Conversational AI

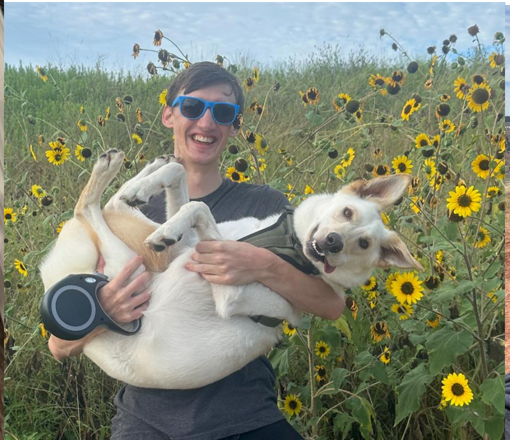
MARIA TELEKI

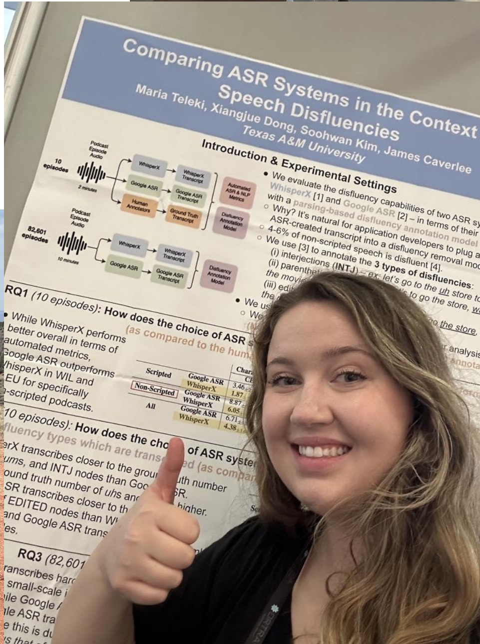
Will post slides after the talk!

Texas A&M University
Department of Computer Science & Engineering


LAVAZZA

about me




Comparing ASR Systems in the Context
Speech Disfluencies
Maria Teleki, Xiangjue Dong, Soohwan Kim, James Caverlee
Texas A&M University
Introduction & Experimental Settings

Agenda

1. Speech language models
2. Robustness of pipeline approaches
3. Speaker variation

First,

Let's review a recent survey of Speech Language Models.

The way we speak and write are different.

The way we speak and write are different.

So he calls me up, and he's like, 'I still love you,' and I'm like, I'm just, I mean, this is exhausting, you know – like we are never getting back together. Like, ever.

- TAYLOR SWIFT

The way we speak and write are different.

So he calls me up, and he's like, 'I still love you,' and I'm like, I'm just, I mean, this is exhausting, you know – like we are never getting back together. Like, ever.

- TAYLOR SWIFT

On The Landscape of Spoken Language Models: A Comprehensive Survey

Siddhant Arora[1*] Kai-Wei Chang[2*] Chung-Ming Chien[3*] Yifan Peng[1*] Haibin Wu[2*#]
Yossi Adi[4+] Emmanuel Dupoux[5+] Hung-Yi Lee[2+] Karen Livescu[3+] Shinji Watanabe[1+]
[1] Carnegie Mellon University, USA
[2] National Taiwan University, Taiwan
[3] Toyota Technological Institute at Chicago, USA
[4] Hebrew University of Jerusalem, Israel
[5] ENS - PSL, EHESS, CNRS, France

ArXiv, highlighted at INTERSPEECH '25 keynote

Big picture context is we live in a **MULTIMODAL** world

LLMs = *Large Language Models*	**SLMs =** *Speech Language Models*	**VLMs =** *Vision Language Models*	**RLMs =** *LLMs for Recommendation (I made this name up)*	…
p(text|text)	p(text|speech,text)	p(text|image,text)	p(text|collaborative-signal,text)	

And right now people are building models for all these different types of modalities

SLMs can do tasks that LLMs can't do

Task	**Examples of Instructions**
Speech recognition	Recognize the speech and give me the transcription. (Tang et al., 2024) Repeat after me in English. (Grattafiori et al., 2024)
Speech translation	Translate the following sentence into English. (Grattafiori et al., 2024) Recognize the speech, and translate it into English (Chu et al., 2023)
Speaker recognition	How many speakers did you hear in this audio? Who are they? (Tang et al., 2024)
Emotion recognition	Describe the emotion of the speaker. (Tang et al., 2024) Can you identify the emotion? Categorise into: sad, angry, neutral, happy (Das et al., 2024)
Question answering	What happened to this person? (Wang et al., 2023b) Generate a factual answer to preceding question (Das et al., 2024) What medicine is mentioned? Briefly introduce that medicine. (Peng et al., 2024b)

Table 2: Examples of instructions for speech-related tasks used in SLM instruction tuning.

SLMs are trained in a super cool new way!!!!!!!

Table 1: Typology of text and spoken LMs. We use a loose notation here, where *speech* and *text* are to be interpreted in context; for example, $p(text|text)$ in post-trained text LMs corresponds to modeling some desired ouptut text given an input text instruction or prompt. "Post-training" refers to any form of instruction-tuning and/or preference-based optimization of the SLM. Please see the sections below for details and references for the example models.

Type of LM	**Training Strategy**	**Model distribution**	**Examples**
pure text LM	pre-training	$p(text)$	GPT, Llama
pure text LM	post-training	$p(text\|text)$	ChatGPT, Llama-Instruct
pure speech LM	pre-training	$p(speech)$	GSLM, AudioLM, TWIST
pure speech LM	post-training	$p(speech\|speech)$	Align-SLM
speech+text LM	pre-training	$p(text, speech)$	SpiRit-LM, Moshi (pre-trained)
speech+text LM	post-training	$p(text, speech\|text, speech)$	Moshi (post-trained), Mini-Omni
speech-aware text LM	post-training	$p(text\|speech, text)$	SALMONN, Qwen-Audio-Chat

"tokenizing speech" → speech becomes like text for modeling

WE ARE HERE

Pipeline Approach vs. End-to-End Approach

ASR → **LLM** → **TTS**

SLM

✅ preferred by industry for *controllability* → think <u>custom vocabulary</u> & ease of <u>debugging</u>

✅ currently in research stage but very promising → major barrier imo is downsampling problem (will explain)

The big meta-level question to pay attention to…

- What’s the input?
- What’s the output?
- How do you glue the parts together?
- Based on the input and output, **what is the SIGNAL being learned (by the model)?**

Let’s talk SLM architecture

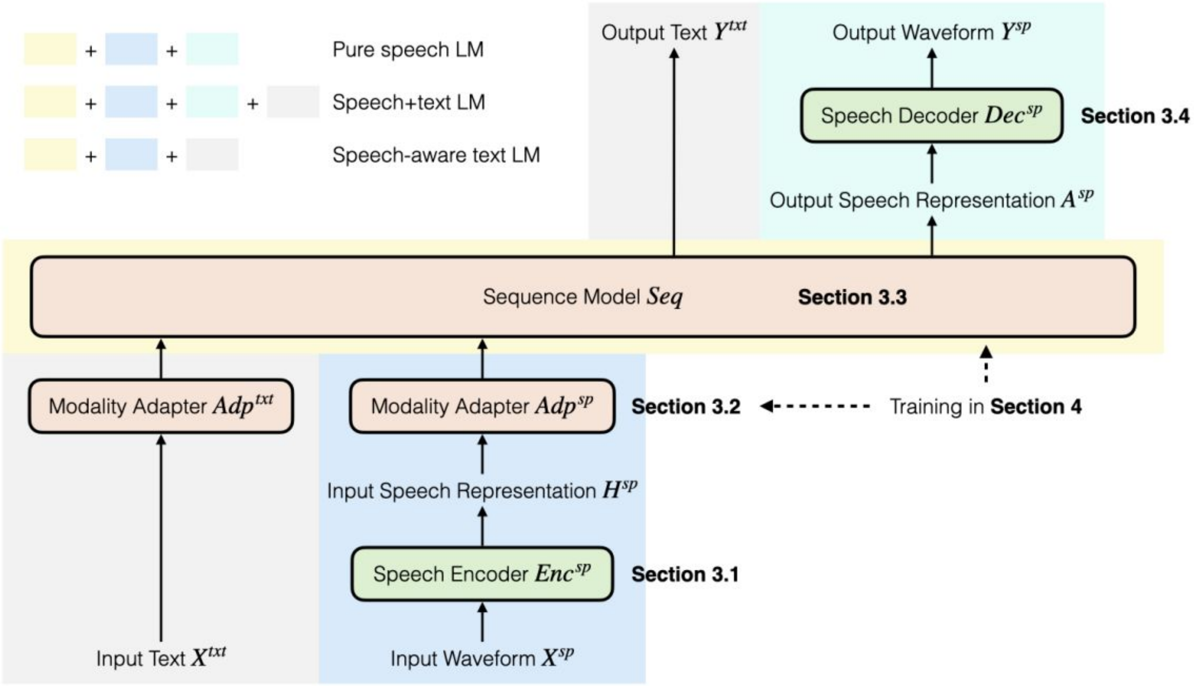


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

The paper is basically walking through this pic like it’s a map – so that’s what we’re going to do! :)

Let's talk SLM architecture

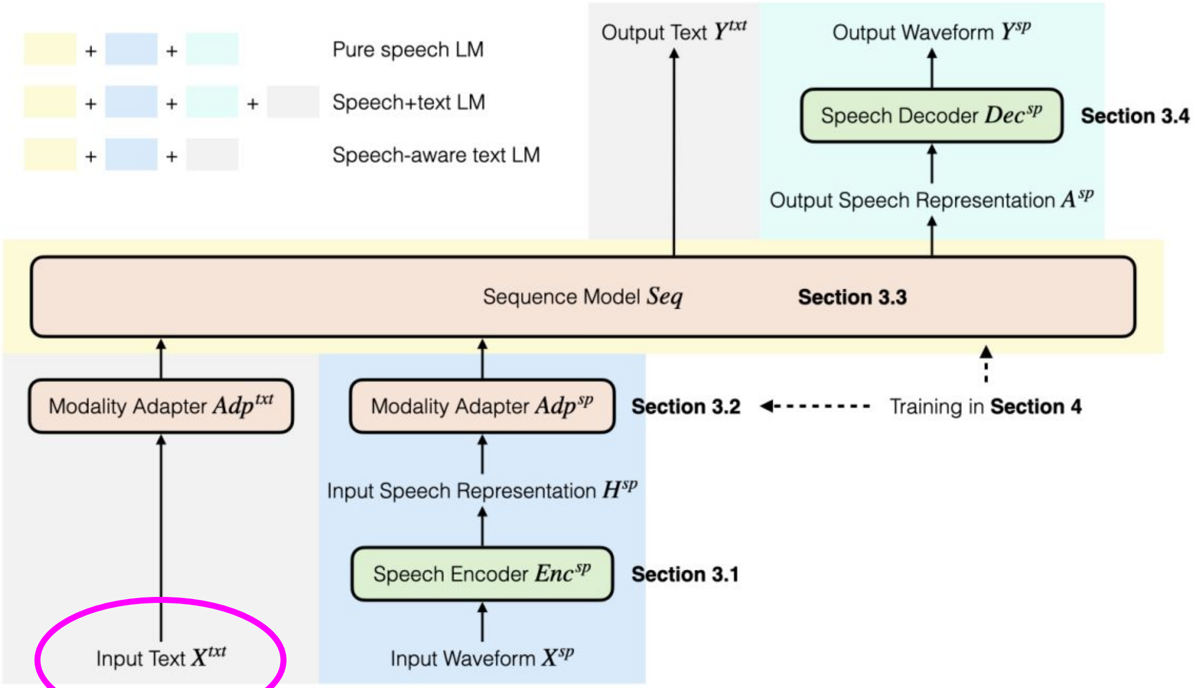


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

We know about input text, noice

Let's talk SLM architecture

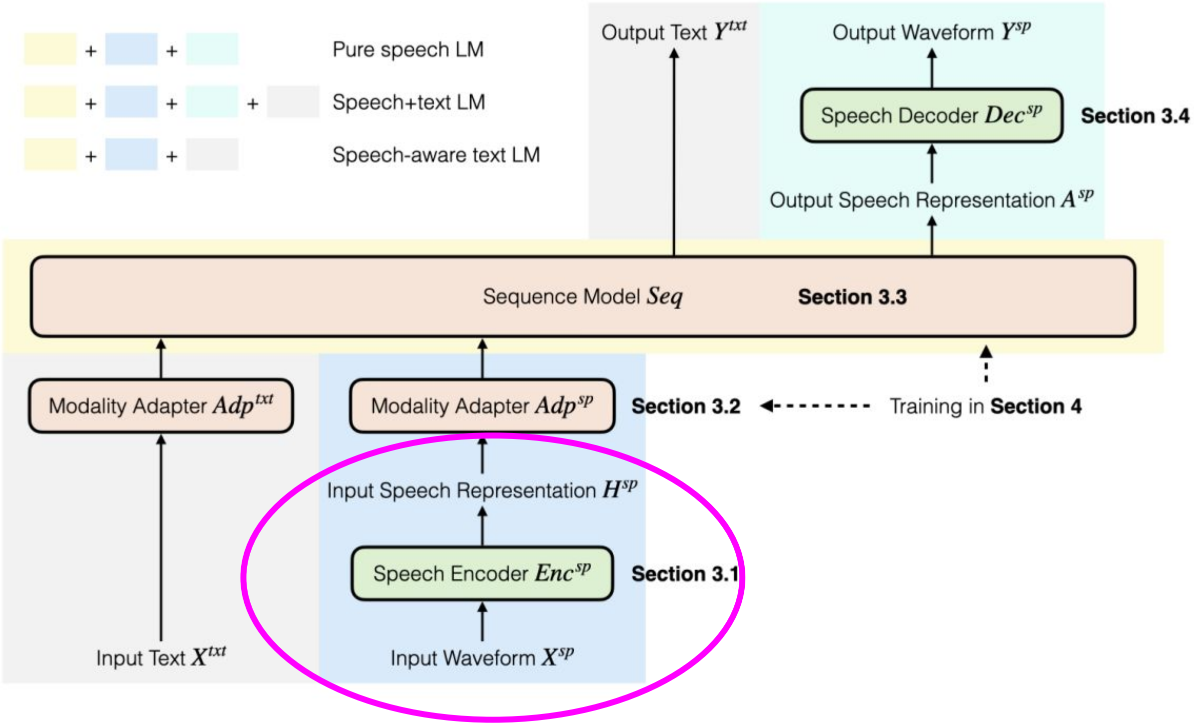


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

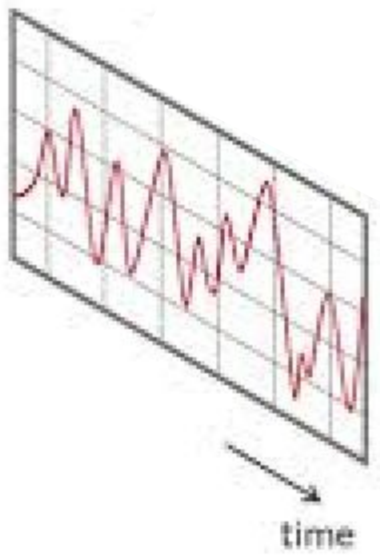
So now let's walk through the speech encoder!

