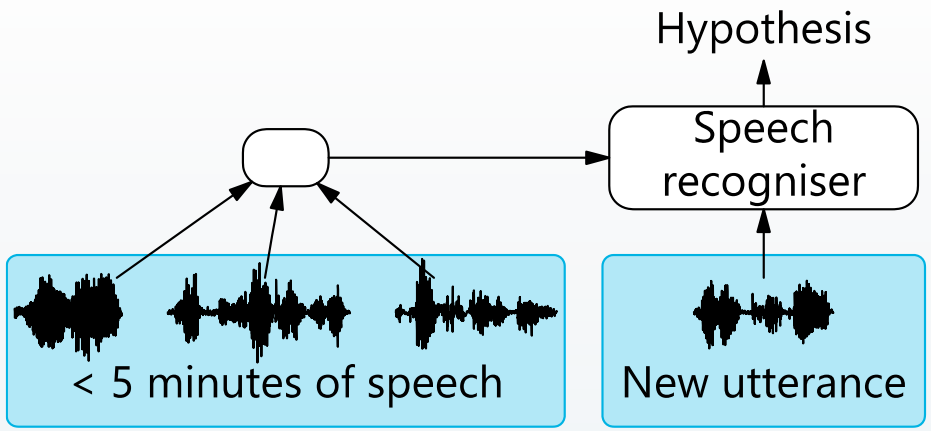
Real-life data, but limited and potentially noisy.

No labels.

The more a user interacts with the speech recogniser, the higher the accuracy should be.

Personalisation.

The more a user interacts with the speech recogniser, the higher the accuracy should be.



- Adapt the speech recogniser on previous utterances to improve accuracy on the next ones.
- Unsupervised adaptation as transcriptions are not provided by the user.

Existing possibilities

Full fine-tuning

Aspect	Options
Adaptation type	Supervised
Speaker representation	Learned
Number of parameters	1M to inf.
Adaptation loss	Maximum (conditional) likelihood

Existing possibilities

LoRA

Aspect	Options
Adaptation type	Supervised
Speaker representation	Learned
Number of parameters	100k to 1M
Adaptation loss	Maximum (conditional) likelihood

Proposal

Speaker codes and/or minimum entropy

Aspect	Options
Adaptation type	(un)supervised
Speaker representation	Learned
Number of parameters	1024
Adaptation loss	Maximum (conditional) likelihood or Minimum entropy

LoRA

Aspect	Options
Adaptation type	Supervised
Speaker representation	Learned
Number of parameters	100k to 1M
Adaptation loss	Maximum (conditional) likelihood

One vector or “code” per speaker.

Proposal

Speaker codes and/or minimum entropy

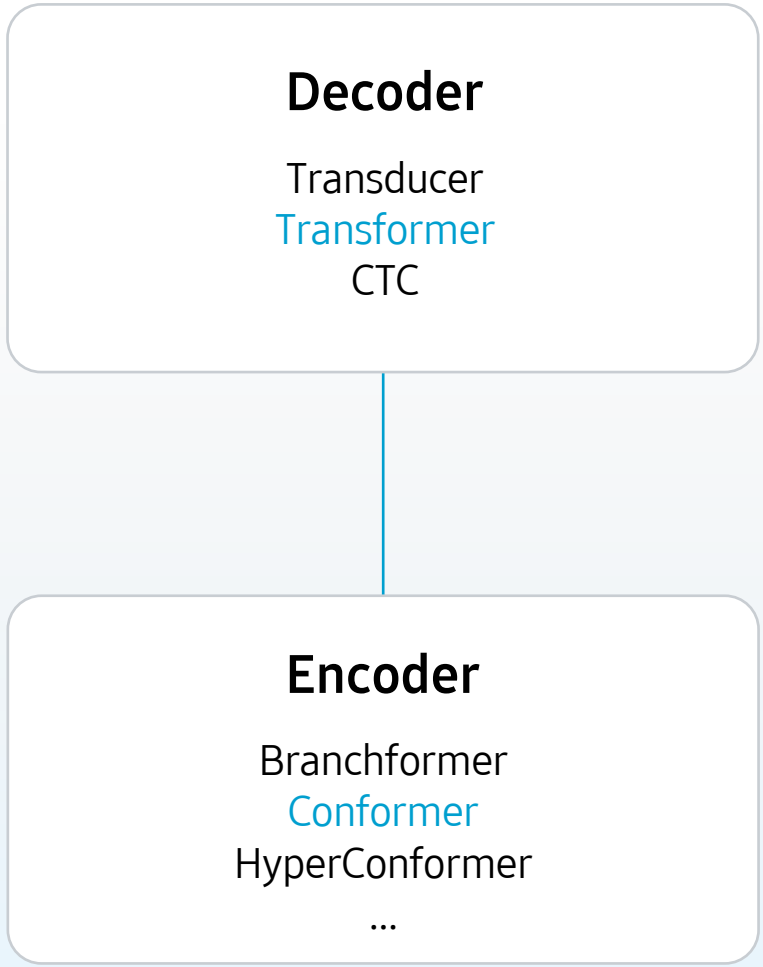
Aspect	Options
Adaptation type	(un)supervised
Speaker representation	Learned
Number of parameters	1024
Adaptation loss	Maximum (conditional) likelihood or Minimum entropy

Speaker codes are not new, Abdel-Hamid et al. proposed them in 2013 for DNN-HMM speech recogniser. We are revisiting them for modern speech recognisers and adapting it to our setting.

Adapting a speech recogniser with speaker codes

Step 0: the architecture

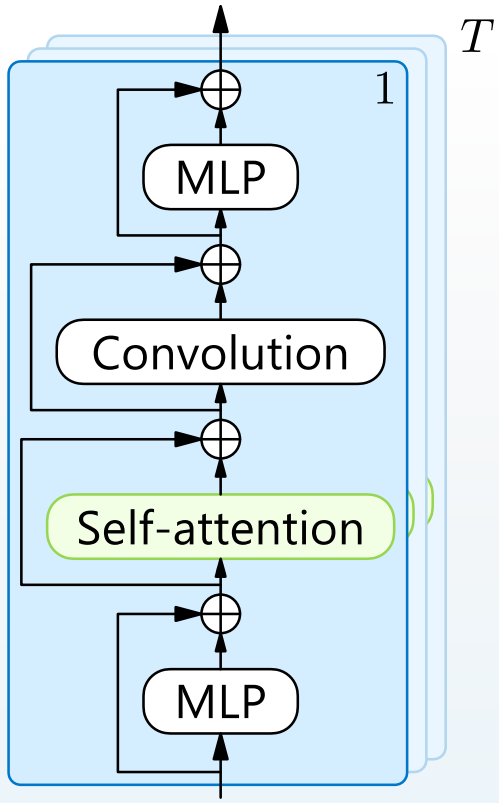
The method is agnostic to the framework, but we will focus on the a conformer encoder-decoder



Adapting a speech recogniser with speaker codes

Step 1: modifying the architecture

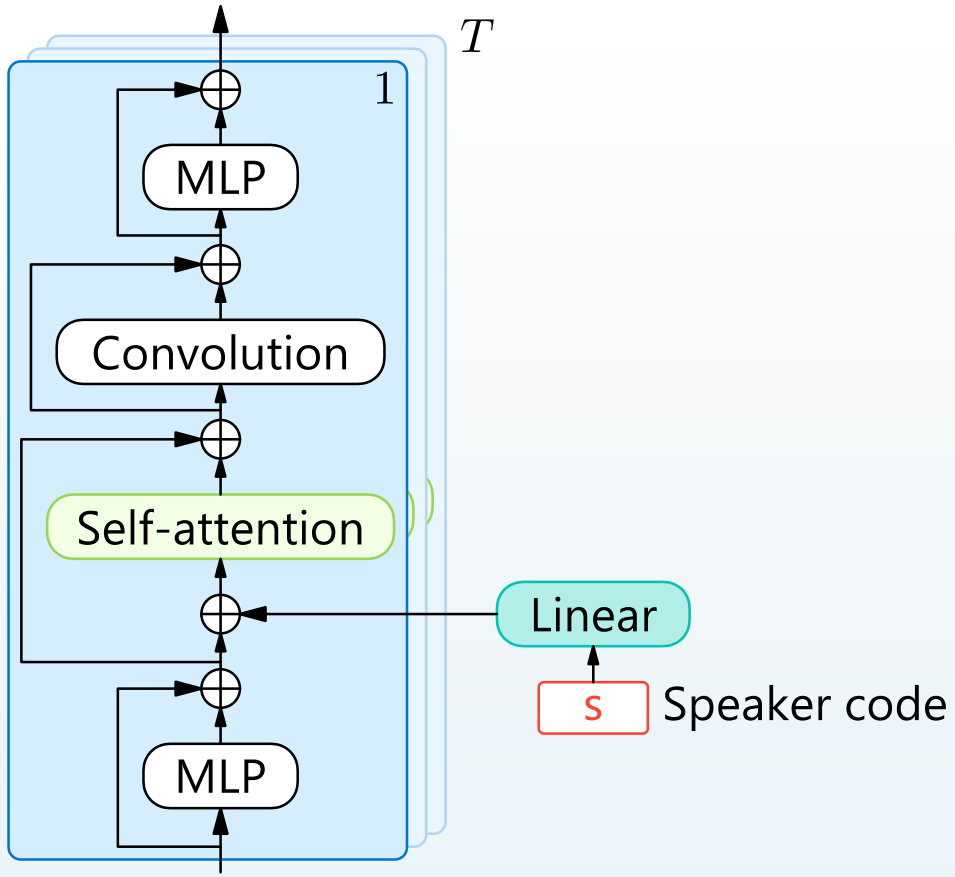
The « Conformer » speech encoder:
(the decoder is a standard Transformer)



