GigaSpeech 2: An Evolving, Large-Scale and Multi-domain ASR Corpus for Low-Resource Languages with Automated Crawling, Transcription and Refinement

Yifan Yang[1], Zheshu Song[1], Jianheng Zhuo[1]
Mingyu Cui[4], Jinpeng Li[3], Bo Yang[2], Yexing Du[2,6], Ziyang Ma[1],
Xunying Liu[4], Ziyuan Wang[7], Ke Li[8], Shuai Fan[9], Kai Yu[1,9]
Wei-Qiang Zhang[3,11], Guoguo Chen[10,11], Xie Chen[1,5,11]

[1]X-LANCE Lab, SJTU [2]PCL [3]THU [4]CUHK [5]SII [6]HIT
[7]Birch AI [8]Dataocean AI [9]AISpeech Ltd [10]Seasalt AI Inc [11]SpeechColab

June 26, 2025

Scaling is Shown Promising in Speech

Leverage large-scale data matters

- ASR: MMS, USM, Whisper, Canary, Parakeet, Dolphin
- TTS: BaseTTS, Llasa, MaskGCT, F5-TTS

Leverage in-the-wild data matters

- Abundant & Readily Collectable
- Gap: Research vs. Industry

Methods

- Semi-Supervised Learning: Pseudo-Labeling (PL), Iterative Pseudo-Labeling (IPL), Noisy Student Training (NST)
- Self-Supervised Learning: HuBERT, WavLM, data2vec, data2vec 2.0, BEST-RQ

Scaling is Rarely Done for Low-Resource Languages

Southeast Asia languages: Thai (th), Indonesian (id), Vietnamese (vi)

Dataset	Language	# Hours (h)	Domain	Speech Type	Labeled	Label Type
Common Voice	th id vi	172.0 28.0 6.0	Open domain	Read	Yes	Manual
FLEURS	th id vi	13.3 12.6 13.3	Wikipedia	Read	Yes	Manual
VoxLingua107	th id vi	61.0 40.0 64.0	YouTube	Spontaneous	No	-
CMU Wilderness	th id vi	15.6 70.9 9.2	Religion	Read	Yes	Manual
BABEL	vi	87.1	Conversation	Spontaneous	Yes	Manual
VietMed	vi	16.0	Medical	Spontaneous	Yes	Manual
Thai Dialect Corpus	th	840.0	Open domain	Read	Yes	Manual
TITML-IDN	id	14.5	News	Read	Yes	Manual
MEDISCO	id	10.0	Medical	Read	Yes	Manual
YODAS manual	th id vi	497.1 1420.1 779.9	YouTube	Spontaneous	Yes	Manual
YODAS automatic	th id vi	1.9 8463.6 9203.1	YouTube	Spontaneous	Yes	Pseudo
GigaSpeech 2 raw	th id vi	12901.8 8112.9 7324.0	YouTube	Spontaneous	Yes	Pseudo
GigaSpeech 2 refined	th id vi	10262.0 5714.0 6039.0	YouTube	Spontaneous	Yes	Pseudo

A Retrospective of ASR Datasets

1993
Switchboard-1
Human-recorded
Human-annotated
English
260h

2004
Fisher
Human-recorded
Human-annotated
English
1,698h

2015
LibriSpeech
Audiobook
Auto-aligned
English
960h

2021
GigaSpeech
Podcast & YouTube & Audiobook
Auto-aligned
English
10,000h

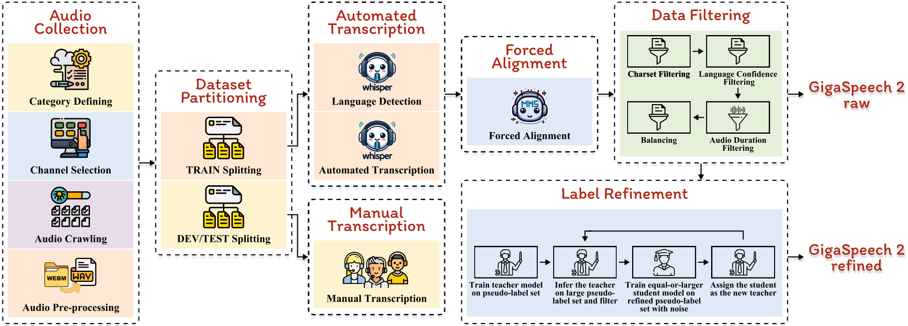
2022
WenetSpeech
Podcast & YouTube
Auto-aligned
Chinese
10,000h