Let’s start from the speech signal

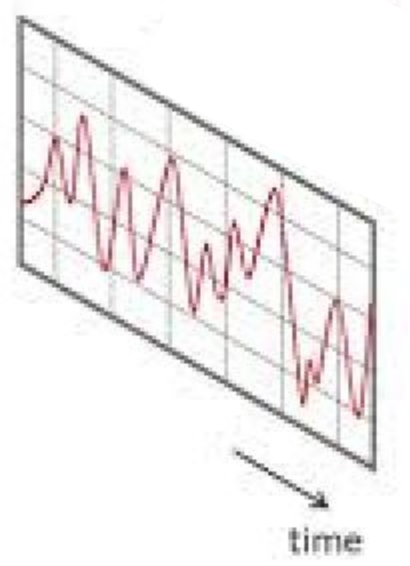
What is it? It’s air pressure over time.

Microphones pick up differences in air pressure – that’s sound.

So we get a lil plot of that.



Let’s start from the speech signal



What is it? It’s air pressure over time.

Microphones pick up differences in air pressure – that’s sound.

So we get a lil plot of that.

What’s the problem? Why can’t we just use that?

You have to super-duper oversample:

. Redundancy in Raw Waveform

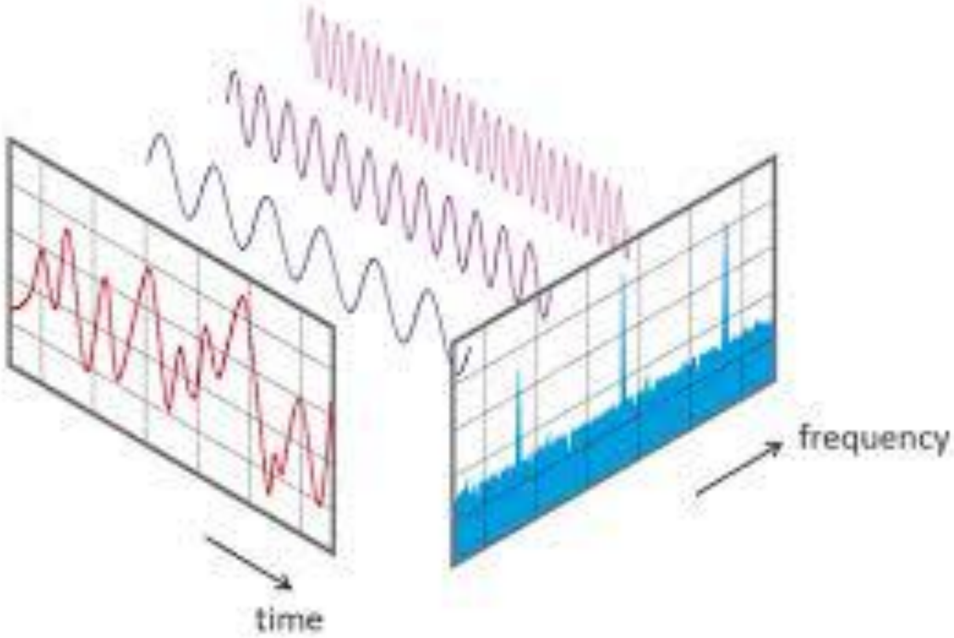
- **Oversampling relative to perception**:
  At 16 kHz, you get **16,000 samples per second**. But most speech information lives below ~4 kHz (Nyquist), and much of it is even lower (<2 kHz). → lots of extra samples.
- **Correlation across samples**:
  Neighboring values in the waveform are highly correlated (they don’t change randomly). For example, one period of a 200 Hz vowel spans ~80 samples — most of those points contain *the same information about the harmonic structure*.
- **Stationarity over frames**:
  Speech characteristics (formants, pitch) are stable for 20–50 ms. That’s **hundreds of samples** where the “meaning” doesn’t change.

o raw data is very long, repetitive, and inefficient to model directly.

But, we have a trick!

We can do a fourier transform –

Just pull out the frequency components for that time chunk!

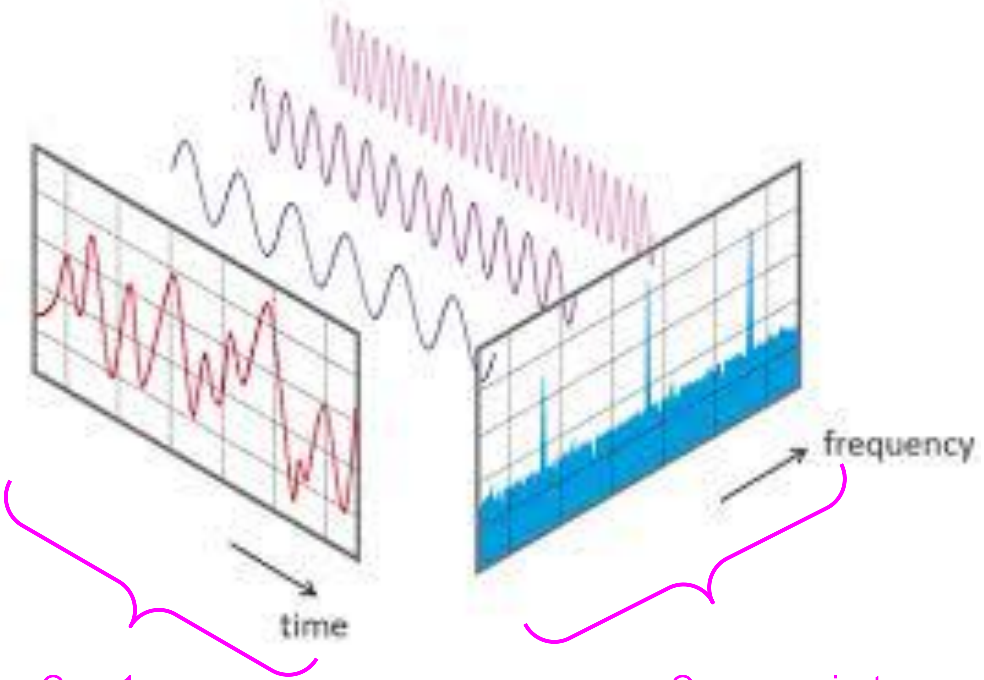


But, we have a trick!

We can do a fourier transform –

Just pull out the frequency components for that time chunk!

So now we can store 80 values per second *instead of 16,000 values per second*



Over 1 second, we're sampling 16,000 times to get this waveform

Or, we can just store 80 values for each frequency component

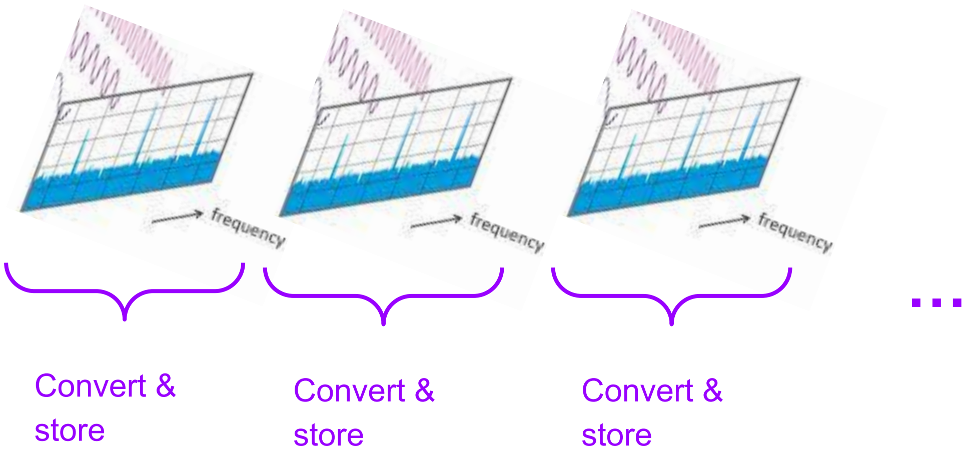
But, we have a trick!

We can do a fourier transform –

Just pull out the frequency components for that time chunk!

So now we can store 80 values per second *instead of 16,000 values per second*

So we store a “frequency snapshot” each second instead



But, we have a trick!

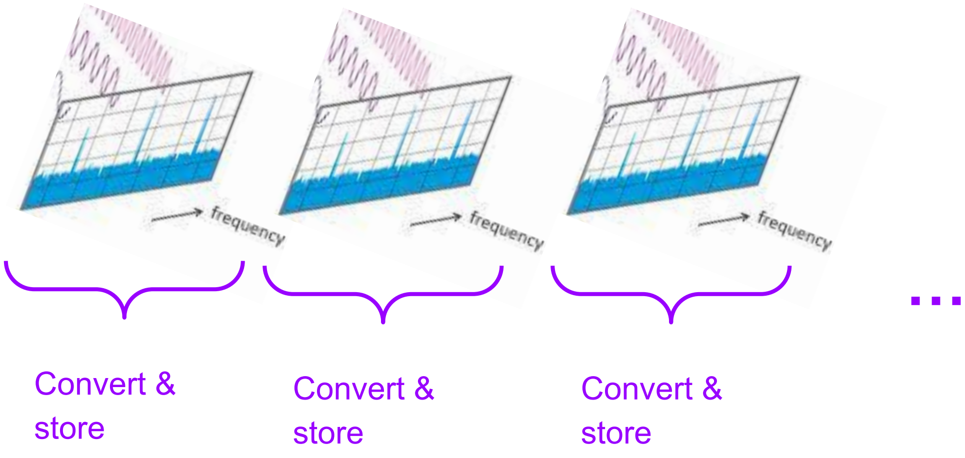
We can do a fourier transform –

Just pull out the frequency components for that time chunk!

So now we can store 80 values per second *instead of 16,000 values per second*

So we store a “frequency snapshot” each second instead

→ so we track how frequencies change *over time* while massively reducing data.

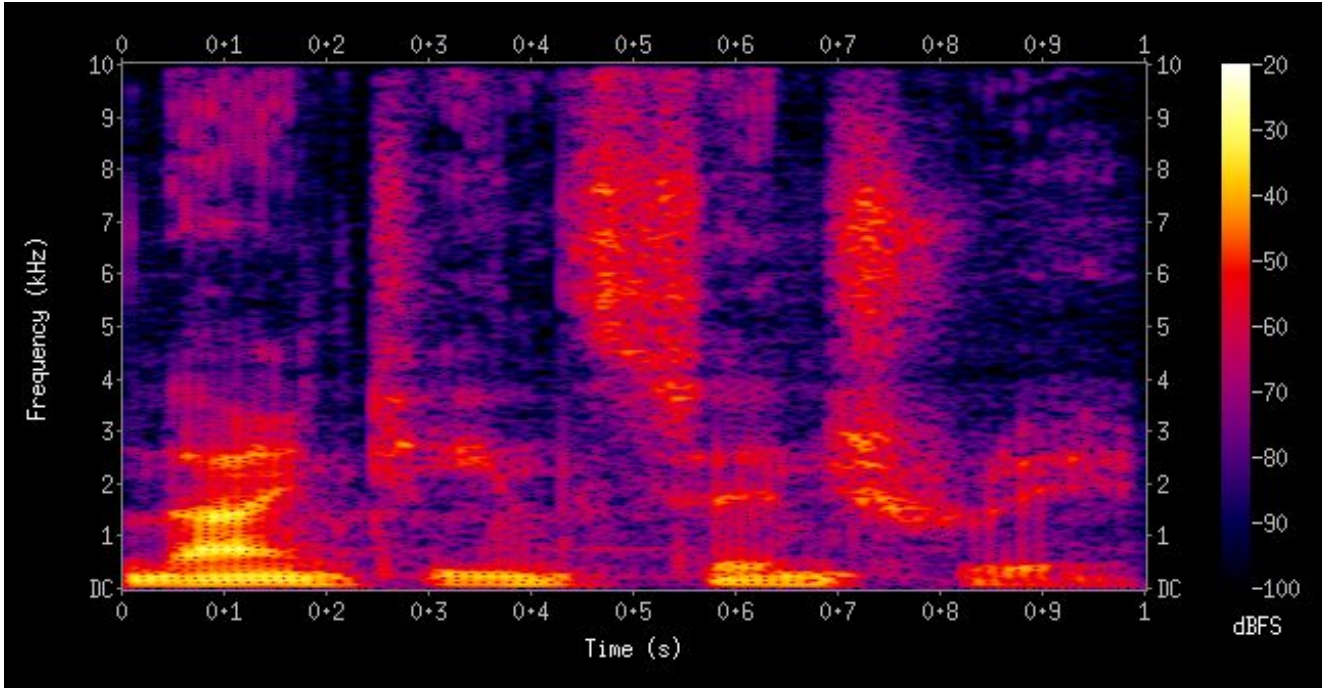


This technique is called the Short-Time Fourier Transform (STFT)!!!!!!!!

It solves our over-sampling problem w/out losing important information <3

We end up with a **spectrogram** that looks like this!

Short-Time Fourier Transform (STFT) Result



In our example, this is the equivalent of **overlaying a couple seconds of the FTs** – so we're seeing how the ***frequency components change over time***

Comparative Table of Function Approximation Methods

Aspect	Fourier Transform	Taylor Series	Neural Networks
Core Idea	Represent $f(x)$ as a sum of sinusoids: $f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}, \quad c_k = \frac{1}{2\pi}\int f(x)e^{-ikx}\,dx$	Approximate $f(x)$ near a point a: $f(x) \approx \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(x-a)^n$	Learn approximation from data using nonlinear units: $f(x) \approx \sum_{i=1}^{M} w_i\,\sigma(\langle v_i, x\rangle + b_i)$ (1 hidden-layer NN)
Basis Functions	Fixed global sines/cosines.	Fixed monomials $(x-a)^n$.	Adaptive nonlinear features via activation functions.
Learning vs. Direct Computation	Coefficients c_k computed directly from integral formulas (no learning).	Coefficients $\frac{f^{(n)}(a)}{n!}$ computed directly from derivatives (no learning).	Coefficients w_i, v_i, b_i **learned** from data via optimization (gradient descent).
Domain of Usefulness	Oscillatory/periodic signals, frequency analysis.	Smooth analytic functions, valid locally around expansion point.	Arbitrary nonlinear, high-dimensional, non-analytic functions.
Local vs. Global	Global — each term affects all x.	Local — accurate only near a.	Both — architecture-dependent (RBFs local, deep nets capture global).
Interpretability	Coefficients ↔ frequency content.	Coefficients ↔ derivatives at expansion point.	Parameters (weights) not directly interpretable.
Convergence	Converges in L^2 for square-integrable f; Gibbs phenomenon at discontinuities.	Converges within radius of convergence if f is analytic.	Universal Approximation Theorem: can approximate any continuous f on compact sets.
Computation	Fast Fourier Transform (FFT): $O(N \log N)$.	Easy if derivatives known; costly at high order.	Training costly (gradient descent), inference efficient.
Noise Sensitivity	Good for filtering (frequency cutoff).	Highly sensitive (derivatives amplify noise).	Can overfit to noise unless regularized.
Applications	Signal/image processing, PDEs, spectral analysis.	Physics models, perturbation methods, local expansions.	Pattern recognition, regression/classification, generative modeling.