Adapting a speech recogniser with speaker codes

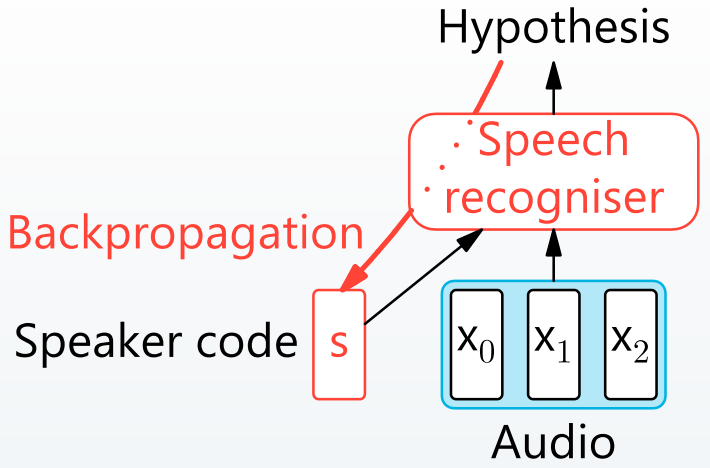
Step 1: modifying the architecture

The « Conformer » speech encoder:
(the decoder is a standard Transformer)



Adapting a speech recogniser with speaker codes

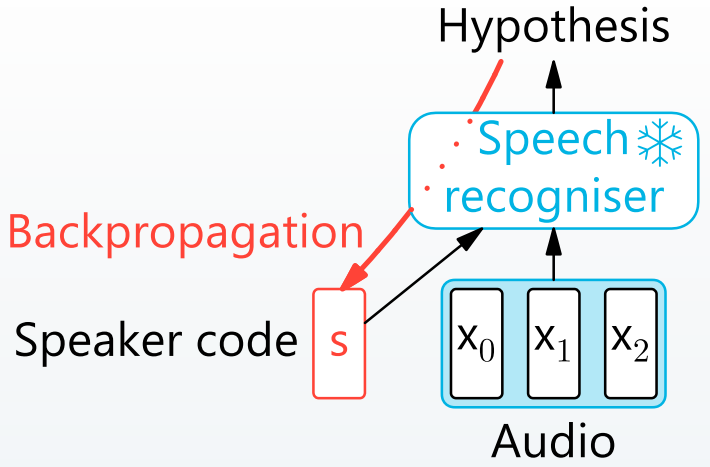
Step 2: supervised training of the speaker dependent speech recogniser



- Each speaker in the dataset gets assigned its speaker code (simple embedding layer).
- Supervised training is done on “large” amount of transcribed speech.
- During this step, speaker codes learn to represent the space of speaker variability.
  - Not only the way this speaker speaks, but also noises, microphones…

Adapting a speech recogniser with speaker codes

Step 3: unsupervised (on-device) adaptation



- All parameters are frozen, except the speaker code embedding.
- The speaker code is initialised with zeros and adapted.
- Unsupervised – The hypothesis is a weak point.
- Maximum (conditional) likelihood or minimum entropy.

Minimum entropy

- Unsupervised learning is a problem for discriminative models.
- Usually: run a model – “pseudo-label” – adaptation .

Minimum entropy

- Unsupervised learning is a problem for discriminative models.
- Usually: run a model – "pseudo-label" – adaptation .
- **Proposal:** minimise the conditional entropy (Grandvalet & Bengio 2004).
  - ✓ *Makes the adaptation more robust to errors in the initial hypothesis.*

$$\mathrm{H}(q_\theta) = -\sum_{X \in D} \sum_{\boldsymbol{w}} P(\boldsymbol{w}|X) \log P(\boldsymbol{w}|X) dX.$$

Minimum entropy

- Unsupervised learning is a problem for discriminative models.
- Usually: run a model – "pseudo-label" – adaptation .
- **Proposal:** minimise the conditional entropy (Grandvalet & Bengio 2004).
  - ✓ *Makes the adaptation more robust to errors in the initial hypothesis.*

$$\mathrm{H}(q_\theta) = -\sum_{X \in D} \sum_{\boldsymbol{w}} P(\boldsymbol{w}|X) \log P(\boldsymbol{w}|X) dX.$$

Speech recogniser

Minimum entropy

- Unsupervised learning is a problem for discriminative models.
- Usually: run a model – "pseudo-label" – adaptation .
- **Proposal:** minimise the conditional entropy (Grandvalet & Bengio 2004).
  - ✓ *Makes the adaptation more robust to errors in the initial hypothesis.*

$$\mathrm{H}(q_\theta) = -\sum_{X \in D} \sum_{\boldsymbol{w}} P(\boldsymbol{w}|X) \log P(\boldsymbol{w}|X) dX.$$

Speech recogniser

Each utterance in the dataset

Minimum entropy

- Unsupervised learning is a problem for discriminative models.
- Usually: run a model – "pseudo-label" – adaptation .
- **Proposal:** minimise the conditional entropy (Grandvalet & Bengio 2004).
  - ✓ *Makes the adaptation more robust to errors in the initial hypothesis.*

$$\mathrm{H}(q_\theta) = -\sum_{X \in D} \sum_{\boldsymbol{w}} P(\boldsymbol{w}|X) \log P(\boldsymbol{w}|X) dX.$$