2023
Libriheavy
Audiobook
Auto-aligned
English
50,000h

2024
YODAS
YouTube
User-uploaded & Auto-generated
Multilingual
422,245h

New Paradigm for Constructing Large-Scale ASR Datasets

- In-the-wild data oriented
- Audio-only, free of scarce paired data
- Automated pipeline



GigaSpeech 2: Key Contributions

Large-scale, Multi-domain, and Multilingual Spontaneous ASR Corpus

- GigaSpeech 2 raw: 30kh, covering Thai, Indonesian, and Vietnamese.
- GigaSpeech 2 refined: Thai (10kh), Indonesian (6kh), and Vietnamese (6kh).

Automated ASR Corpus Construction Pipeline

Audio-only, without reliance on labeled data.

Modified NST Method to Refine Flawed Pseudo Labels Iteratively

Challenging and Realistic Manual Evaluation Sets

Covers spontaneous speech across multiple topics and content formats.

Strong Empirical Validation of GigaSpeech 2

- Multiple test sets: GigaSpeech 2, Common Voice, and FLEURS
- Outperforms Whisper Large-v3 and commercial APIs (Azure, Google)

Dataset Construction: GigaSpeech 2 raw (1/3)

Audio Collection

- Select YouTube channels
- Multiple topics: Agriculture, Art, Business, Climate, Culture, Economics, Education, Entertainment, Health, History, Literature, Music, Politics, Relationships, Shopping, Society, Sport, Technology, Travel
- Various content formats: Audiobook, Commentary, Lecture, Monologue, Movie, News, Talk, Vlog

Creating TRAIN/DEV/TEST Splits

- Ensuring no speaker overlap between the splits.
- DEV and TEST sets each contain 10 hours, manually transcribed by professionals.

Dataset Construction: GigaSpeech 2 raw (2/3)

Transcription with Whisper

- Whisper Large-v3 model
- Language detection: 30-second segment from the middle

Forced Alignment with MMS

- Whisper can generate timestamps, but not precise enough.
- CTC alignment model from MMS: robust to noise, GPU efficient, effective for long sequences.

Text Normalization

- Normalization Form Compatibility Composition (NFKC)
- Uppercase all characters
- Remove punctuation
- Map Arabic numerals to corresponding words

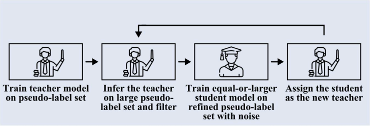
Dataset Construction: GigaSpeech 2 raw (3/3)

Multi-dimensional Filtering

- **Charset Filtering:** Keep segments with characters only from the target language permitted charset.
- **Language Confidence Filtering:** Use fastText LID model to filter by confidence score.
- **Audio Duration Filtering:** Filter segments based on min/max duration thresholds.
- **Balancing:** Control the duplication of transcripts caused by channel-specific content.

Dataset Construction: GigaSpeech 2 refined (1/10)

Modified NST method for iterative label refinement

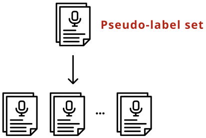


Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (2/10)

Modified NST method for iterative label refinement



Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (3/10)

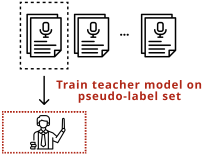
Modified NST method for iterative label refinement

Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\boxed{\mathcal{R} \leftarrow \mathcal{P}_1;}$
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (4/10)

Modified NST method for iterative label refinement

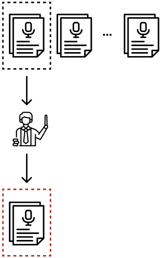


Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
    $\mathcal{R} \leftarrow \varnothing$;
    if $i == 1$ **then**
        `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
        $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
    else
        for $j \leftarrow 1$ **to** i **do**
            `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
            $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
            $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
        end
    end
    Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (5/10)

Modified NST method for iterative label refinement



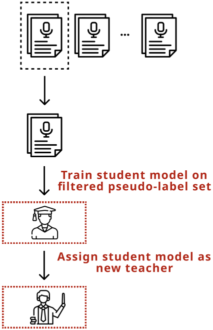
Filter train set by teacher model with CER

Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    // Filter $\mathcal{P}_i$ by teacher model $\mathcal{M}_i$ with CER $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      // Relabel $\mathcal{P}_j$ by teacher model $\mathcal{M}_i$ and filter with CER $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (6/10)

Modified NST method for iterative label refinement

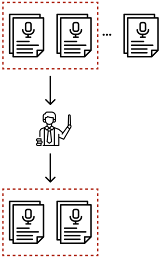


Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (7/10)

Modified NST method for iterative label refinement



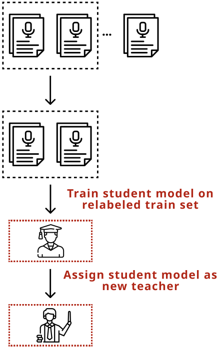
Relabel train set by teacher model and filter with CER

Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
    $\mathcal{R} \leftarrow \varnothing$;
    if $i == 1$ **then**
        // Filter $\mathcal{P}_i$ by teacher model $\mathcal{M}_i$ with CER $\leq \tau$
        $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
    else
        for $j \leftarrow 1$ **to** i **do**
            // Relabel $\mathcal{P}_j$ by teacher model $\mathcal{M}_i$ and filter with CER $\leq \tau$
            $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
            $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
        end
    end
    Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (8/10)

Modified NST method for iterative label refinement

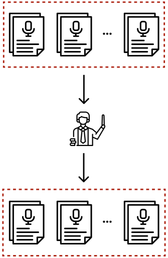


Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
    $\mathcal{R} \leftarrow \varnothing$;
    if $i == 1$ **then**
        // Filter $\mathcal{P}_i$ by teacher model $\mathcal{M}_i$ with CER $\leq \tau$
        $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
    else
        for $j \leftarrow 1$ **to** i **do**
            // Relabel $\mathcal{P}_j$ by teacher model $\mathcal{M}_i$ and filter with CER $\leq \tau$
            $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
            $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
        end
    end
    Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (9/10)

Modified NST method for iterative label refinement



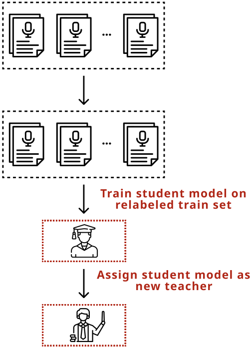
Relabel train set by teacher model and filter with CER

Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
    $\mathcal{R} \leftarrow \varnothing$;
    if $i == 1$ **then**
        `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
        $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
    else
        for $j \leftarrow 1$ **to** i **do**
            `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
            $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
            $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
        end
    end
    Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

Dataset Construction: GigaSpeech 2 refined (10/10)

Modified NST method for iterative label refinement



Algorithm 1: Iterative Label Refinement

Input: Pseudo-label set $\mathcal{P}$, Number of iterations n, Threshold τ
Output: Refined-label set $\mathcal{R}$
Divide $\mathcal{P}$ into n splits $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$;
$\mathcal{R} \leftarrow \mathcal{P}_1$;
Train teacher model $\mathcal{M}_1$ on $\mathcal{R}$ with noise;
for $i \leftarrow 1$ **to** n **do**
  $\mathcal{R} \leftarrow \varnothing$;
  if $i == 1$ **then**
    `// Filter` $\mathcal{P}_i$ `by teacher model` $\mathcal{M}_i$ `with CER` $\leq \tau$
    $\mathcal{R} \leftarrow \{(x, y) \in \mathcal{P}_i \mid \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
  else
    for $j \leftarrow 1$ **to** i **do**
      `// Relabel` $\mathcal{P}_j$ `by teacher model` $\mathcal{M}_i$ `and filter with CER` $\leq \tau$
      $\mathcal{R}_{tmp} \leftarrow \{(x, \mathcal{M}_i(x)) \mid (x, y) \in \mathcal{P}_j, \mathrm{CER}(y, \mathcal{M}_i(x)) \leq \tau\}$;
      $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{tmp}$;
    end
  end
  Train equal-or-larger student model $\mathcal{M}_{i+1}$ on $\mathcal{R}$ with noise and assign as new teacher;
end
return $\mathcal{R}$;

ASR Model Training on GigaSpeech 2

Our modified NST is effective

- Consistent improvements in the WER performance on four evaluation sets until the final iteration.