Comparative Table of Function Approximation Methods

Aspect	Fourier Transform	Taylor Series	Neural Networks
Core Idea	Represent $f(x)$ as a sum of sinusoids: $f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx}, \quad c_k = \frac{1}{2\pi}\int f(x)e^{-ikx}\,dx$	Approximate $f(x)$ near a point a: $f(x) \approx \sum_{n=0}^{N} \frac{f^{(n)}(a)}{n!}(x-a)^n$	Learn approximation from data using no... units: $f(x) \approx$ $\sum_{i=1}^{M} w_i \sigma(\langle v_i, x\rangle +$ ...
Basis Functions	Fixed global sines/cosines.	Fixed monomials $(x-a)^n$.	Ada...
Learning vs. Direct Computation	Coefficients c_k computed directly from integral formulas (no learning).	Coefficients $\frac{f^{(n)}(a)}{n!}$ computed directly from derivatives (no learning).	
Domain of Usefulness	Oscillatory/periodic signals, frequency analysis	Smooth analytic functions, valid locally around expansion point.	
Local vs. Global		... accurate only near a.	Both — a... eep nets capture global).
Interpretab...		...nts ↔ derivatives at ...point.	Parameters (weights) not directly interpretable.
Converg...		...within radius of ...nce if f is analytic.	Universal Approximation Theorem: can approximate any continuous f on compact sets.
Computat...		Easy if derivatives known; costly at high order.	Training costly (gradient descent), inference efficient.
Noise Sensitivity	Goo...	Highly sensitive (derivatives amplify noise).	Can overfit to noise unless regularized.
Applications	Signal/image processing, PDEs, spectral analysis.	Physics models, perturbation methods, local expansions.	Pattern recognition, regression/classification, generative modeling.

There are lots of signals where we don't really KNOW how they work… so this "math tool" is the best tool we can use

Speech moves through the air in waves of air pressure… so this "math tool" pretty realistically models how speech works & is cheap to use & solves our downsampling problem

There are actually lots of ways to encode speech features –

3.1.1 Continuous Features

To extract informative representations from raw waveforms, a speech representation model—either a learned encoder or a digital signal processing (DSP) feature extractor—converts speech into continuous features. These continuous features may include:

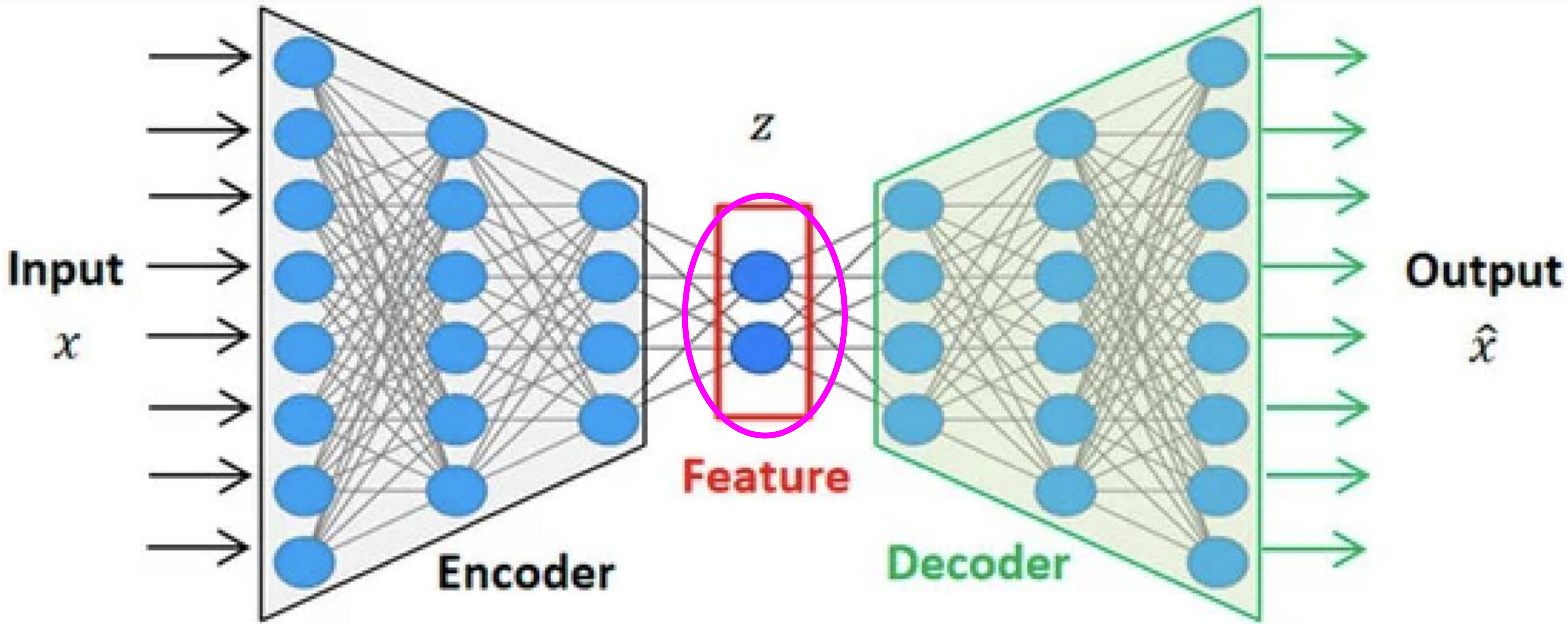
✅ 1. Traditional spectrogram features, such as mel filter bank features (Huang et al., 2001).

2. Hidden representations from self-supervised learning-based (SSL) speech encoders, such as wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), or WavLM (Chen et al., 2022).

3. Hidden representations from supervised pre-trained models, such as Whisper (Radford et al., 2023) or USM (Zhang et al., 2023b).

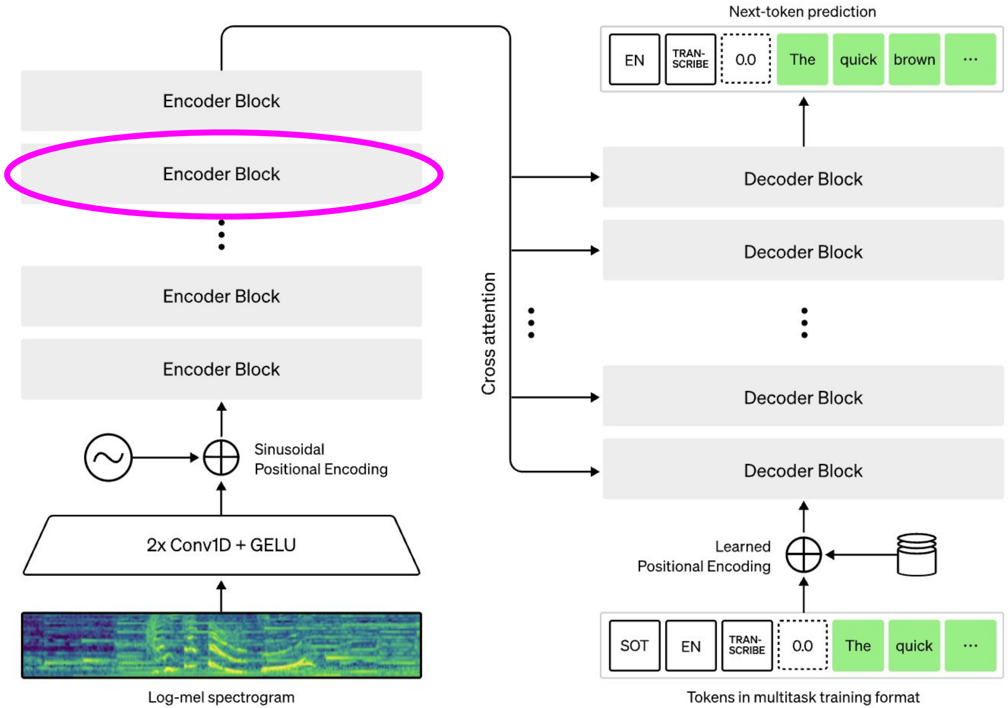
4. Hidden representations from neural audio codec models, such as SoundStream (Zeghidour et al., 2022) or EnCodec (Défossez et al., 2023).

2. Hidden representations from self-supervised learning-based (SSL) speech encoders, such as wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), or WavLM (Chen et al., 2022).



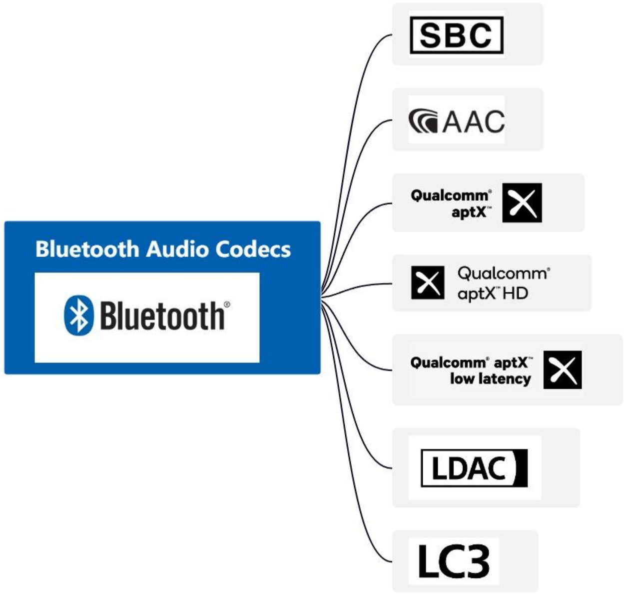
Take it from an autoencoder

3. Hidden representations from supervised pre-trained models, such as Whisper (Radford et al., 2023) or USM (Zhang et al., 2023b).



Take it from a Whisper layer

4. Hidden representations from neural audio codec models, such as SoundStream (Zeghidour et al., 2022) or EnCodec (Défossez et al., 2023).



Codec = compressed, lossLESS audio representation

(big topic rn)

Encoder

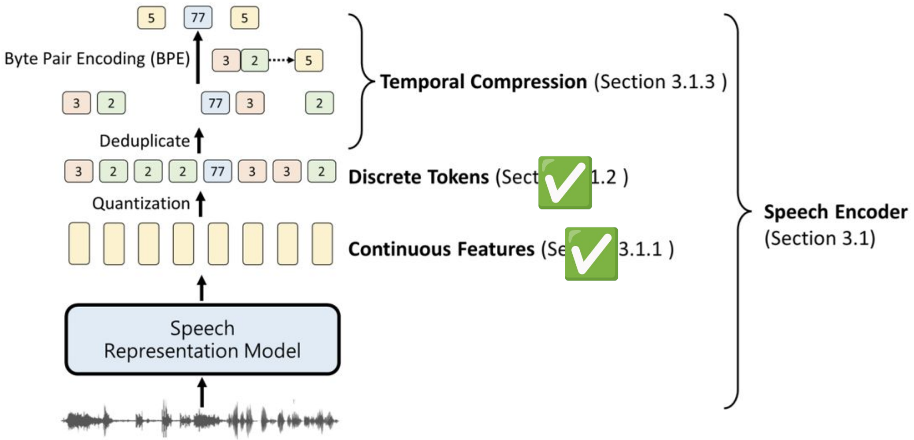


Figure 2: A general pipeline for speech encoders. Note that different encoders use different components of the pipeline. See Section 3.1 for more details.

Now we know how to go from speech signal -> tokens

(more deets in the paper)

But we're not done – there's ANOTHER over-sampling/compression issue…

Encoder

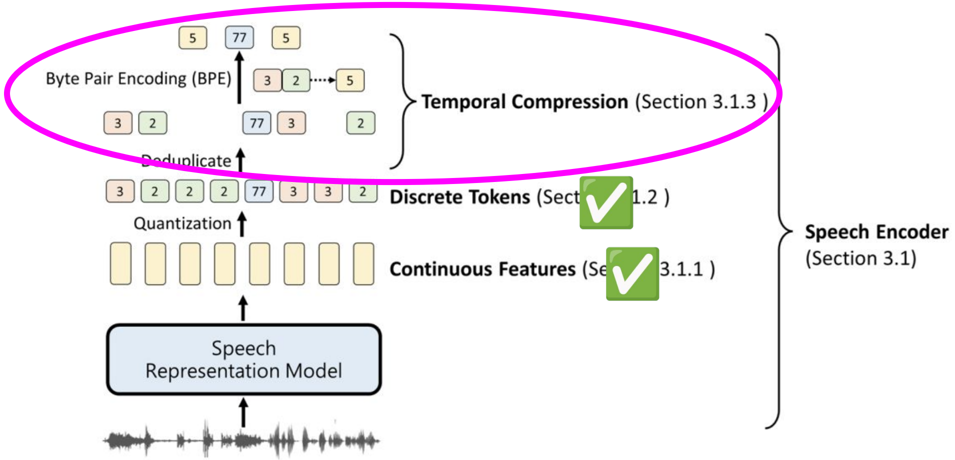


Figure 2: A general pipeline for speech encoders. Note that different encoders use different components of the pipeline. See Section 3.1 for more details.

So there are lots of tools to apply here to downsample AGAIN:

- BPE
- Multi-stream capture
- Duration prediction

Big theme: downsampling is a big deal in audio/speech world!!!!!!!!