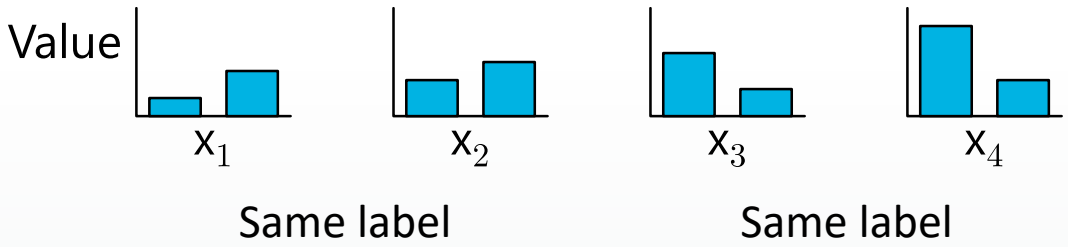
Speech recogniser

Each utterance in the dataset

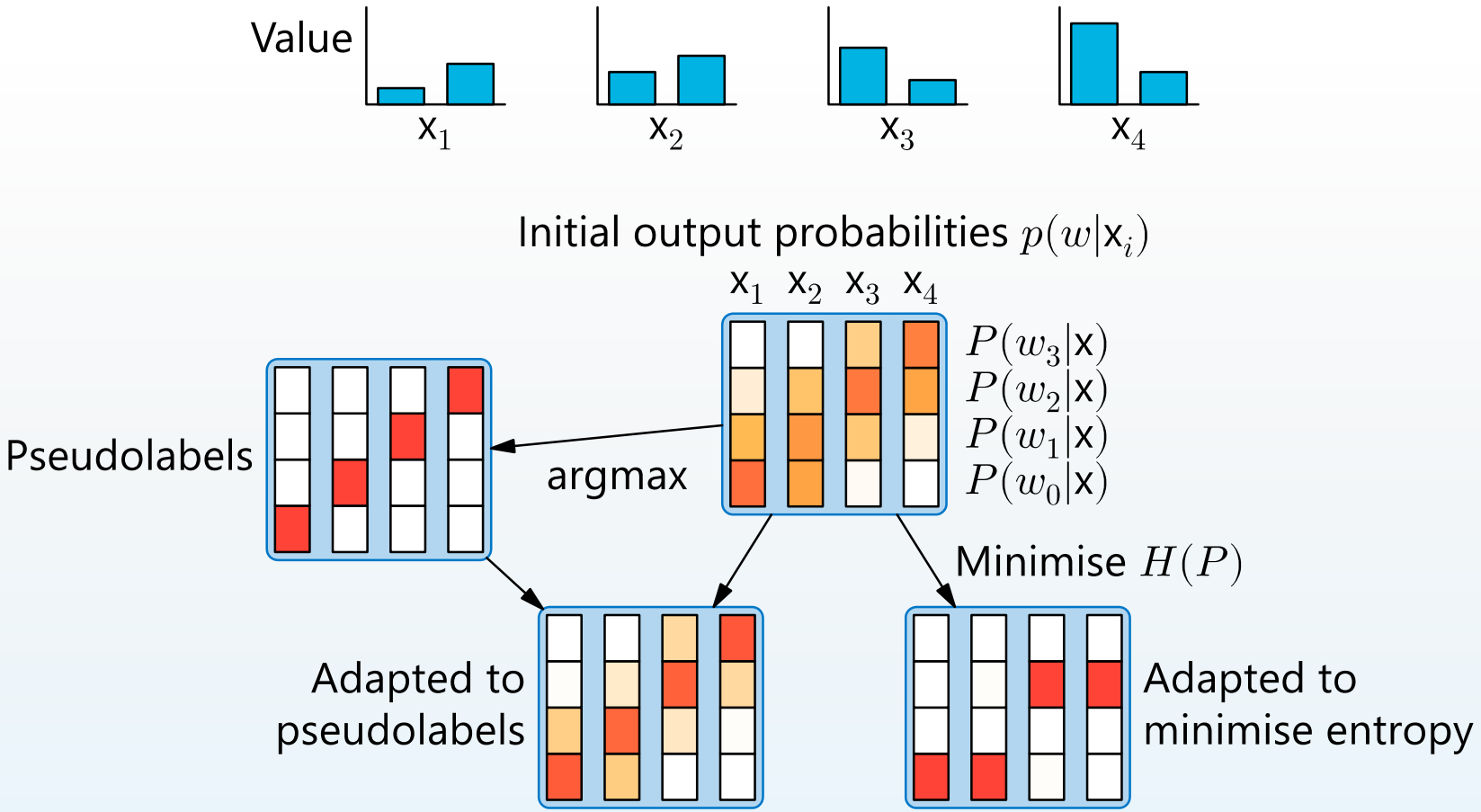
A sum over all possible word sequences

Intractable, so we use the n-best sequences instead

Minimum entropy – an example



Minimum entropy – an example



The data

- Enough transcribed hours.
- Enough speakers with different acoustic environments.
- Challenging acoustic conditions.

The data

- Enough transcribed hours (850 hours).
- Enough speakers with different acoustic environments (1370 + 100 speakers).
  - *Each speaker has at least 10 minutes of available speech.*
- Challenging acoustic conditions (Musan noises + reverberation).

Build our own dataset from Common Voice 18 with Musan noises and musics rendered in a virtual room (reverberation).

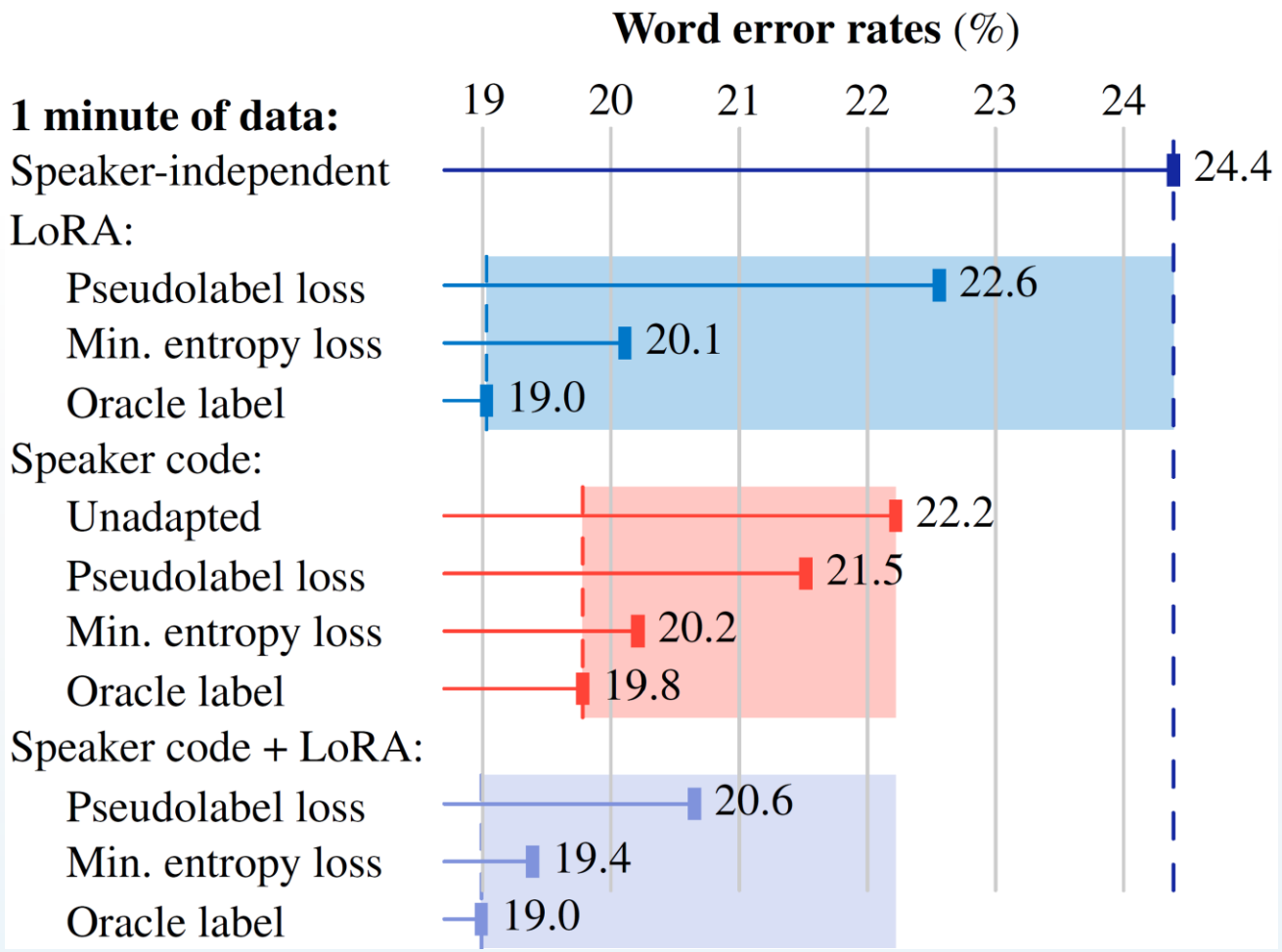
A few more experimental details

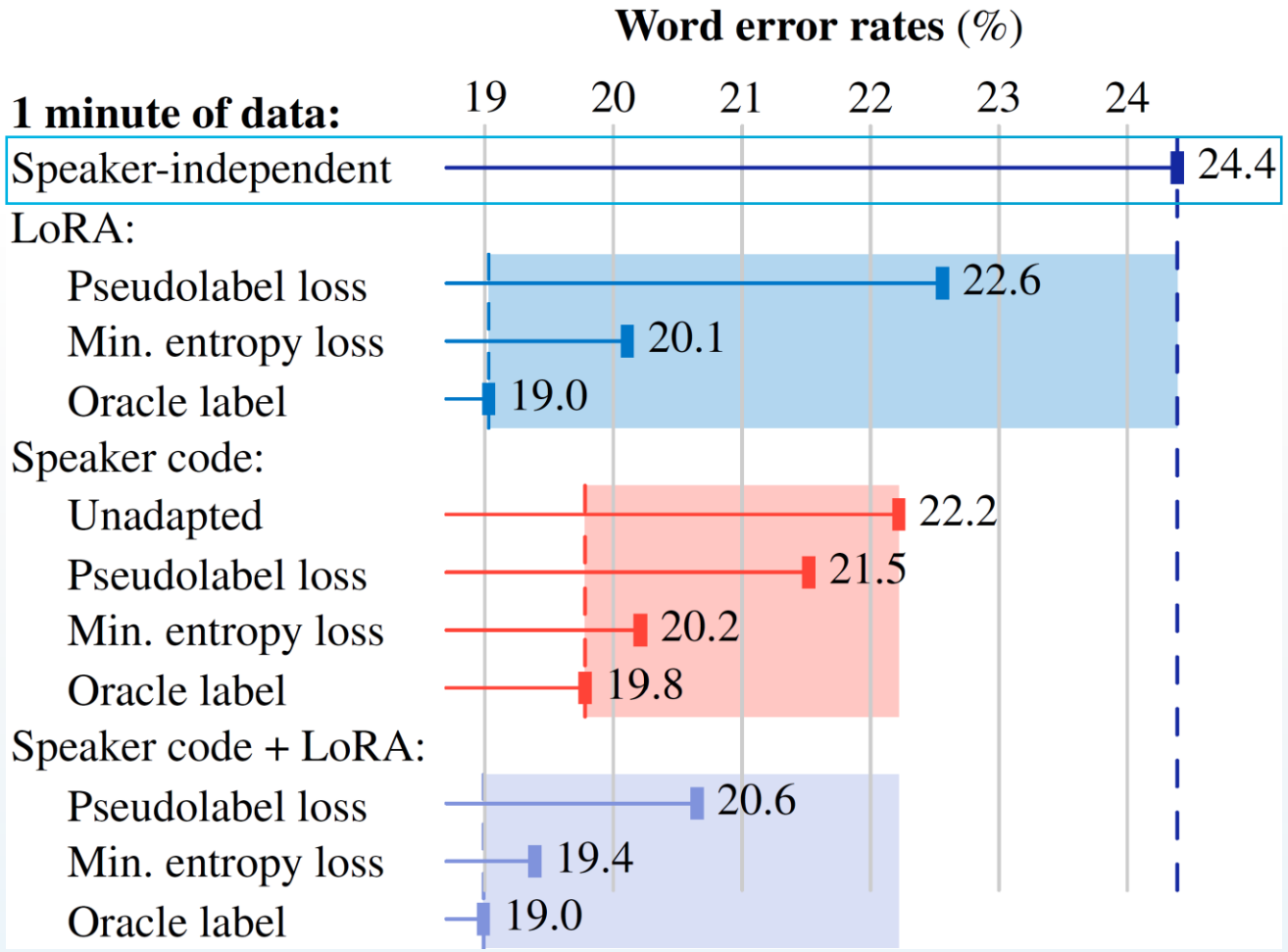
- **Base model:**
  100M parameters Conformer CTC-Attention from SpeechBrain.

- **Adaptation:**
  LoRA for 1.6M parameters.
  Speaker codes for 1024 parameters (from layer 0 to 5).

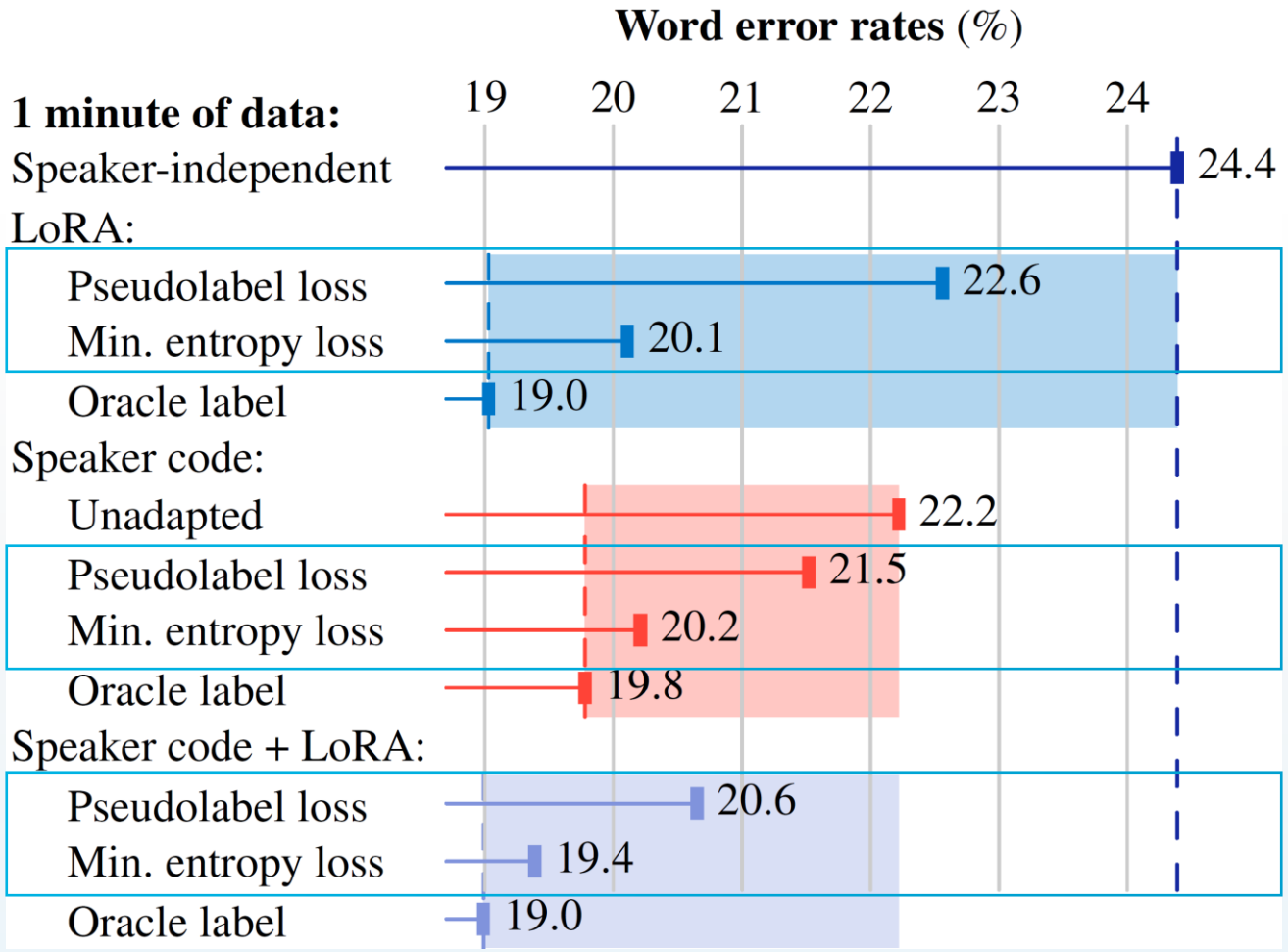
- **Speech recognition training:**
  Speaker codes dropout with 0.5 probability for optimal performance without adaptation.

- **Adaptation fine-tuning:**
  Done independently and per speaker (over 100 of them).