3 Main Approaches to TRAIN 🚂 SLMs

Train the (big) sequence model (aka the LLM) *[most expensive $$$]*

```
Speech -> ❄️Speech Encoder ---------------------------\
                                                       -> 🔥Sequence Model (fine-tuned jointly)
Text   -> ❄️Token Embedding --------------------------/
```

Train the modality adapters *[least expensive $]*

```
Speech -> ❄️Speech Encoder -> 🔥Modality Adapter -> ❄️Sequence Model
Text   -> ❄️Token Embedding -> 🔥Modality Adapter -> ❄️Sequence Model
```

Train the speech encoder *[mid-expensive $$]* *[PS this is kinda like just training 1 modality adapter]*

```
Speech -> 🔥Speech Encoder -> (aligned directly to LM token space) -> ❄️Sequence Model
Text   -> ❄️Token Embedding ----------------------------------> ❄️Sequence Model
```

Let's talk SLM architecture

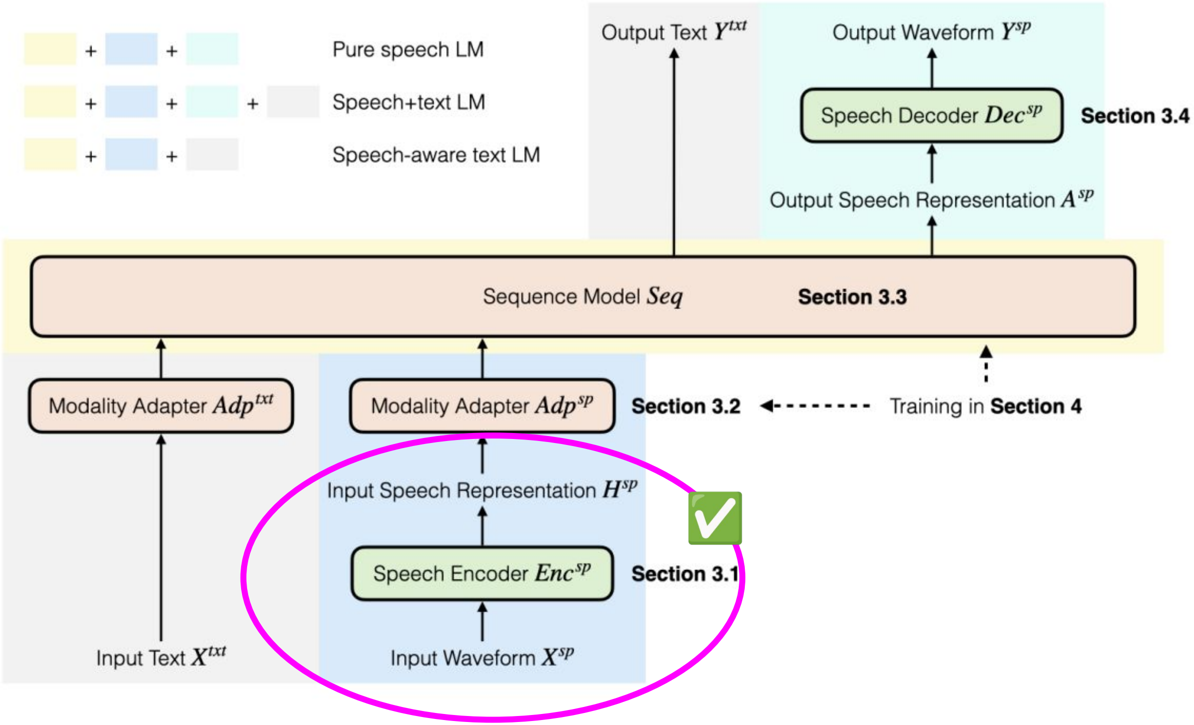


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

Ok yay so now we know about the speech encoder

Let’s talk SLM architecture

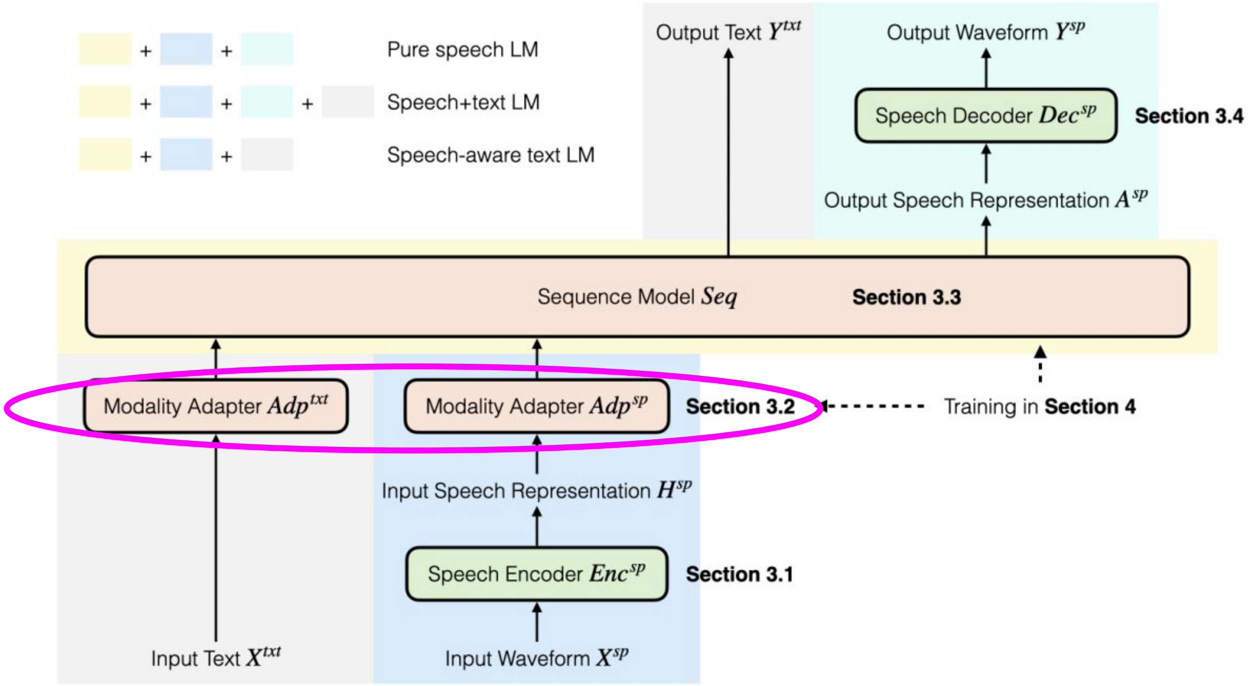


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

Let’s talk about the **modality adapters**

Modality Adapters

Train the modality adapters ***[least expensive $]***

```
Speech -> Speech Encoder -> Modality Adapter -> Sequence Model
Text   -> Token Embedding -> Modality Adapter -> Sequence Model
```

In many SLMs (especially speech-aware text LMs), the speech encoder (Section 3.1) and the sequence model (Section 3.3) are initially developed separately and then combined. It is therefore necessary to somehow align the output of the speech encoder with the expectations of the sequence model, and this is the role of the modality adapter. The modality adapter is typically trained on downstream tasks or, in the case of speech+text LMs, as part of pre-training (see Section 4 for more details on training).

The whole point is to make the **SPEECH TOKENS and TEXT TOKENS match up correctly!!!!**

Lots of ways to do a Modality Adapter… Broad Overview

Comparative Table of Adapter Architectures in Multimodal Models

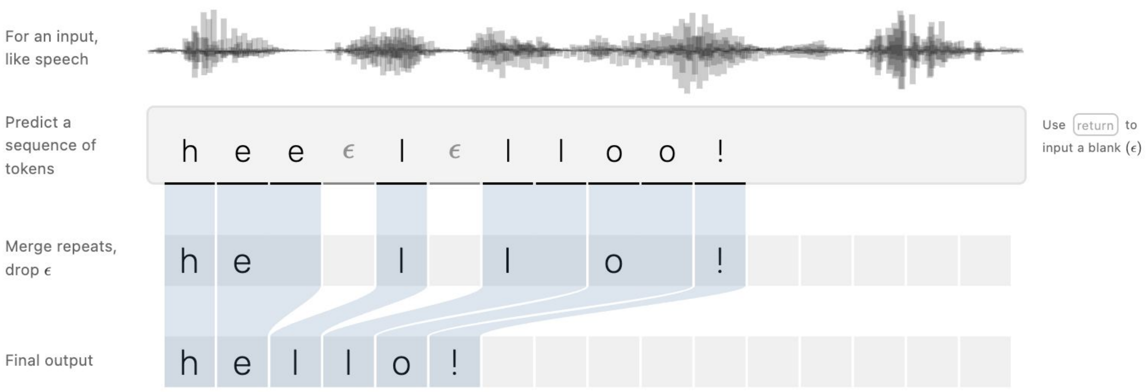
Adapter Architecture	Speech–Language	Vision–Language	Recommender–Language
Linear / MLP Projection	Project continuous speech embeddings (e.g., wav2vec2, HuBERT) into LLM token space.	Project vision encoder outputs (ViT/CNN/CLIP) into the same dimensionality as LLM embeddings.	Map item embeddings or categorical features into LLM-compatible vectors.
Quantization (Discrete Tokens)	VQ-VAE, k-means clustering to produce "pseudo-text" tokens from speech, directly fed into LLM.	Less common (though discrete VQ-image tokens exist, e.g., VQGAN), but most models use continuous projection instead.	Rare — item IDs can act as discrete tokens, sometimes directly embedded as vocab.
Cross-Attention Adapters	LLM attends to speech embeddings via cross-attention blocks (e.g., spoken QA grounding).	Popular: Flamingo inserts cross-attention layers where LLM queries image embeddings.	LLM attends over user history/item embeddings via cross-attention, aligning behavioral context with language tokens.
Prefix / Prompt Tuning	Encode speech embeddings into a sequence of pseudo-tokens prepended as "soft prompts."	Encode image embeddings as prefix tokens before text input (BLIP-2 Q-Former, Kosmos-1).	Encode user/item history as prompt tokens describing context (P5, TALM).
Bottleneck Adapters	Small trainable modules (e.g., LSTM, Conformer bottlenecks) compress variable-length speech into manageable embeddings.	Smaller adapter layers inserted in the vision encoder or between encoder–decoder.	Trainable bottlenecks align high-dim item features with LLM hidden states.
Fusion / Multi-Modal Attention	Speech features and LLM hidden states jointly processed via fusion transformer blocks.	Multimodal transformer fusion (e.g., CLIP-LMs, PaLM-E) combining vision & text tokens.	Fusion of user/item embeddings with natural language tokens for personalized reasoning.

The whole point is to make the **[SPEECH/VISION/REC] TOKENS and LLM TOKENS match up correctly!!!!**

Lots of ways to do a Modality Adapter…

Connectionist Temporal Classification (CTC)-based compression: This method compresses H^{sp} (Eq. 1) according to the posterior distribution from a CTC-based speech recognizer (Gaido et al., 2021). CTC (Graves et al., 2006), a commonly used approach for ASR, assigns each time step a probability distribution over a set of label tokens, including a blank ("none of the above") symbol. The time steps with high non-blank probabilities indicate segments that are likely to carry important linguistic information. CTC compression aggregates the frame-level labels, specifically by merging repeated non-blank labels and removing blanks. This approach produces a compressed representation intended to retain the relevant content of the original sequence while significantly reducing its length (Wu et al., 2023b; Tsunoo et al., 2024).

How CTC collapsing works



How do you make speech tokens & text tokens match up correctly? Compression!

Lots of ways to do a Modality Adapter…

Q-Former: The Q-Former (Li et al., 2023) is an adapter that produces a fixed-length representation by encoding a speech representation sequence of arbitrary length into M embedding vectors, where M is a hyperparameter (Lu et al., 2024b).

Let the input speech representation sequence be:

$$X = \{x_1, x_2, \ldots, x_{L'}\}, \quad x_i \in \mathbb{R}^{d'}, \tag{3}$$

where L' is the sequence length and d' is the dimension of the embeddings.

To achieve a fixed-length representation, Q-Former introduces M trainable query embeddings:

$$Q = \{q_1, q_2, \ldots, q_M\}, \quad q_i \in \mathbb{R}^{d'}. \tag{4}$$

These queries interact with X via a cross-attention mechanism:

$$\mathrm{Attn}(Q, X) = \mathrm{softmax}\left(\frac{QW_Q(XW_K)^T}{\sqrt{d'}}\right) XW_V, \tag{5}$$

where W_Q, W_K, and W_V are learnable projection matrices. The result is a sequence of M embeddings.

In some approaches, instead of directly encoding the entire utterance into M vectors, a *window-level Q-Former* is applied (Yu et al., 2024; Pan et al., 2024; Tang et al., 2024) to retain temporal information. In the window-level Q-Former, the input embedding sequence is segmented, and the Q-Former is applied to each segment.

Lu et al. (2024a) compare the Q-Former with CNN-based modality adapters in a speech-aware text LM, finding that the Q-Former produces better performance on the Dynamic-SUPERB benchmark (Huang et al., 2024) (see Section 7 for more on this and other SLM benchmarks).

How do you make speech tokens & text tokens match up correctly? Compression!

This approach plugs more directly into the LLM (in the attention mechanism)

Train the speech encoder [mid-expensive $$] [PS this is kinda like just training 1 modality adapter]

```
Speech -> Speech Encoder -> (aligned directly to LM token space) -> Sequence Model
Text   -> Token Embedding ---------------------------------> Sequence Model
```

Implicit alignment Speech and text modalities can be implicitly aligned through techniques such as the "modal-invariance trick" (Fathullah et al., 2024) or behavior alignment (Wang et al., 2023a). The idea is that the model should produce identical responses regardless of the input modality, provided the input conveys the same meaning. This approach often utilizes ASR datasets. The text transcript is input to a text LLM to generate a text response, while the corresponding speech recording is input into the SLM, which is trained to generate the same text response. Another idea found to be useful for implicit alignment is training spoken LLMs for audio captioning, where a spoken LLM takes audio as input and outputs its description. It has been observed that training a spoken LLM solely through audio captioning can generalize to tasks it has never seen during training (Lu et al., 2024a;b).

Explicit alignment Speech and text modalities can also be explicitly aligned by matching speech features to corresponding text embeddings, via optimization of appropriate distance/similarity measures. For example, Wav2Prompt (Deng et al., 2024) and DiVA (Held et al., 2024) align modalities by minimizing the L_2 distance between speech features and the token embeddings of their transcripts in a text LLM while keeping the text embeddings fixed.

Train the (big) sequence model (aka the LLM) *[most expensive $$$]*

```
Speech -> Speech Encoder -------------------------\
                                                   -> Sequence Model (fine-tuned jointly)
Text   -> Token Embedding -------------------------/
```

You just train the whole thing end-to-end, it’s expensive and there’s not really any tricks

→ Another great reference!

Summarizing Speech: A Comprehensive Survey

Fabian Retkowski[1] **Maike Züfle**[1] **Andreas Sudmann**[2] **Dinah Pfau**[3]
Shinji Watanabe[4] **Jan Niehues**[1] **Alexander Waibel**[1,4]
[1]KIT [2]University Bonn [3]Deutsches Museum [4]CMU
{fabian.retkowski,maike.zuefle,jan.niehues,alex.waibel}@kit.edu
asudmann@uni-bonn.de d.pfau@deutsches-museum.de swatanab@andrew.cmu.edu

EMNLP '25

→ Another great reference!