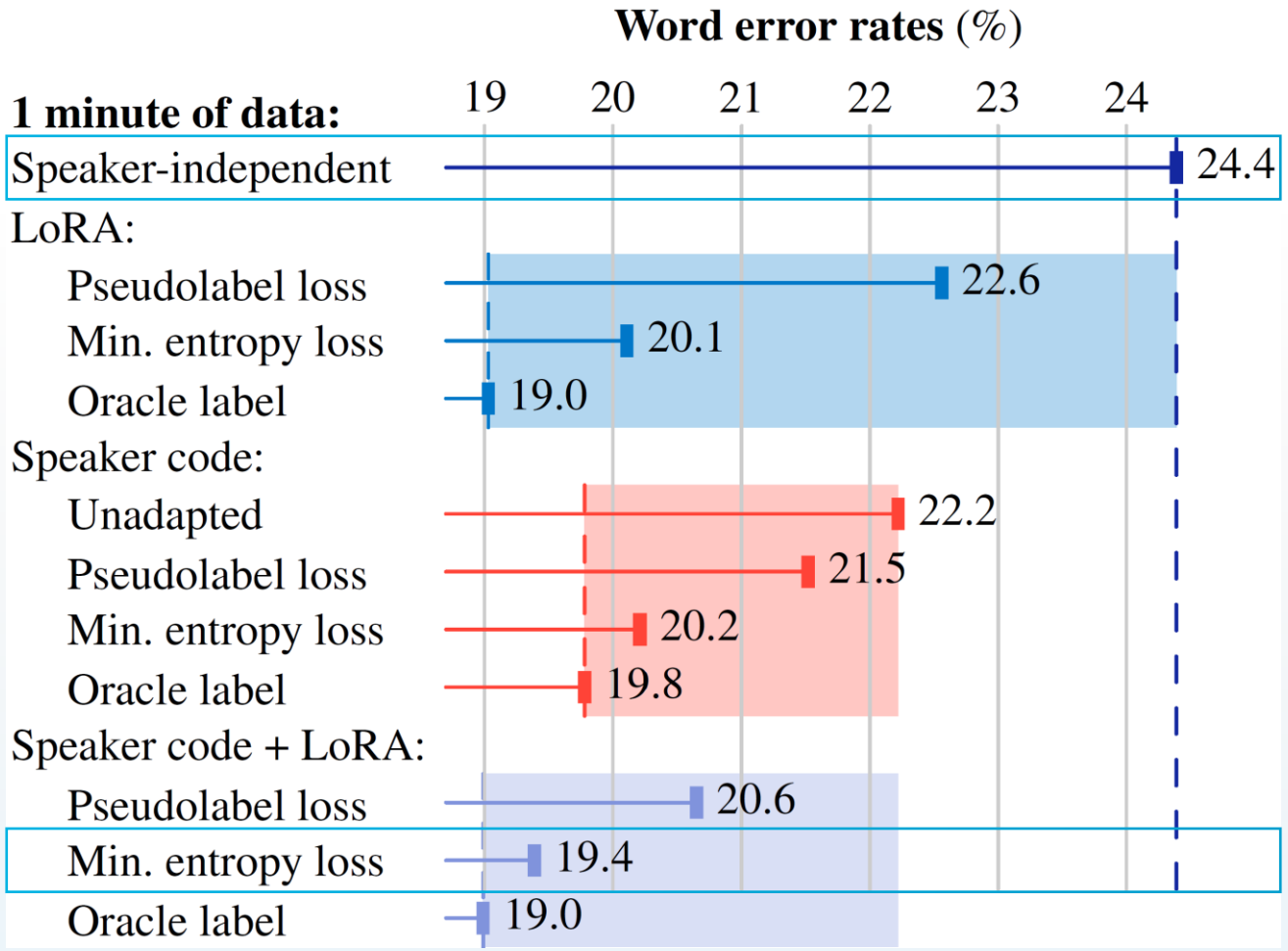

Word error rates (%)
19
20
21
22
23
24
1 minute of data:
Speaker-independent
24.4
LoRA:
Pseudolabel loss
22.6
Min. entropy loss
20.1
Oracle label
19.0
Speaker code:
Unadapted
22.2
Pseudolabel loss
21.5
Min. entropy loss
20.2
Oracle label
19.8
Speaker code + LoRA:
Pseudolabel loss
20.6
Min. entropy loss
19.4
Oracle label
19.0



The WER of the unadapted model is high. Our task is hard.

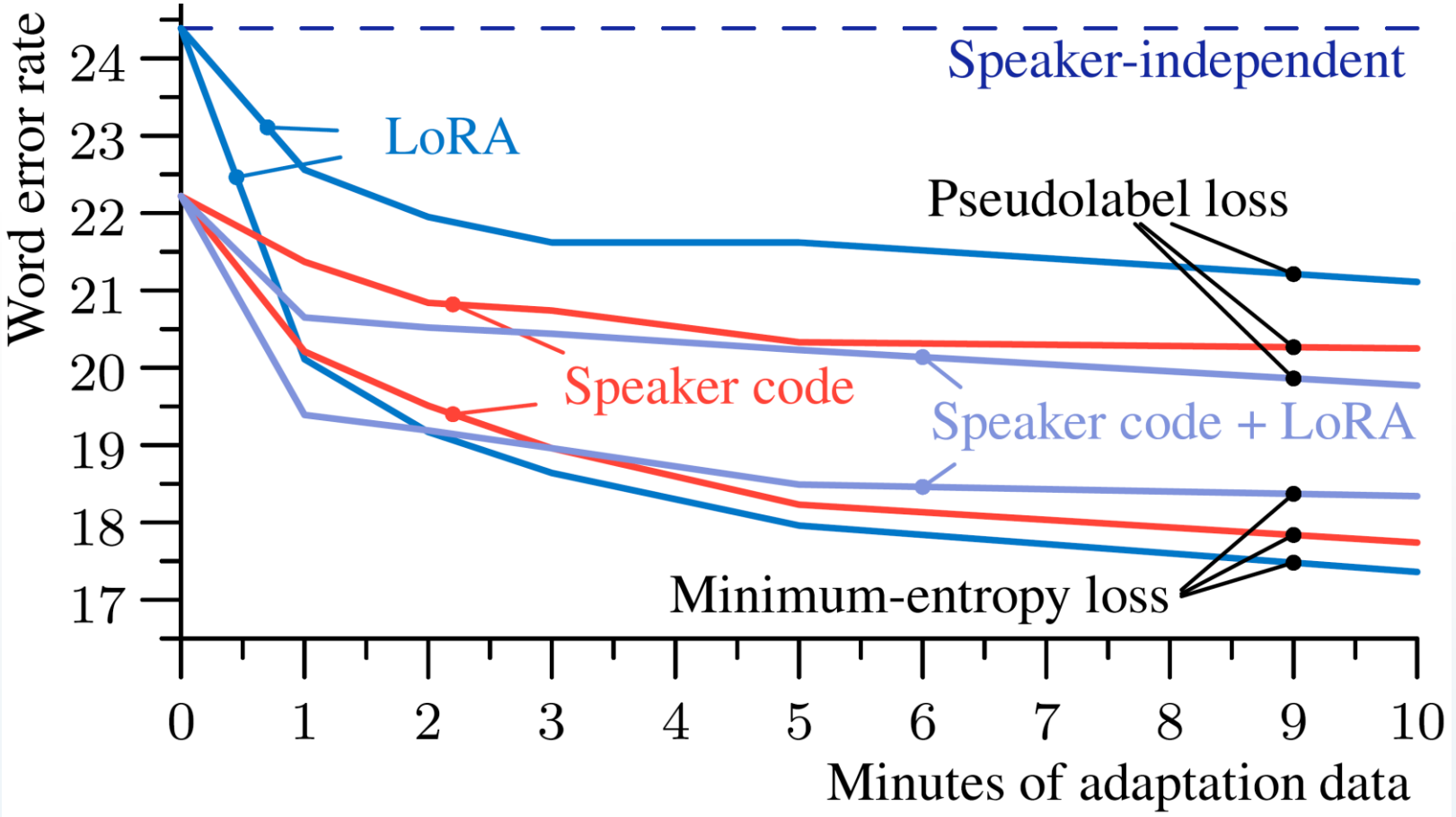


Minimum entropy always outperforms pseudolabel adaptation.