Reference	Audio Encoder	Projector	LLM
Fathullah et al. (2024)	🔥 Conformer (Gulati et al., 2020)	🔥 Linear	❄ LLaMA-2-7B-chat (Touvron et al., 2023)
Shang et al. (2024)	🔥 Conformer (Gulati et al., 2020)	🔥 Q-Former (Li et al., 2023)	≈ LLaMA-2-7B-chat (Touvron et al., 2023)
Microsoft et al. (2025)	🔥 Conformer (Gulati et al., 2020)	🔥 MLP	❄ Phi-4-mini-instruct (Microsoft et al., 2025)
Kang and Roy (2024)	🔥 HuBERT-Large (Hsu et al., 2021)	🔥 Linear	❄ MiniChat-3B (Zhang et al., 2024a)
Züfle et al. (2025)	❄ HuBERT-Large (Hsu et al., 2021)	🔥 Q-Former (Li et al., 2023)	❄ LLaMA3.1-8B-Instruct (Grattafiori et al., 2024)
He et al. (2025)	❄ MERaLiON-Whisper (He et al., 2025)	🔥 MLP	≈ SEA-LION V3 (He et al., 2025)
Chu et al. (2024)	🔥 Whisper-large-v3 (Radford et al., 2023)	🔥 Linear	🔥 Qwen-7B (Bai et al., 2023)
Eom et al. (2025)	❄ Whisper-large-v2 (Radford et al., 2023)	🔥 Q-Mamba (Eom et al., 2025)	🔥 Mamba-2.8B-Zephyr (xiuyul/mamba-2.8b-zephyr)

Table 2: Overview of Audio Encoder → Projector → LLM Architectures (🔥 trainable, ❄ frozen, ≈ LoRA)

Let's talk SLM architecture

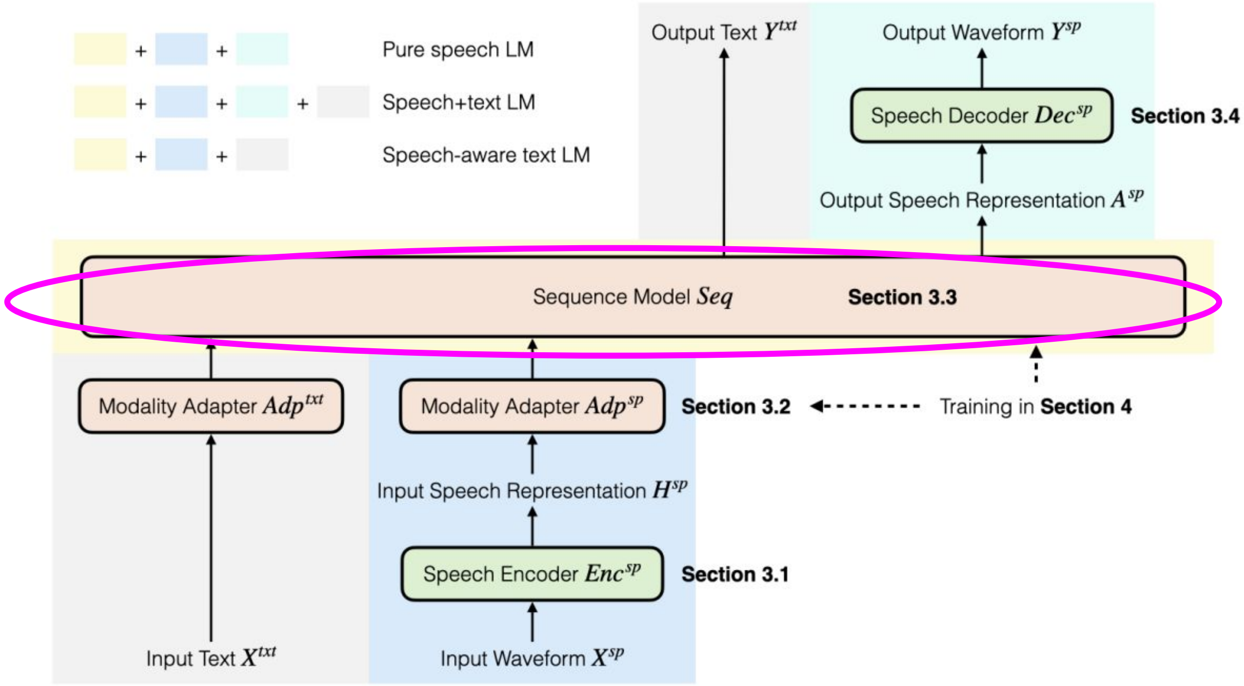


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

Let's talk about the **sequence model**

… not much to say, **this is the core LLM**

Let’s talk SLM architecture

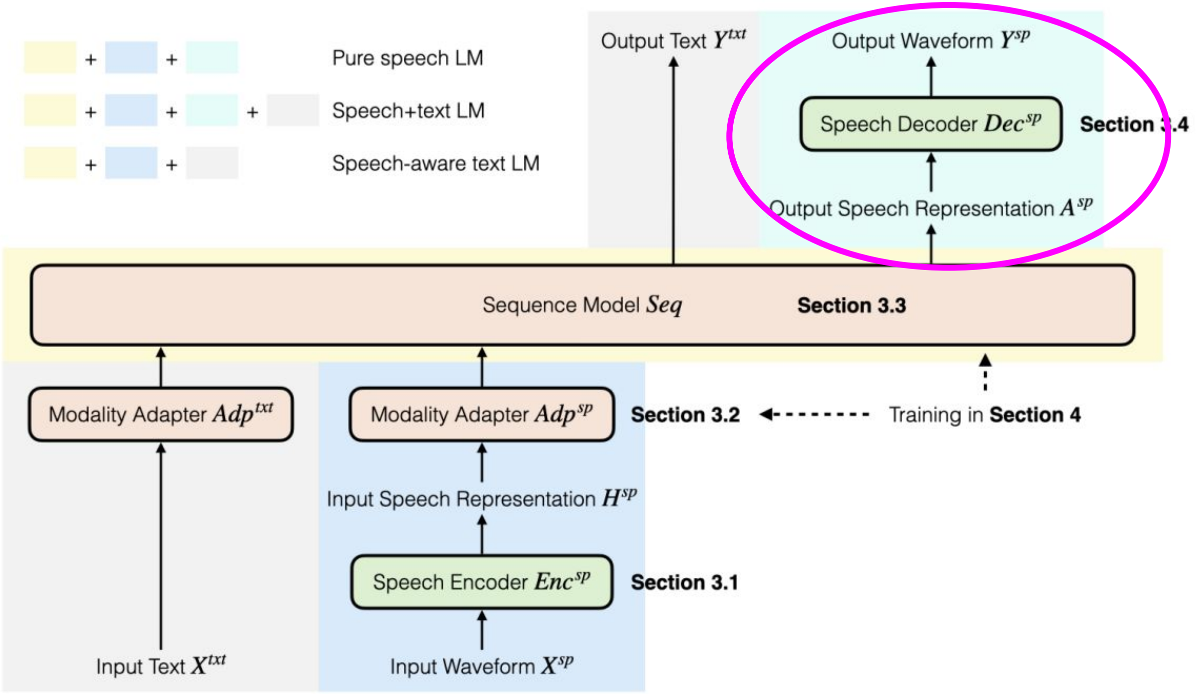


Figure 1: Overview of SLM architecture. See Sections 3 and 4 for more detailed descriptions of the components and training methods, respectively.

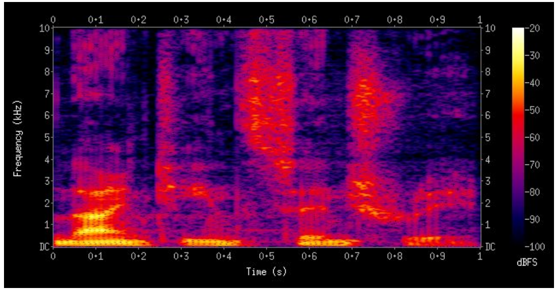
Let’s talk about the **speech decoder**

3.4 Speech Decoder

The speech decoder converts speech representations—whether continuous features, phonetic tokens, or audio codec tokens—back into waveforms. The speech decoder can take various forms:

1. Vocoder (Kong et al., 2020) for continuous features, similar to those used in traditional synthesis systems. For instance, in Spectron (Nachmani et al., 2024), a generated mel spectrogram is synthesized into audio using the WavFit vocoder (Koizumi et al., 2023).

They generate these →



2. Unit-based vocoder (Polyak et al., 2021) based on HiFi-GAN (Kong et al., 2020) for phonetic tokens. These vocoders take phonetic tokens as inputs and optionally combine them with additional information to improve synthesis quality. For example, when phonetic tokens are deduplicated, a duration modeling module is often included in the vocoder (Lakhotia et al., 2021).

3. Codec decoder (Guo et al., 2025). When the SLM generates audio codec tokens, these tokens can be input directly into the corresponding pre-trained audio neural codec decoder (without additional training) to get the waveform.

They generate codecs

3 Main Approaches to TRAIN 🚂 SLMs

Train the (big) sequence model (aka the LLM) *[most exp*

```
                   ❄️                                    🔥
Speech -> Speech Encoder --------------------------\
                                                    -> Sequence Model (fine-tu
                  ❄️
Text   -> Token Embedding -------------------------/
```

Let's revisit this…

Train the modality adapters *[least expensive $]*

```
             ❄️                  🔥                   ❄️
Speech -> Speech Encoder -> Modality Adapter -> Sequence Model
            ❄️                   🔥                   ❄️
Text   -> Token Embedding -> Modality Adapter -> Sequence Model
```

Train the speech encoder *[mid-expensive $$]* *[PS this is kinda like just training 1 modality adapter]*

```
               🔥                                                 ❄️
Speech -> Speech Encoder -> (aligned directly to LM token space) -> Sequence Model
               ❄️                                                  ❄️
Text   -> Token Embedding ----------------------------------> Sequence Model
```

They're doing the training with **special speech tokens!!!!**

$$p(\cdot|speech, instruction) \begin{cases} p(\cdot|speech, textinstruction), \\ p(\cdot|speech, speechinstruction) \end{cases}$$

Either one! Just depending on setup

TASK-SPECIFIC TUNING

INSTRUCTION TUNING

Token / Specifier	Training Paradigm	Task	Example Input	Expected Output
`<task:TRANSCRIBE>`	Task-specific	ASR	[speech: "Hello, how are you today?"]	`"Hello, how are you today?"`
`<task:SUMMARIZE>`	Task-specific	Speech Summarization	[speech: lecture audio]	`"The lecture explains Fourier transforms as frequency decompositions."`
`<task:TRANSLATE_EN>`	Task-specific	Speech Translation	[speech: "Bonjour, comment ça va?"]	`"Hello, how are you?"`
`<speech_instruction:CLASSIFY_EMOTION>`	Instruction-tuning	Emotion Detection	[speech: angry utterance]	`"angry"`
`<speech_instruction:CONVERSATION>`	Instruction-tuning	Dialogue Response	[speech: "What time is it?"]	`"It's 3:00 PM."`
`<speech_instruction:MULTIMODAL_QA>`	Instruction-tuning	Multimodal Fusion (Speech + Text)	[speech: "Look at this picture."] + [text: "What color is the car?"]	`"The car is red."`

Task-specific = "menu of fixed commands" (rigid, controlled).
Instruction tuning = "free-form instructions" (flexible, generalizable).

So let's talk about how you **design your loss** aka objective **function for training** 🔥

Train the (big) sequence model (aka the LLM) *[most expensive $$$]*

```
Speech -> Speech Encoder ------------------------\
                                                  -> Sequence Model (fine-tuned jointly)
Text   -> Token Embedding -----------------------/
```

Train the speech encoder *[mid-expensive $$] [PS this is kinda like just training 1 modality adapter]*

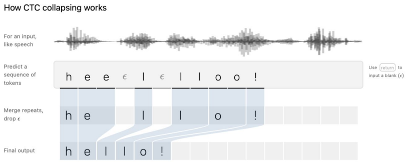
```
Speech -> Speech Encoder -> (aligned directly to LM token space) -> Sequence Model
Text   -> Token Embedding -------------------------------------> Sequence Model
```

Train the modality adapters *[least expensive $]*

```
Speech  -> Speech Encoder  -> Modality Adapter -> Sequence Model
Text    -> Token Embedding -> Modality Adapter -> Sequence Model
```

TASK-SPECIFIC TUNING

INSTRUCTION TUNING



If you want to do THIS

Set up your math like THIS

Stage	Objective / Loss	Formula
Pretraining (AR LM)	Autoregressive next-token prediction	$\mathcal{L}_{\text{AR}} = -\sum_t \log p(y_t \mid y_{<t}, x)$
Adapter / Alignment	L2 alignment	$\mathcal{L}_{\text{align}} = \lVert f_{\text{speech}}(X^{sp}) - f_{\text{text}}(X^{txt}) \rVert_2^2$
	Contrastive (CLIP-style)	$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp(\text{sim}(h^{sp}, h^{txt})/\tau)}{\sum_{h'} \exp(\text{sim}(h^{sp}, h')/\tau)}$
Task-specific training	Cross-entropy w/ task token	$\mathcal{L}_{\text{task}} = -\sum_t \log p(y_t \mid y_{<t}, X^{sp}, \langle task \rangle)$
Instruction tuning	Cross-entropy w/ instruction	$\mathcal{L}_{\text{instr}} = -\sum_t \log p(y_t \mid y_{<t}, X^{sp}, \text{instruction})$
Specialized (optional)	CTC (speech alignment)	$\mathcal{L}_{\text{CTC}} = -\log p(\text{transcript} \mid X^{sp})$
	Reconstruction	$\mathcal{L}_{\text{recon}} = \lVert X^{sp} - \hat{X}^{sp} \rVert_2^2$
Multi-task combo	Weighted mix	$\mathcal{L} = \sum_i \lambda_i \mathcal{L}_i$

You'll probably end up combo-ing multiple of these together

This is cool: interruption handling w/ a duplex model

Duplex = 2 parallel streams for user and SLM, open at all times → robust to interruptions, no assumption of "turn-taking" really

Walkie-Talkie vs. Phone Call

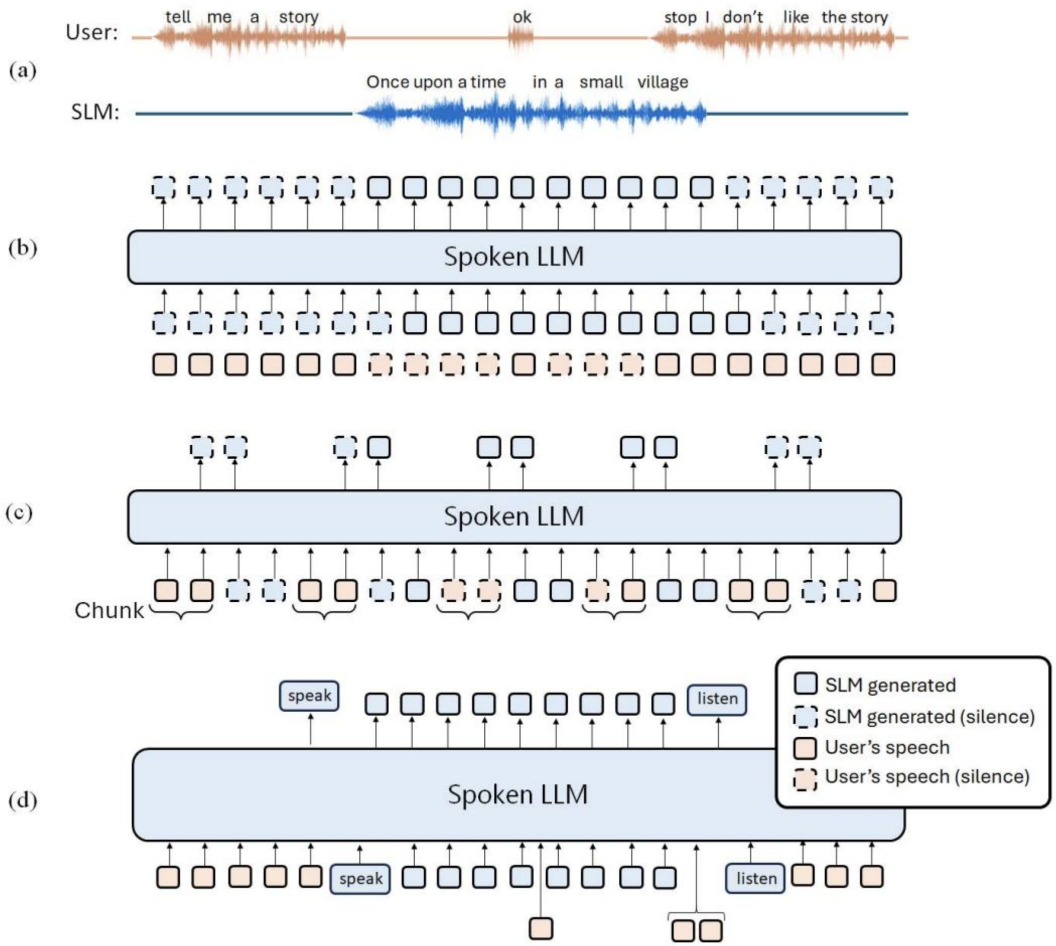


Figure 5: Full-duplex speech conversation. (a) An example of full-duplex speech conversation between a user and an SLM. (b) Dual-channel approach. (c) Time-multiplexing approach (with equal chunks). (d) Time-multiplexing approach (where the SLM controls the switching between listening and speaking modes).

8 Challenges and future work

Model architecture The optimal representation of speech within SLMs remains unclear. Speech representations in SLMs include both discrete and continuous varieties, derived from a wide range of encoders. This design choice can also influence other architectural choices in an SLM, for example depending on the information rate of the encoder and whether it encodes more phonetic or other types of information.

Another open question is determining the best method to combine speech and text, which applies to all aspects of SLM modeling and training. We have described various choices of modality adapters and approaches for interleaving speech and text. These have not been thoroughly compared, so the effect of each modeling choice is still unclear.

A final architectural challenge is that current SLMs are large and slow, making them impractical for real-time and on-device settings. To some extent this is because various compression algorithms (e.g., (Lai et al., 2021; Peng et al., 2023a; Ding et al., 2024)) and alternative architectures (e.g., Park et al. (2024)) have not been widely applied to SLMs. However, there is also an inherent efficiency challenge that arises when combining multiple pre-trained components, sometimes with different architectures and frame rates.

Second,

Let’s talk about the robustness of pipeline approaches.

Pipeline Approach vs. End-to-End Approach

ASR → **LLM** → **TTS**

SLM

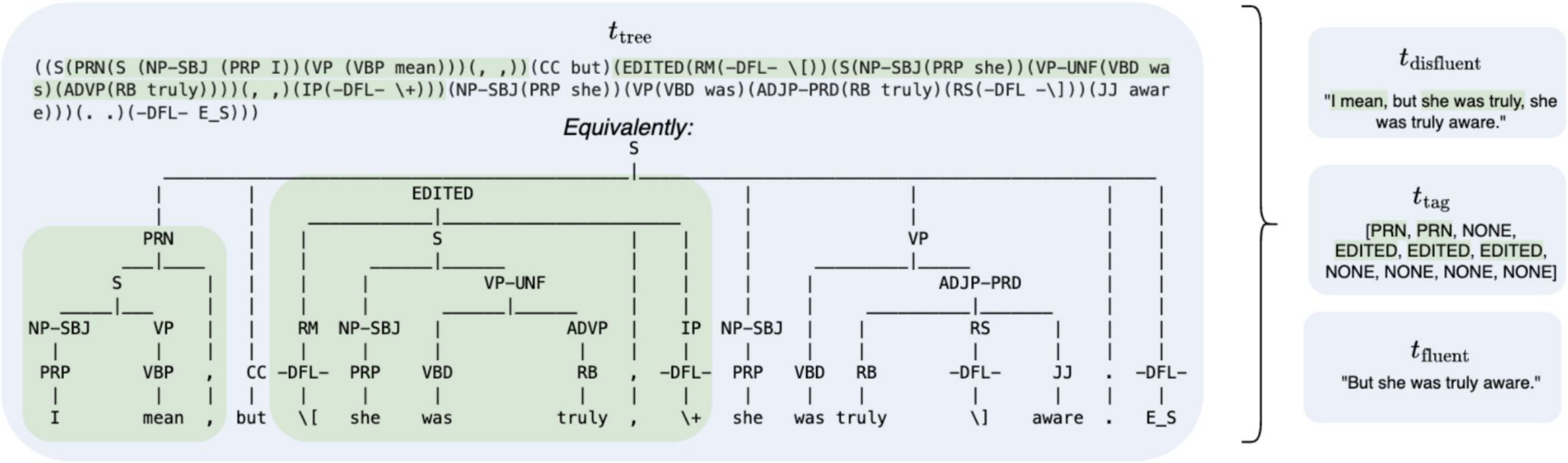
✅ preferred by industry for *controllability* → think <u>custom vocabulary</u> & ease of <u>debugging</u>

✅ currently in research stage but very promising → major barrier imo is downsampling problem (will explain)

Pipeline approaches are super susceptible to issues processing **disfluencies**:

Fig. 1. Removing disfluencies such as INTJ (*uh*), EDITED (*gas station* is replaced with *grocery store*), and PRN (*you know*) ensures clean text input for downstream tasks. Our

Pipeline approaches are super susceptible to issues processing **disfluencies**:



Quantifying the Impact of Disfluency on Spoken Content Summarization

Maria Teleki, Xiangjue Dong, James Caverlee
Texas A&M University
College Station, Texas, USA
{mariateleki, xj.dong, caverlee}@tamu.edu

LREC-COLING '24

SImple disfluencies can kill model performance.

Original

Hello and welcome to our podcast! Let's get right to it. Today we're going to be interviewing a very special guest, someone I know you guys have been excited about having on the show.

Repeats with N=3

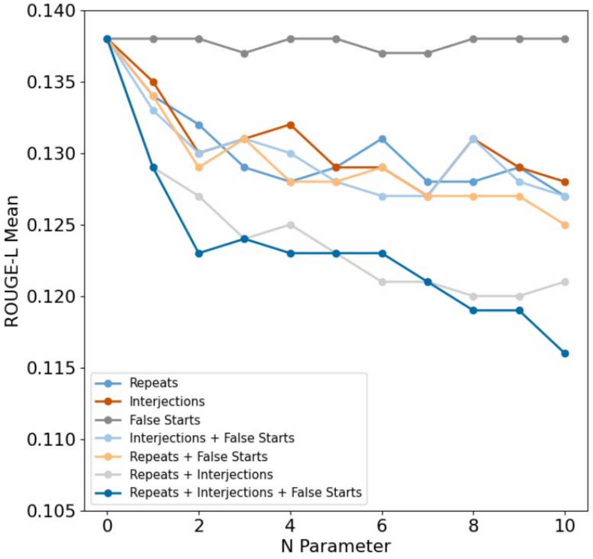
Hello and welcome to our podcast! Let's get **get get get** right to it. Today we're going to be interviewing a **a a a** very special guest, someone I know you guys have been excited about having on the show.

Interjections with N=3

Hello and welcome to our podcast! Let's get right **uh okay okay** to it. Today we're going to be interviewing a very special **um so I mean** guest, someone I know you guys have been excited about having on the show.

False Starts with N=3

Hello and welcome to our podcast! Let's get right to it. Today we're **today we're today we're today we're** going to be interviewing a very special guest, someone I know you guys have been excited about having on the show.



(a) ROUGE-L over increased N on **BART** model.

Decrease Rouge-L

Increase N

Ok but BART's old…

I agree! Here's our recent work evaluating:

- gpt-{4o,4o-mini,o4}
- llama-{1B,3B,8B,70B}
- qwen-{0.6B,1.7B,4B,8B}
- phi-4-mini
- mobileLLM{125M,350M,600M,1B}

DRES: BENCHMARKING LLMS FOR DISFLUENCY REMOVAL

Maria Teleki, Sai Janjur, Haoran Liu, Oliver Grabner, Ketan Verma, Thomas Docog, Xiangjue Dong, Lingfeng Shi, Cong Wang, Stephanie Birkelbach, Jason Kim, Yin Zhang, James Caverlee

Texas A&M University

ArXiv '25

Long story short, these models still struggle!