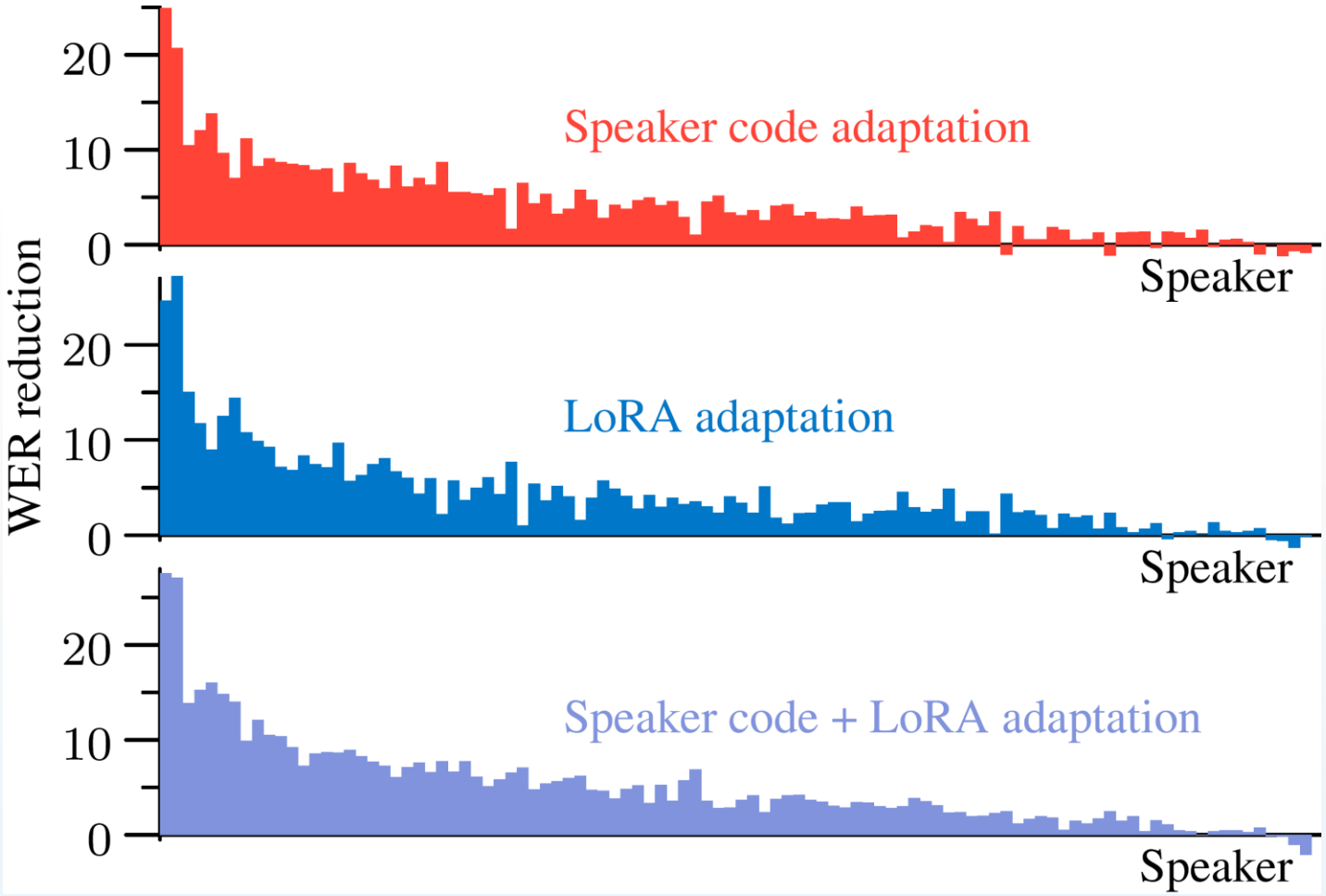
20% relative improvement of WER with 1 minute of unlabeled speech.

What if we have more available speech per speaker?



- With less data speaker codes an speaker codes + LoRA are much better.
- As the data amount increases, it is better to switch to just LoRA.
- Minimum entropy training always is superior to pseudolabel adaptation.

Are all the speakers getting better accuracies?



Most speakers are improving, with the highest initial WER benefitting the most.

Conclusion

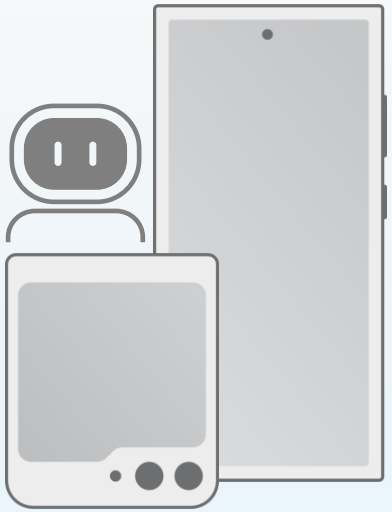
- If you don’t have labels, do not perform adaptation with pseudolabels.
  Use minimum entropy training.

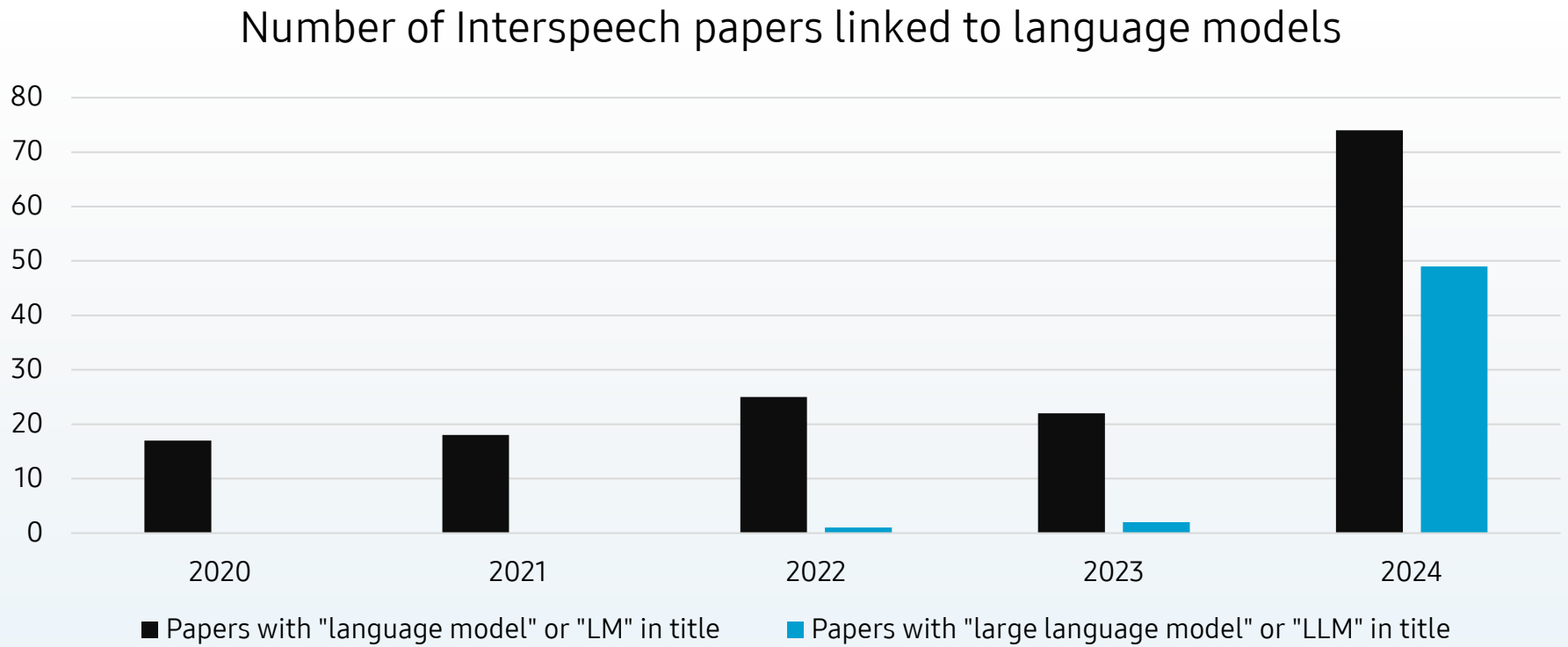
- If you have very few samples per speaker.
  Use a combination of speaker codes and LoRA.

Adaptation over 1 minute of unlabeled speech gave a 20% relative WER improvement.

Contents

Rising interest over Speech + LLMs





GPT-4O, Gemini, Moshi, LlaMA 4 …

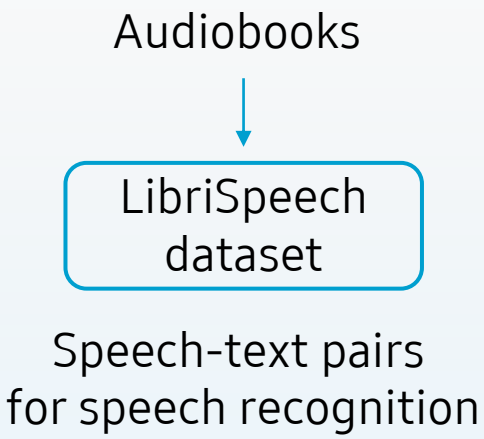
LLMs are trained on the test sets of speech recognition benchmarks.

LLMs are trained on the test sets of speech recognition benchmarks.