M	k	$\mathcal{E}_F$	$\mathcal{E}_P$	$\mathcal{E}_R$	$\mathcal{Z}_E$	$\mathcal{Z}_I$	$\mathcal{Z}_P$	$\mathcal{E}_F$	$\mathcal{E}_P$	$\mathcal{E}_R$	$\mathcal{Z}_E$	$\mathcal{Z}_I$	$\mathcal{Z}_P$
		gpt-4o						**gpt-4o-mini**					
f	0	$74.19_{7.87}$	$77.91_{10.63}$	$73.18_{12.98}$	$83.86_{13.15}$	$71.25_{16.57}$	$52.52_{31.91}$	$70.52_{9.22}$	$75.56_{9.89}$	$68.17_{13.90}$	$86.13_{10.30}$	$60.38_{18.22}$	$55.26_{26.17}$
f	1	$76.13_{7.63}$	$76.94_{10.48}$	$78.02_{13.05}$	$85.11_{11.73}$	$77.37_{15.64}$	$62.71_{29.59}$	$68.85_{9.53}$	$74.35_{9.65}$	$66.02_{14.04}$	$86.05_{10.37}$	$57.62_{19.88}$	$52.04_{24.06}$
f	3	$73.99_{9.33}$	$81.33_{10.10}$	$70.75_{15.11}$	$77.83_{17.30}$	$72.93_{15.26}$	$49.35_{29.73}$	$68.70_{10.64}$	$73.20_{11.59}$	$67.68_{15.21}$	$86.64_{10.10}$	$61.42_{20.83}$	$51.58_{25.42}$
f	5	$73.06_{8.04}$	$81.24_{10.69}$	$68.75_{13.39}$	$76.92_{15.55}$	$71.49_{13.97}$	$42.39_{30.38}$	$69.03_{11.17}$	$76.93_{13.18}$	$65.86_{14.69}$	$84.77_{11.10}$	$59.46_{19.58}$	$48.78_{24.87}$
s	0	$78.17_{4.04}$	$78.44_{5.83}$	$78.30_{5.27}$	$82.92_{8.67}$	$77.55_{7.16}$	$69.80_{20.56}$	$72.69_{5.79}$	$75.61_{7.05}$	$70.48_{7.35}$	$85.20_{8.23}$	$61.89_{11.08}$	$65.02_{20.99}$
s	1	$82.38_{4.18}$	$83.61_{6.03}$	$81.53_{5.17}$	$83.77_{9.41}$	$79.74_{6.94}$	$79.65_{19.40}$	$77.69_{5.08}$	$81.84_{6.00}$	$74.34_{6.81}$	$84.56_{9.47}$	$65.46_{10.36}$	$77.59_{20.30}$
s	3	$83.72_{3.85}$	$85.64_{5.28}$	$82.22_{5.41}$	$83.42_{8.90}$	$81.94_{6.66}$	$78.26_{19.29}$	$77.01_{4.82}$	$82.10_{5.73}$	$72.88_{6.56}$	$83.50_{9.32}$	$64.80_{9.94}$	$73.83_{17.24}$
s	5	$84.52_{3.35}$	$88.09_{4.84}$	$81.56_{5.15}$	$81.15_{9.70}$	$82.97_{6.39}$	$77.14_{19.05}$	$77.76_{4.69}$	$83.31_{5.42}$	$73.29_{6.61}$	$83.84_{9.17}$	$65.12_{10.14}$	$74.69_{18.04}$
		medium-o4-mini						**high-o4-mini**					
f	0	$48.18_{9.41}$	$33.04_{9.81}$	$95.43_{4.92}$	$96.90_{6.46}$	$93.66_{7.38}$	$96.51_{7.17}$	$55.81_{11.28}$	$41.40_{12.59}$	$92.69_{7.82}$	$95.49_{7.10}$	$90.49_{10.52}$	$93.48_{12.34}$
f	1	$40.71_{8.79}$	$26.19_{7.57}$	$97.49_{2.44}$	$98.19_{2.50}$	$96.40_{4.27}$	$98.68_{3.44}$	$46.03_{11.09}$	$31.17_{10.83}$	$96.57_{4.03}$	$97.42_{5.44}$	$95.39_{6.38}$	$97.77_{5.13}$
f	3	$39.65_{7.74}$	$25.20_{6.42}$	$97.74_{2.21}$	$98.67_{1.94}$	$96.84_{3.51}$	$98.40_{4.24}$	$42.91_{9.50}$	$28.11_{8.49}$	$97.27_{2.69}$	$98.02_{2.93}$	$96.35_{4.17}$	$97.64_{5.85}$
f	5	$38.68_{8.02}$	$24.43_{6.63}$	$98.07_{2.02}$	$98.89_{1.64}$	$97.45_{3.03}$	$98.00_{5.26}$	$41.71_{9.26}$	$27.03_{8.01}$	$97.62_{2.15}$	$98.15_{2.51}$	$97.04_{3.03}$	$97.92_{5.04}$
		Llama-1B-Instruct						**Llama-3B-Instruct**					
f	0	$35.27_{17.26}$	$54.71_{22.71}$	$36.72_{26.13}$	$44.18_{26.76}$	$35.97_{27.80}$	$23.35_{27.59}$	$58.45_{10.66}$	$49.96_{15.09}$	$78.76_{14.57}$	$84.46_{13.29}$	$77.40_{15.91}$	$69.96_{25.35}$
f	1	$33.35_{11.37}$	$28.67_{17.55}$	$72.58_{29.09}$	$75.84_{26.84}$	$72.69_{30.26}$	$66.53_{35.43}$	$50.45_{13.19}$	$41.22_{17.23}$	$81.70_{18.33}$	$86.01_{17.16}$	$82.32_{18.14}$	$71.11_{29.67}$
f	3	$32.20_{11.45}$	$26.65_{17.91}$	$77.77_{28.82}$	$80.83_{26.69}$	$77.41_{29.70}$	$73.35_{33.29}$	$49.24_{15.20}$	$46.74_{19.42}$	$69.55_{24.60}$	$76.19_{23.61}$	$69.10_{25.99}$	$56.35_{33.18}$
f	5	$35.60_{13.72}$	$34.99_{21.79}$	$68.37_{31.82}$	$73.13_{29.46}$	$68.51_{32.72}$	$59.15_{39.18}$	$48.74_{14.78}$	$50.54_{17.93}$	$60.98_{26.24}$	$67.52_{26.10}$	$62.42_{27.51}$	$45.47_{32.88}$
s	0	$61.88_{5.82}$	$68.31_{8.65}$	$57.49_{7.39}$	$56.38_{10.73}$	$70.81_{7.12}$	$27.84_{16.32}$	$67.52_{5.90}$	$65.81_{8.48}$	$70.20_{6.80}$	$77.96_{8.24}$	$69.57_{9.17}$	$57.13_{19.47}$
s	1	$32.98_{6.22}$	$29.09_{8.45}$	$40.70_{8.26}$	$43.15_{13.15}$	$41.44_{8.28}$	$34.47_{18.00}$	$48.69_{8.88}$	$45.74_{13.31}$	$54.80_{7.07}$	$57.20_{10.45}$	$53.30_{8.21}$	$53.65_{19.15}$
s	3	$38.26_{6.12}$	$31.37_{7.59}$	$51.71_{8.19}$	$53.39_{11.33}$	$56.34_{8.26}$	$38.24_{18.11}$	$53.78_{8.89}$	$57.00_{15.41}$	$52.86_{7.06}$	$56.94_{11.02}$	$54.27_{9.37}$	$40.14_{17.67}$
s	5	$39.34_{6.97}$	$30.24_{7.93}$	$59.38_{8.02}$	$62.51_{10.85}$	$63.46_{8.22}$	$44.34_{17.61}$	$60.43_{7.84}$	$71.60_{13.85}$	$53.22_{6.82}$	$60.13_{11.05}$	$55.63_{9.75}$	$34.58_{16.03}$
		Llama-8B-Instruct						**Llama-70B-Instruct**					
f	0	$45.48_{9.37}$	$31.93_{9.82}$	$87.43_{12.08}$	$88.57_{13.03}$	$88.83_{11.67}$	$81.12_{22.77}$	$67.83_{9.90}$	$63.90_{16.75}$	$78.48_{13.30}$	$81.14_{13.92}$	$79.48_{14.76}$	$69.48_{26.23}$
f	1	$37.46_{11.94}$	$27.99_{15.68}$	$80.38_{22.39}$	$81.35_{22.81}$	$81.58_{22.97}$	$75.18_{27.90}$	$62.85_{12.61}$	$58.99_{17.71}$	$74.55_{15.59}$	$78.90_{16.01}$	$74.84_{16.51}$	$66.90_{26.51}$
f	3	$30.32_{5.90}$	$18.02_{4.19}$	$99.94_{0.60}$	$100.00_{0.00}$	$99.94_{0.59}$	$99.85_{2.03}$	$58.37_{12.91}$	$48.95_{17.06}$	$82.83_{14.57}$	$85.32_{12.83}$	$84.17_{16.30}$	$75.74_{23.83}$
f	5	$30.33_{5.91}$	$18.02_{4.19}$	$100.00_{0.00}$	$100.00_{0.00}$	$100.00_{0.00}$	$100.00_{0.00}$	$53.37_{14.54}$	$44.39_{17.41}$	$82.87_{18.75}$	$83.42_{18.37}$	$85.37_{18.61}$	$76.18_{27.59}$
s	0	$68.30_{5.95}$	$63.97_{8.94}$	$74.22_{5.58}$	$80.84_{8.19}$	$74.66_{7.23}$	$60.62_{20.33}$	$76.14_{4.84}$	$77.87_{7.82}$	$75.10_{5.53}$	$73.61_{10.81}$	$76.36_{5.50}$	$71.56_{21.92}$
s	1	$68.90_{6.98}$	$67.91_{9.82}$	$70.60_{6.10}$	$75.98_{9.23}$	$66.07_{8.72}$	$69.04_{19.92}$	$68.31_{8.47}$	$75.97_{14.31}$	$63.25_{6.43}$	$61.65_{11.28}$	$60.78_{8.25}$	$68.39_{20.89}$
s	3	$65.50_{6.72}$	$66.80_{9.56}$	$65.06_{7.17}$	$71.32_{9.56}$	$60.78_{8.39}$	$63.12_{20.91}$	$63.20_{8.53}$	$72.30_{15.56}$	$57.72_{7.17}$	$53.18_{11.18}$	$56.67_{9.35}$	$64.54_{21.82}$
s	5	$66.65_{6.50}$	$67.05_{9.37}$	$66.98_{6.41}$	$74.17_{9.45}$	$63.13_{9.31}$	$62.23_{21.11}$	$65.99_{7.41}$	$76.46_{13.85}$	$59.26_{6.86}$	$54.21_{11.52}$	$59.15_{9.26}$	$64.88_{20.04}$
		Qwen3-0.6B						**Qwen3-1.7B**					
f	0	$18.04_{9.30}$	$43.29_{26.59}$	$16.46_{14.61}$	$22.51_{17.29}$	$14.26_{14.25}$	$11.56_{15.87}$	$10.25_{9.49}$	$86.02_{20.54}$	$5.91_{7.31}$	$10.36_{9.49}$	$4.10_{7.96}$	$2.75_{6.48}$
f	1	$22.36_{9.02}$	$29.02_{20.01}$	$30.24_{26.06}$	$35.09_{26.05}$	$28.20_{26.43}$	$27.18_{29.81}$	$10.12_{9.34}$	$79.60_{29.76}$	$13.39_{28.55}$	$16.82_{27.74}$	$11.84_{29.26}$	$10.70_{28.97}$
f	3	$21.17_{9.28}$	$38.69_{23.36}$	$20.34_{14.32}$	$24.92_{17.04}$	$19.29_{13.91}$	$14.89_{17.93}$	$7.78_{6.27}$	$88.20_{21.59}$	$5.09_{10.74}$	$8.82_{12.02}$	$3.24_{10.26}$	$2.98_{11.34}$
f	5	$19.87_{9.99}$	$40.95_{26.37}$	$19.67_{15.96}$	$24.09_{18.15}$	$17.87_{15.44}$	$16.06_{19.42}$	$8.64_{6.49}$	$85.42_{24.87}$	$6.66_{15.30}$	$10.71_{15.52}$	$4.88_{15.55}$	$4.17_{15.88}$
s	0	$43.74_{6.31}$	$44.90_{9.86}$	$44.05_{7.56}$	$60.78_{11.78}$	$33.34_{8.34}$	$41.94_{19.35}$	$36.00_{8.24}$	$71.41_{10.04}$	$24.79_{7.60}$	$39.13_{11.40}$	$18.96_{7.61}$	$14.56_{15.02}$
s	1	$51.97_{6.39}$	$47.23_{9.62}$	$59.72_{6.79}$	$70.39_{9.05}$	$55.32_{8.05}$	$53.93_{17.38}$	$35.78_{7.46}$	$65.77_{11.56}$	$25.10_{6.39}$	$23.24_{7.90}$	$33.64_{8.64}$	$8.18_{9.65}$
s	3	$48.90_{6.31}$	$45.89_{8.84}$	$54.00_{8.14}$	$62.76_{10.75}$	$52.82_{9.24}$	$42.36_{18.94}$	$31.26_{7.05}$	$67.30_{10.40}$	$20.75_{5.80}$	$18.55_{7.01}$	$28.84_{8.57}$	$5.09_{7.21}$
s	5	$48.38_{6.33}$	$48.74_{10.75}$	$50.29_{8.46}$	$58.66_{11.18}$	$50.05_{9.90}$	$36.34_{18.73}$	$34.46_{7.97}$	$81.04_{11.03}$	$22.31_{6.50}$	$23.00_{8.94}$	$29.00_{8.67}$	$5.08_{6.61}$
		Qwen3-4B						**Qwen3-8B**					
f	0	$66.39_{16.56}$	$75.58_{16.33}$	$64.58_{20.96}$	$62.47_{23.52}$	$70.30_{22.66}$	$49.66_{31.16}$	$71.04_{10.77}$	$69.94_{12.12}$	$76.17_{15.49}$	$75.24_{18.17}$	$79.32_{14.48}$	$67.37_{29.01}$
f	1	$59.89_{20.20}$	$79.63_{14.52}$	$54.23_{23.08}$	$55.44_{26.35}$	$59.91_{24.54}$	$35.45_{28.72}$	$68.86_{13.90}$	$74.54_{15.50}$	$69.69_{17.63}$	$67.92_{20.37}$	$76.12_{16.61}$	$54.80_{30.83}$
f	3	$62.29_{20.97}$	$80.58_{14.28}$	$57.13_{22.87}$	$54.85_{23.60}$	$65.16_{24.09}$	$39.32_{28.43}$	$71.77_{11.73}$	$76.82_{10.36}$	$70.78_{16.75}$	$68.89_{18.34}$	$77.84_{15.80}$	$54.49_{30.81}$
s	0	$68.48_{5.55}$	$67.96_{8.94}$	$69.95_{6.29}$	$70.68_{10.41}$	$72.43_{7.71}$	$62.02_{19.69}$	$71.46_{6.01}$	$78.32_{7.62}$	$66.17_{7.12}$	$73.77_{10.23}$	$63.29_{10.24}$	$57.96_{22.96}$
s	1	$69.66_{5.59}$	$80.77_{7.32}$	$61.60_{6.43}$	$72.82_{10.90}$	$54.30_{9.35}$	$57.76_{22.39}$	$70.91_{5.30}$	$70.41_{8.29}$	$72.25_{6.22}$	$73.13_{10.83}$	$74.45_{8.67}$	$62.03_{22.19}$
s	3	$64.14_{6.23}$	$80.56_{7.25}$	$53.72_{7.15}$	$66.76_{10.58}$	$47.47_{10.41}$	$44.88_{22.09}$	$67.78_{5.87}$	$74.06_{7.97}$	$63.12_{7.34}$	$68.64_{11.34}$	$63.81_{9.49}$	$50.23_{20.53}$
s	5	$63.37_{6.55}$	$83.74_{7.48}$	$51.46_{7.47}$	$64.45_{11.10}$	$44.56_{11.12}$	$44.35_{22.42}$	$66.64_{5.87}$	$73.42_{7.91}$	$61.50_{6.73}$	$65.22_{10.97}$	$63.24_{9.64}$	$50.63_{21.16}$
		Phi-4-mini-instruct						**Phi-4-mini-reasoning**					
f	0	$27.65_{10.63}$	$27.41_{16.18}$	$58.22_{36.39}$	$63.15_{35.84}$	$54.33_{37.21}$	$59.29_{39.26}$	$31.72_{6.26}$	$20.17_{5.30}$	$85.91_{18.36}$	$88.66_{16.58}$	$84.47_{20.18}$	$85.25_{21.16}$
f	1	$26.03_{9.22}$	$25.09_{15.37}$	$60.63_{36.65}$	$65.80_{37.01}$	$55.60_{36.94}$	$63.76_{39.45}$	$30.84_{6.30}$	$20.22_{6.15}$	$86.03_{24.10}$	$88.99_{20.86}$	$84.17_{26.94}$	$86.54_{24.74}$
f	3	$27.00_{10.51}$	$32.71_{15.59}$	$35.84_{25.59}$	$42.49_{26.41}$	$32.25_{25.59}$	$33.76_{29.60}$	$33.79_{7.53}$	$23.06_{6.88}$	$79.18_{22.99}$	$82.80_{21.91}$	$77.84_{24.47}$	$77.43_{24.64}$
f	5	$25.64_{9.46}$	$33.52_{16.85}$	$32.80_{23.46}$	$39.64_{24.78}$	$28.62_{23.47}$	$31.76_{29.64}$	$34.42_{7.32}$	$24.37_{8.47}$	$75.20_{22.62}$	$79.17_{21.61}$	$73.97_{23.50}$	$71.52_{27.27}$
s	0	$36.91_{6.51}$	$25.42_{6.02}$	$70.73_{7.71}$	$75.57_{10.01}$	$72.50_{7.88}$	$58.36_{16.58}$	$36.83_{6.72}$	$26.71_{7.11}$	$62.88_{8.07}$	$66.11_{11.09}$	$64.61_{8.33}$	$51.63_{19.00}$
s	1	$39.14_{6.56}$	$28.45_{6.76}$	$66.12_{7.60}$	$67.84_{10.63}$	$71.55_{6.54}$	$48.62_{16.75}$	$36.94_{6.40}$	$24.86_{5.83}$	$75.31_{6.83}$	$78.67_{9.96}$	$76.94_{6.47}$	$65.10_{17.50}$
s	3	$44.50_{7.70}$	$34.74_{8.99}$	$64.79_{7.32}$	$67.76_{10.90}$	$69.29_{7.01}$	$48.68_{18.34}$	$37.15_{6.03}$	$25.41_{5.79}$	$72.40_{6.56}$	$74.39_{10.97}$	$76.00_{6.41}$	$59.47_{17.63}$
s	5	$49.37_{8.37}$	$42.15_{11.78}$	$62.92_{7.08}$	$66.12_{10.61}$	$67.10_{7.91}$	$47.13_{16.23}$	$38.44_{6.03}$	$26.44_{5.67}$	$73.07_{6.47}$	$75.58_{8.86}$	$75.58_{6.43}$	$62.29_{14.12}$
		MobileLLM-125M						**MobileLLM-350M**					
s	0	$33.23_{6.19}$	$20.58_{4.79}$	$90.18_{4.51}$	$91.42_{6.51}$	$91.95_{4.72}$	$83.76_{12.10}$	$34.25_{6.38}$	$21.85_{5.15}$	$82.74_{5.47}$	$82.78_{8.54}$	$85.27_{5.10}$	$76.78_{14.04}$
s	1	$33.90_{6.30}$	$21.13_{4.85}$	$89.38_{4.80}$	$91.02_{6.24}$	$90.79_{4.63}$	$83.87_{11.54}$	$35.16_{6.37}$	$24.75_{6.22}$	$63.83_{7.27}$	$68.40_{10.90}$	$64.54_{7.66}$	$55.39_{17.50}$
s	3	$34.40_{6.36}$	$22.20_{5.25}$	$80.11_{6.58}$	$84.82_{8.07}$	$80.91_{6.72}$	$70.45_{15.53}$	$36.35_{6.33}$	$28.03_{7.01}$	$54.38_{7.73}$	$60.63_{11.27}$	$54.72_{7.83}$	$44.07_{17.69}$
s	5	$34.99_{6.17}$	$22.64_{5.11}$	$80.40_{5.77}$	$83.64_{7.59}$	$84.12_{5.00}$	$66.45_{15.72}$	$37.84_{6.73}$	$31.96_{8.82}$	$49.07_{7.26}$	$57.45_{9.64}$	$48.28_{7.38}$	$38.26_{18.38}$
		MobileLLM-600M						**MobileLLM-1B**					
s	0	$33.96_{6.45}$	$22.03_{5.30}$	$77.53_{7.00}$	$79.65_{10.03}$	$78.33_{6.82}$	$71.91_{16.25}$	$33.38_{6.05}$	$22.40_{5.36}$	$68.78_{7.75}$	$69.21_{11.51}$	$71.55_{7.65}$	$62.87_{17.81}$
s	1	$36.13_{6.33}$	$27.10_{6.48}$	$56.91_{8.40}$	$61.45_{12.22}$	$58.28_{8.32}$	$45.65_{17.68}$	$33.12_{5.92}$	$22.81_{5.45}$	$63.68_{8.23}$	$66.74_{11.72}$	$63.90_{7.93}$	$57.95_{18.52}$
s	3	$36.18_{6.44}$	$29.21_{7.69}$	$49.84_{7.57}$	$57.54_{11.21}$	$48.52_{7.63}$	$40.83_{17.95}$	$33.55_{5.76}$	$23.94_{5.71}$	$59.55_{8.39}$	$61.97_{12.41}$	$61.06_{8.38}$	$52.16_{18.97}$
s	5	$36.98_{6.48}$	$30.83_{7.65}$	$48.51_{7.86}$	$57.93_{11.09}$	$46.94_{7.56}$	$36.70_{17.85}$	$34.57_{6.05}$	$25.51_{6.31}$	$56.64_{7.68}$	$59.28_{12.09}$	$58.67_{7.50}$	$47.52_{18.44}$

Table 3. ***k*-Shot Results**: We highlight the 1st and 2nd highest and 1st and 2nd lowest $\mathcal{E}$-Scores ($mean_{std.\ dev}$), which are then annotated with the highest and lowest $\mathcal{Z}$-Scores ($mean_{std.\ dev}$), cases with low $\mathcal{E}_P$ and high $\mathcal{E}_R$ (an over-deletion failure mode), and cases with high $\mathcal{E}_P$ and low $\mathcal{E}_R$ (an under-deletion failure mode).

So we make a set of concrete recommendations to help these models perform better on disfluent, spoken data!

3. RESULTS & RECOMMENDATIONS

We analyze the disfluency removal behavior of LLMs and provide recommendations (R1-R9).

Open-Source vs. Proprietary. Looking to Table 3, proprietary models (gpt-4o, gpt-4o-mini) achieve the highest scores, with margins of 10–15 points over the best open-source alternatives. We attribute this to training exposure to Whisper-transcribed speech data [24]. **(R1) Proprietary models are currently the most reliable for production systems, while open-source models require targeted augmentation with spoken data.**

Segmentation (s) vs. Full Input (f). Segmenting transcripts consistently improves both mean performance and stability, e.g., gpt-4o improves from $\mathcal{E}_F$=76.13 (f) to 82.38 (s) at k=1. This supports prior evidence of long-context degradation in LLMs [25, 26]. **(R2) Segmentation is an effective preprocessing step that should be applied.**

Few-Shot Sensitivity (k). Increasing k does not uniformly improve results. Small models (e.g., MobileLLM) gain slightly, but others show degradation (e.g. Llama-3B/8B/70B) when more examples are provided. **(R3) Few-shot prompting should be used with caution, as some model families misinterpret exemplars and over-edit fluent text.**

Disfluency Category Performance. $\mathcal{Z}$-Scores show that EDITED nodes are handled well, but INTJ and PRN nodes are frequently missed, despite prior work suggesting these are the easiest to detect [19, 17]. **(R4) Future modeling should focus on under-served categories (INTJ, PRN) to improve robustness across all disfluency types.**

Over-Deletion Failures. Several models (e.g., Llama-8B, o4-mini) achieve near perfect recall but at the cost of very low precision, deleting fluent tokens. Segmentation often mitigates this collapse mode. **(R5) Segment-level evaluation helps reduce over-deletion risk.**

Under-Deletion Failures. Some models (e.g., Qwen series) exhibit the opposite trend of over-deletion, achieving high precision but low recall (purple). These models preserve most fluent tokens but fail to remove many true disfluencies, especially in INTJ and PRN categories. This reflects conservative editing strategies and limited exposure to conversational disfluency distributions. **(R6) Models prone to under-deletion require additional filtering or targeted fine-tuning to ensure sufficient disfluency coverage.**

Reasoning-Oriented Models. Models tuned for reasoning (o4-mini, Phi-4) perform poorly, showing high recall but extreme over-deletion (blue). **(R7) Reasoning capability does not translate to disfluency removal; specialized evaluation remains necessary.**

Impact of Model Size. Model scaling generally improves disfluency removal, with Qwen, GPT, and Llama families showing upward trends. However, gains are nonlinear – e.g., Qwen3-1.7B underperforms both smaller and larger variants – likely due to training data or optimization differences rather than capacity limits. **(R8) Model choice should be guided by empirical benchmarks on target domains and disfluency categories rather than size alone.**

Fine-Tuning and Generalization. Looking to Tables 4 and 5, fine-tuning improves performance to near SOTA levels (e.g., gpt-4o-mini$_{ft}$ achieves $\mathcal{E}_P$=96.6), but evaluation on GSM8K, MMLU, and CoQA shows degraded performance on unrelated tasks. **(R9) Fine-tuning is suitable for dedicated disfluency pipelines, but not for general-purpose conversational models.**

Ok, so that's LLMs – what about ASR systems?

Comparing ASR Systems in the Context of Speech Disfluencies

Maria Teleki[1], *Xiangjue Dong*[1], *Soohwan Kim*[1], *James Caverlee*[1]

[1]Texas A&M University, USA

{mariateleki,xj.dong,cocomox26,caverlee}@tamu.edu

INTERSPEECH '24

→ TLDR: Whisper!

Last,

Let's talk about
speaker variation!

Masculine Defaults via Gendered Discourse in Podcasts and Large Language Models

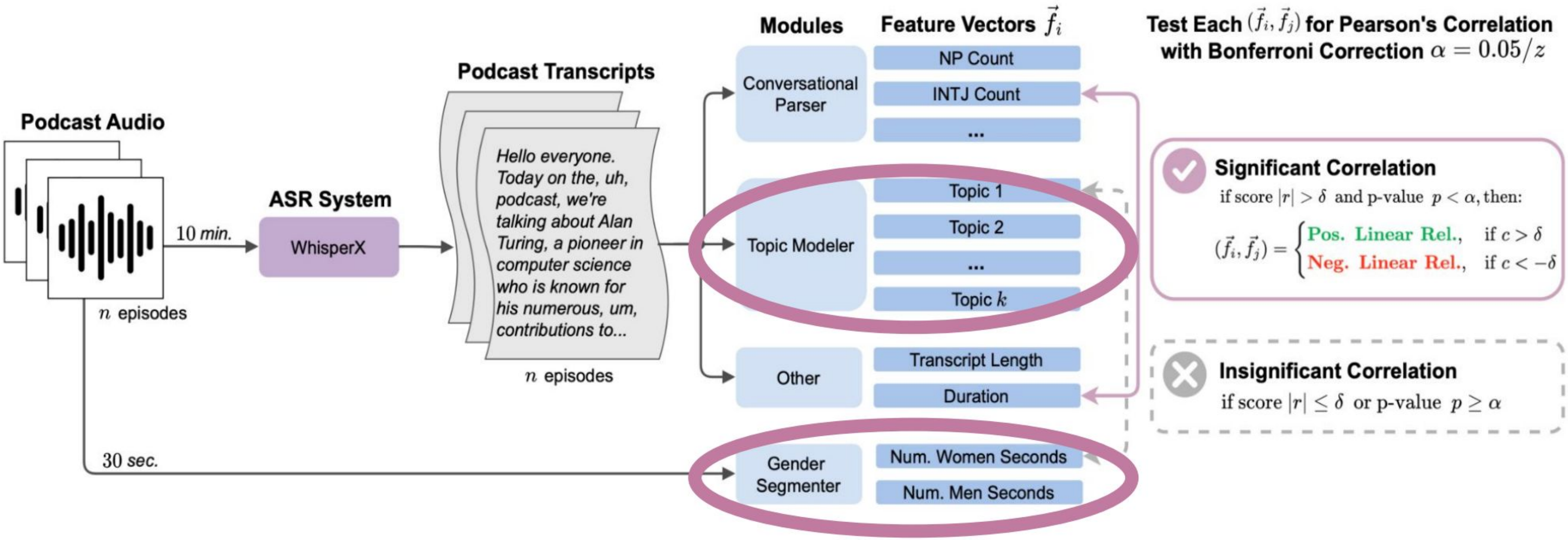
Maria Teleki, Xiangjue Dong, Haoran Liu, James Caverlee

Texas A&M University

{mariateleki, xj.dong, liuhr99, caverlee}@tamu.edu

ICWSM '25 + IC2S2 '25, SiCon@ACL '25

We propose the Gendered* Discourse Correlation Framework (GDCF) to monitor & identify discourse terms at scale.



* we use binary gender due to speech tool limitations - this is an important direction for future work.

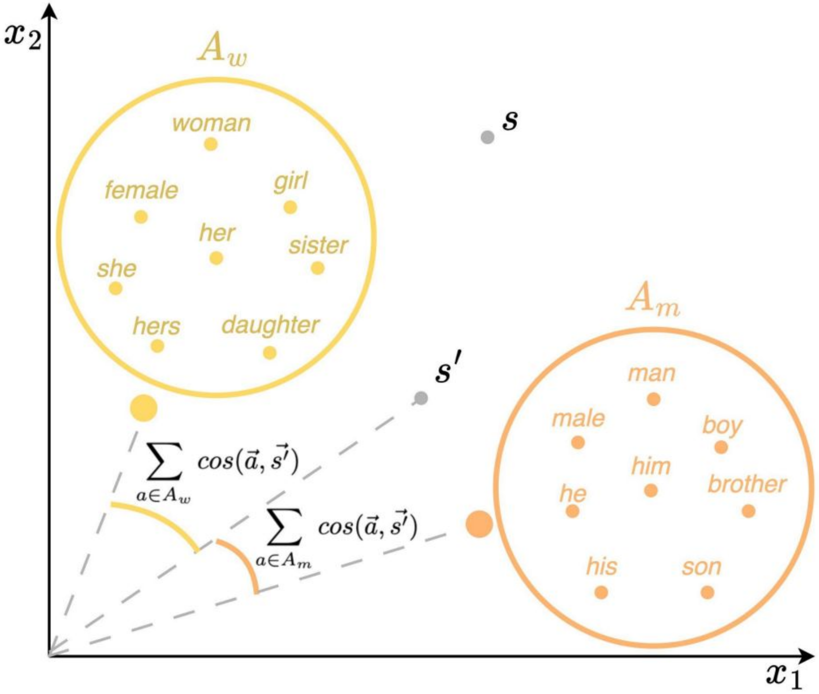
We find that women & men's speaking patterns are **different**.

Topic N	Gender	r	Topic N Word List	Topic N Categories	Topic N Gender
Topic 3	Women	0.15	women, woman, men, baby, pregnant, girls, male, doctor, health, birth	Content - Pregnancy	Women
	Men	-0.14			
Topic 10	Women	0.10	energy, body, feel, mind, space, yoga, love, beautiful, feeling, meditation	Content - Yoga	Women
	Men	-0.12			
Topic 49	Women	-0.21	game, know, think, team, going, mean, play, year, one, good	Content - Sports	Men
	Men	0.17			
Topic 71	Women	0.14	christmas, sex, girl, hair, love, get, date, girls, let, wear	Content - Dating	Women
	Men	-0.14			
Topic 54	Women	–	get, like, know, right, people, going, podcast, make, want, one	Discourse	Men
	Men	0.12			
Topic 60	Women	-0.27	going, know, think, get, got, one, really, good, well, yeah	Discourse	Men
	Men	0.20			
Topic 62	Women	0.33	like, know, really, going, people, want, think, get, things, life	Discourse	Women
	Men	-0.28			

Men: $s =$ *And I was* ***going****, hey, it's cold outside...*

Women: $s' =$ *And I was* ***like****, hey, it's cold outside...*

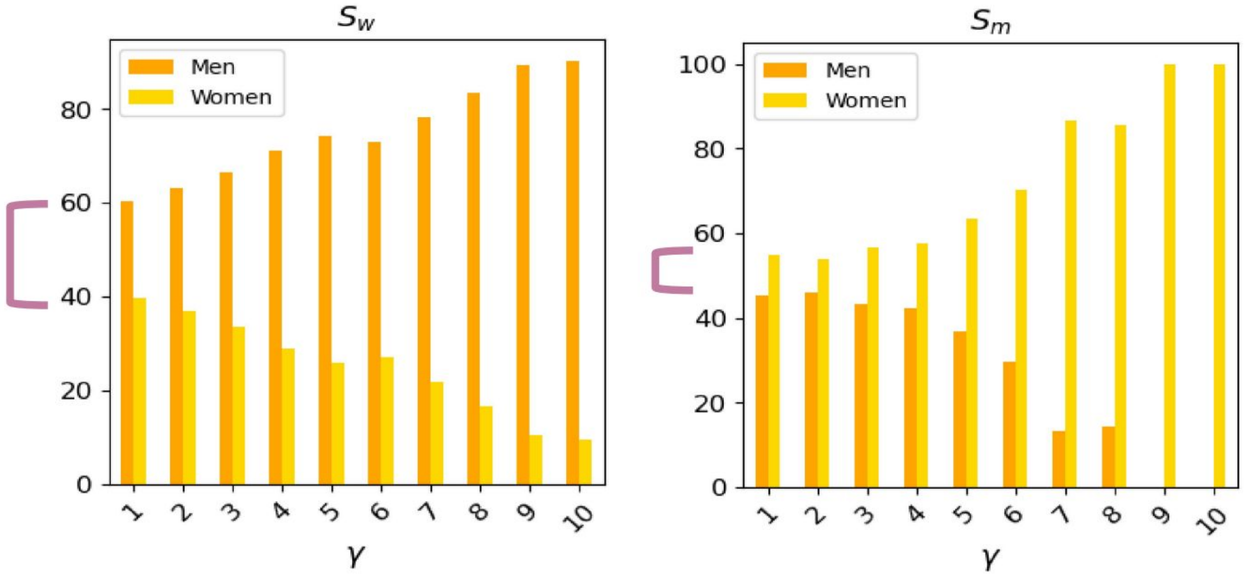
Is this difference reflected in LLM embeddings?



Men: $s =$ *And I was* ***going****, hey, it's cold outside...*

Women: $s' =$ *And I was* ***like****, hey, it's cold outside...*

Yes. Women have a less stable/robust embedding representation than men.



Men: $s =$ *And I was* ***going****, hey, it's cold outside...*

Women: $s' =$ *And I was* ***like****, hey, it's cold outside...*

The use of these masculine discourse terms is associated with economic rewards.

Topic N	Topic M	r	Topic N Word List	Topic N Categories	Topic M Word List	Topic M Categories
Topic 11	Topic 54	0.11	data, new, technology, public, bill, theory, science, system, security, article	Content - Technology/ Political $$$	get, like, know, right, people, going, podcast, make, want, one	Discourse (Men)
	Topic 62	-0.20			like, know, really, going, people, want, think, get, things, life	Discourse (Women)
Topic 12	Topic 54	0.24	business, money, company, market, buy, right, million, companies, pay, sell	Content - Business $$$	get, like, know, right, people, going, podcast, make, want, one	Discourse (Men)
Topic 79	Topic 60	0.18	game, games, play, playing, like, played, nintendo, video, fun, switch	Content - Video Games	going, know, think, get, got, one, re-ally, good, well, yeah	Discourse (Men)
	Topic 62	-0.13			like, know, really, going, people, want, think, get, things, life	Discourse (Women)

Men: $s =$ *And I was going, hey, it's cold outside...*

Women: $s' =$ *And I was like, hey, it's cold outside...*

The use of these masculine discourse terms is associated with economic rewards.

This is an opportunity to build systems that actually WORK for female speakers – there's an untapped market out there!

A quick plug for our recent work →

Maria Teleki · You
CS PhD Student @ TAMU | Speech AI/RecSys
1d ·

In our new work, 🎶 CHOIR: Collaborative Harmonization fOr Inference Robustness, we show that different LLM personas often get different benchmark questions right! CHOIR leverages this diversity to boost performance across benchmarks. 📊

You always hear about the "bias-accuracy tradeoff," meaning that ⬇️ model bias, ⬇️ system performance, so ⬇️ $$$ for a company. So much of the conversation around bias and diversity has focused on how to incentivize companies to debias their models (e.g., through new legislation).

🌟 To me, this work is super exciting because we take a totally different perspective: we show that ⬆️ diverse perspectives, ⬆️ system performance, so ⬆️ $$$ for a company! With this work, we argue that <<< 🚨 diverse perspectives are absolutely necessary >>> from an economic standpoint.

📄 https://lnkd.in/gr_Mmvsy
⚽ Xiangjue Dong (1st author), Cong ("Nicole") Wang, Millennium Bismay, and James Caverlee

We've also got some upcoming work on social-media-as-a-signal, stay tuned!

Collaborators

James Caverlee
(my advisor)

Xiangjue Dong

Haoran Liu

Sai Tejas
Janjur

Thomas
Docog

Oliver Grabner

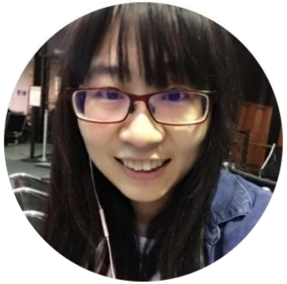
Soohwan Kim

Stephanie
Birkelbach

Rohan
Chaudhury

Ketan Verma

Chengkai Liu



Yin Zhang

+ Jason Kim, Lingfeng Shi, Cong Wang, and shoutout to work-in-progress collaborators too :)

Conversational AI

MARIA TELEKI

Will post slides after the talk!

Texas A&M University
Department of Computer Science & Engineering