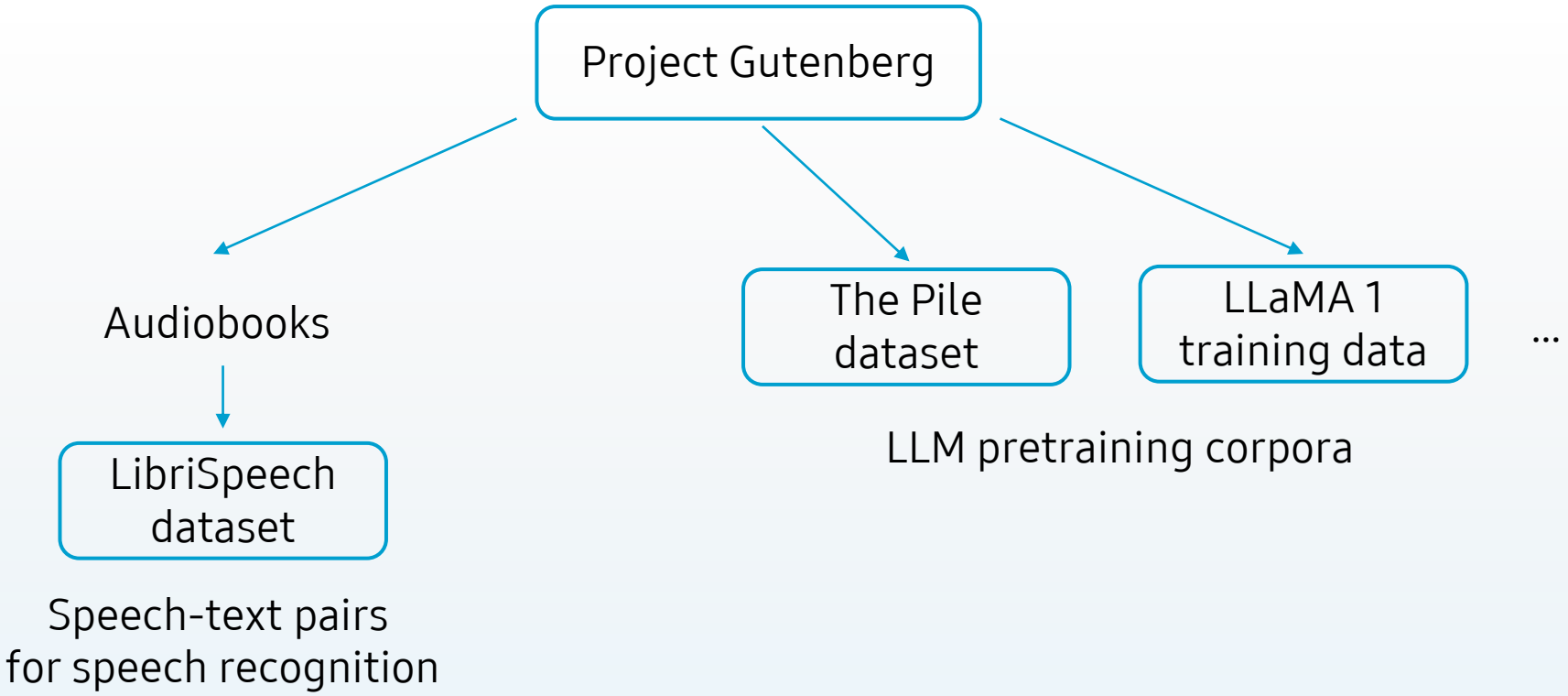
How bad is the contamination?

Is it actually problematic for speech recognition?

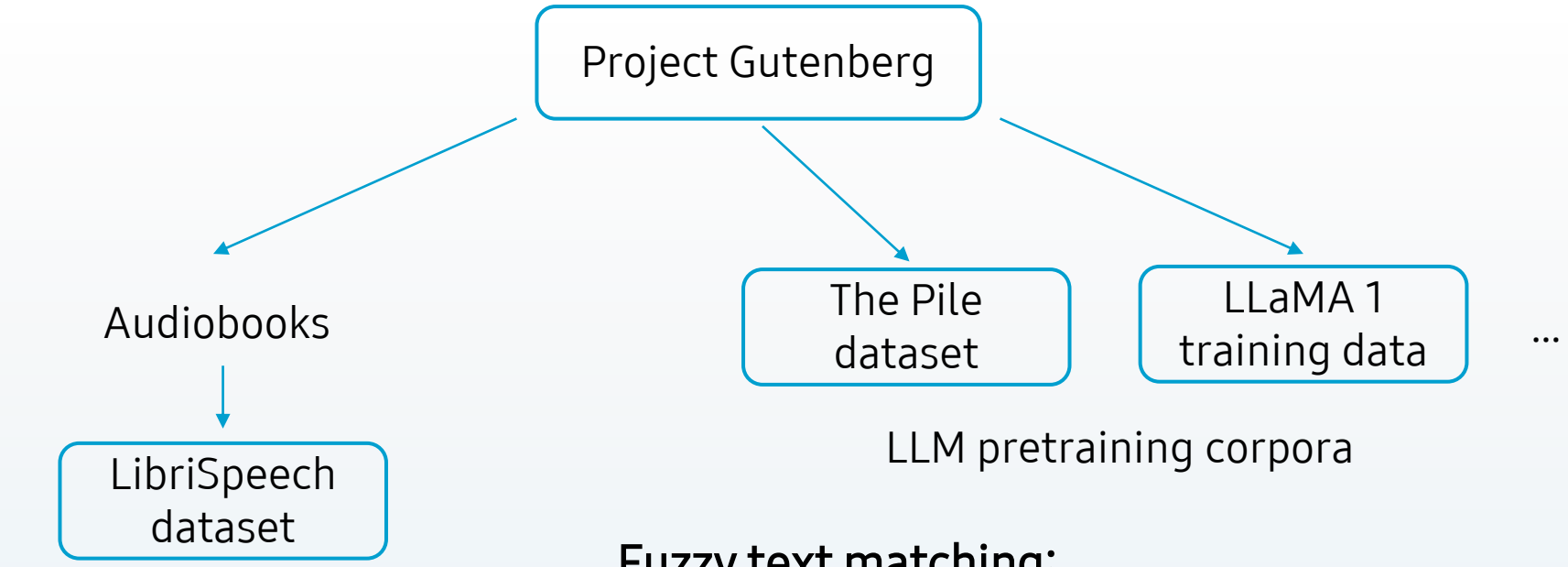
The case of LibriSpeech



The case of LibriSpeech



The case of LibriSpeech



Fuzzy text matching:

- **More than 60%** of LibriSpeech validation/testing sentences are in the Pile train set.
- 6848 out of 10848 sentences.

The case of CommonVoice

1/3 of CommonVoice validation and test sets are in the train set of the Pile.

How much are speech recognition evaluations affected?

Controlled contamination experiment

Train LLMs (>1B) from scratch on part of the Pile.

Uncontaminated	Trained on 30B tokens excluding LibriSpeech devs/tests.
Contaminated	Trained on 30B tokens including LibriSpeech devs/tests. Sentences used for contamination are termed « Leaked Sentences ».

How likely is each model to predict leaked and non-leaked sentences?

Word-level negative log-likelihoods comparison

Params.	Contaminated?	Non-leaked Sentences	Leaked Sentences
6.9 B	×		4.719
6.9 B	✓		4.682 (-0.036)

Lower is better.

The contaminated model is more likely to generate leaked test sentences.

Word-level negative log-likelihoods comparison

Params.	Contaminated?	Non-leaked Sentences	Leaked Sentences
6.9 B	×	4.729	4.719
6.9 B	✓	4.734 (+0.005)	4.682 (-0.036)

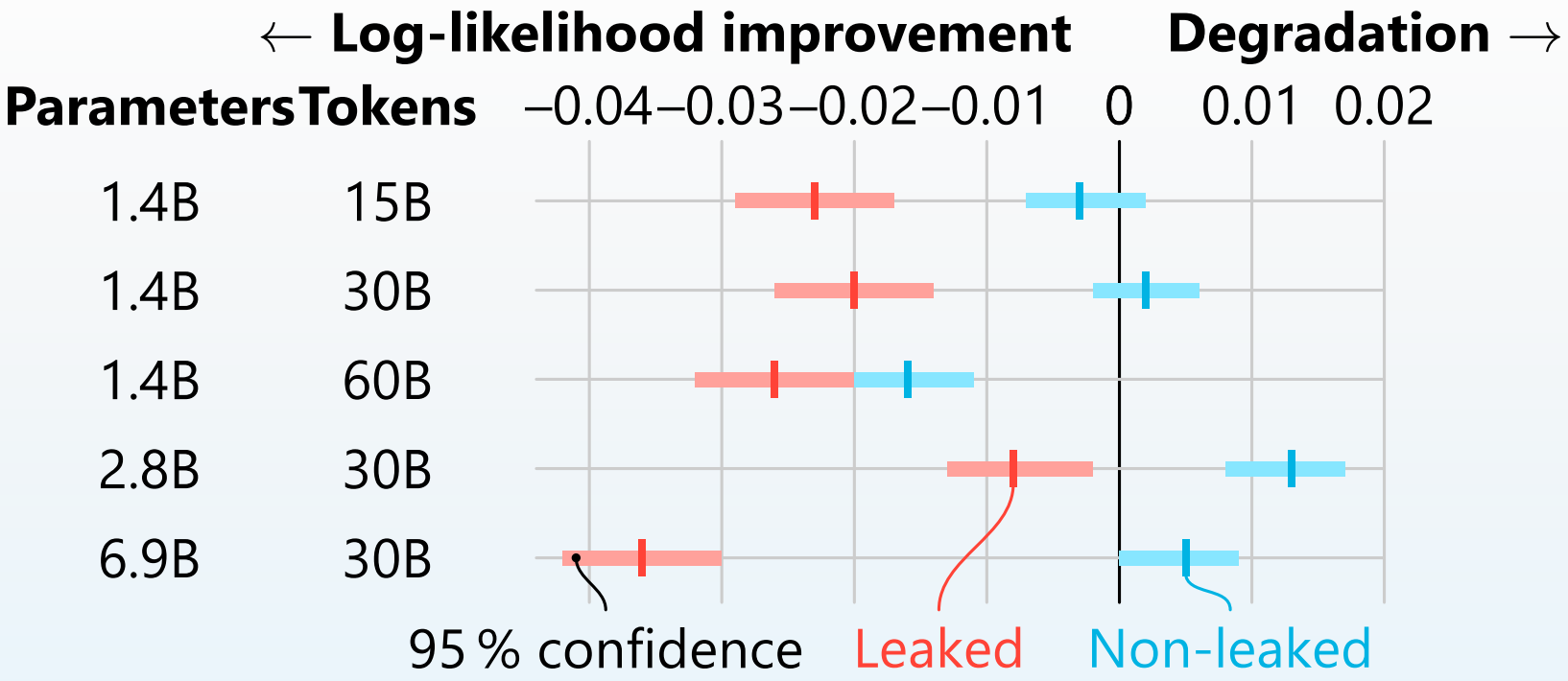
Lower is better.

The contaminated model is more like to generate leaked test sentences, but very similar at generating unseen sentences (non-leaked).

What about different model and training sizes?

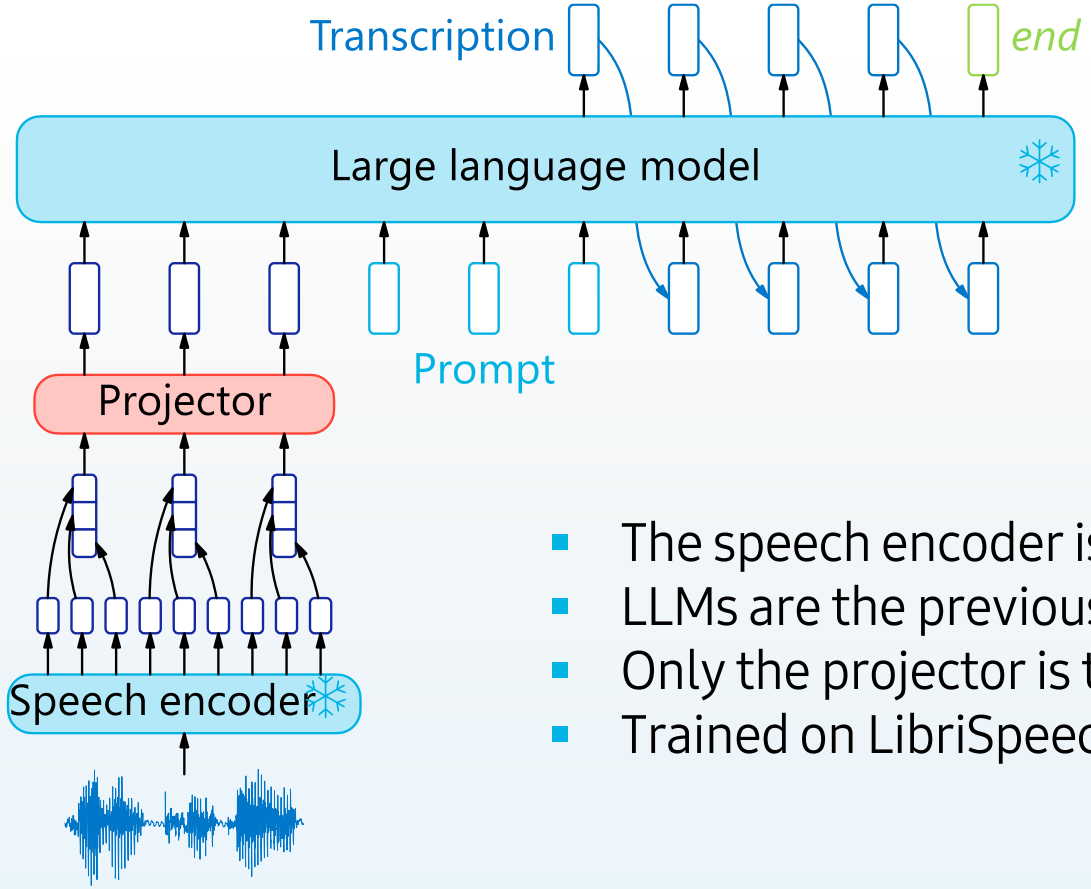
Probabilities are consistently different for contamined LLMs

Differences in log-likelihood between contaminated and uncontaminated LLMs.



Is it affecting speech recognition with SpeechLLMs?

SpeechLLMs for speech recognition



- The speech encoder is WavLM large.
- LLMs are the previous ones.
- Only the projector is trained.
- Trained on LibriSpeech 960 hours.

Log-Likelihoods comparison

Results on LibriSpeech test-clean

Params.	Contaminated?	Non-leaked Sentences	Leaked Sentences
6.9 B	×	0.1420	0.1496
6.9 B	✓	0.1407 (-0.0013)	0.1446 (-0.005)

Lower is better.

Statistically significant improvement as well for leaked sentences, when prompted with speech embeddings!

Log-Likelihoods comparison

Results on LibriSpeech test-clean (punctuated)

Params.	Contaminated?	Non-leaked Sentences	Leaked Sentences
6.9 B	×	0.4751	0.4767
6.9 B	✓	0.4816	0.4723 (-0.0044)

Lower is better.

It is worse with punctuated speech recognition.

Word Error Rates (WER) comparison

Results on LibriSpeech test-clean

Params.	Contaminated?	Non-leaked Sentences	Leaked Sentences
6.9 B	×	3.96%	3.94%
6.9 B	✓	3.61% (-0.35)	3.50% (-0.44)

Lower is better.

Differences are not significant when looking at word error rates.

Conclusions

- The evaluation protocol for speech recognition with SpeechLLMs is scientifically invalid due to significant test set contamination of LLMs.
- Contamined SpeechLLM only subtly differ in word error rates, but likelihoods are significant different.

Putting the impact aside, can we get back to not train our models on test sets?

Samsung Research

Thank you

Table 1: *Comparing log-likelihood, character error rate (CER), and word error rate (WER) of ASR systems using uncontaminated or contaminated LLMs pretrained on 30B tokens. Each result is averaged over 5 runs with different random seeds. Contaminated results that are significantly better are bolded. We observe that ASR systems become significantly more likely to generate leaked sentences when using contaminated 6.9B LLMs, but the differences in error rates can be subtle.*

Params.	Contaminated?	dev-clean		dev-other		test-clean		test-other	
		Non-leaked	Leaked	Non-leaked	Leaked	Non-leaked	Leaked	Non-leaked	Leaked
Punctuated ASR - Negative Log-Likelihood									
1.4B	×	0.5020	0.5041	0.6382	0.6481	0.5114	0.5063	0.6347	0.6630
	✓	0.5051	**0.5016**	0.6354	0.6511	0.5115	0.5031	0.6340	0.6624
6.9B	×	0.4694	0.4765	0.5994	0.6196	0.4751	0.4767	0.5900	0.6207
	✓	0.4745	**0.4717**	0.6038	**0.6066**	0.4816	**0.4723**	0.5945	**0.6172**
Punctuated ASR - CER (includes punctuation errors)									
6.9B	×	6.32	5.94	8.72	8.07	6.59	6.13	8.17	7.76
	✓	6.33	5.99	**8.15**	**7.67**	6.55	5.80	8.24	7.67
Regular ASR - Negative Log-Likelihood									
6.9B	×	0.1363	0.1587	0.2685	0.3048	0.1420	0.1496	0.2696	0.2978
	✓	0.1336	0.1581	0.2640	**0.2947**	0.1407	**0.1446**	0.2655	**0.2958**
Regular ASR - WER									
6.9B	×	3.28	3.59	7.39	6.51	3.96	3.94	6.88	6.85
	✓	**2.77**	3.77	7.01	**5.93**	3.61	3.50	6.89	6.97