Thai achieves the lowest CER

- WER relative reductions of 13.92%, 17.48%, 53.27%, and 26.45% respectively (Thai, Iteration 4 vs. Iteration 1).

NST Iter	**#Hours (h)**	**#Vocab**	**#Params (M)**	**CER / WER**			
				GigaSpeech 2		Common Voice	FLEURS
				DEV	TEST	TEST	TEST
Thai							
1	4378	500	65.5	12.14	15.10	8.88	14.33
2	3497	500	65.5	$10.97_{-9.6\%}$	$13.15_{-12.9\%}$	$6.99_{-21.3\%}$	$11.93_{-16.7\%}$
3	7219	2000	68.6	$10.50_{-4.3\%}$	$12.46_{-5.2\%}$	$4.61_{-34.0\%}$	$10.94_{-8.3\%}$
4	10262	2000	151.9	$10.45_{-0.5\%}$	$12.46_{-0.0\%}$	$4.15_{-10.0\%}$	$10.54_{-3.7\%}$
Indonesian							
1	5765	2000	68.6	16.68	15.99	19.82	16.29
2	4534	2000	68.6	$15.60_{-6.5\%}$	$15.23_{-4.8\%}$	$15.83_{-20.1\%}$	$14.30_{-12.2\%}$
3	5714	2000	151.9	$14.58_{-6.5\%}$	$14.92_{-2.0\%}$	$13.83_{-12.6\%}$	$13.77_{-3.7\%}$
Vietnamese							
1	2351	2000	68.6	16.08	16.95	24.63	17.86
2	1764	2000	68.6	$15.08_{-6.2\%}$	$14.72_{-13.2\%}$	$18.81_{-23.6\%}$	$13.50_{-24.4\%}$
3	6039	2000	151.9	$14.09_{-6.6\%}$	$12.83_{-12.8\%}$	$14.43_{-23.3\%}$	$11.59_{-14.1\%}$

Comparison to Existing ASR Systems (1/3)

Thai outperforms all baselines

- Outperform commercial services from Azure and Google.
- Outperform Whisper large-v3 by WER relative reductions of 39.04%, 31.06%, and 8.74% (Thai, Row 7 vs. Row 1).
- Nearly 10% parameters compared to Whisper large-v3 (151.9 M vs. 1542 M).

Model	#Params (M)	CER / WER GigaSpeech 2	Common Voice	FLEURS
Thai				
Whisper large-v3	1542	20.44	6.02	11.55
Whisper large-v2	1541	22.47	8.79	15.50
Whisper base	72	46.47	32.59	42.28
MMS L1107	964	31.75	14.49	23.07
Azure Speech CLI 1.37.0†	-	17.25	10.20	13.35
Google USM Chirp v2†	-	49.70	14.75	63.35
GigaSpeech 2 (proposed)	151.9	**12.46**	**4.15**	**10.54**
Indonesian				
Whisper large-v3	1542	20.03	7.43	7.85
Whisper large-v2	1541	21.44	8.93	8.95
Whisper base	72	39.37	34.70	33.76
MMS L1107	964	35.27	20.72	24.49
Azure Speech CLI 1.37.0†	-	18.07	10.33	11.18
Google USM Chirp v2†	-	19.63	9.70	**7.23**
GigaSpeech 2 (proposed)	151.9	**14.92**	13.83	13.77
+ Common Voice + FLEURS	151.9	14.95	**7.33**	12.74
Vietnamese				
Whisper large-v3	1542	17.94	13.74	**8.59**
Whisper large-v2	1541	18.74	18.00	10.26
Whisper base	72	39.88	44.07	40.41
MMS L1107	964	46.62	43.88	55.35
Azure Speech CLI 1.37.0†	-	**11.86**	**10.21**	11.88
Google USM Chirp v2†	-	13.28	12.46	11.75
GigaSpeech 2 (proposed)	151.9	12.83	14.43	11.59
+ Common Voice + FLEURS	151.9	12.39	11.47	9.94

Comparison to Existing ASR Systems (2/3)

Indonesian and Vietnamese achieve competitive performance

- Indonesian outperforms all baseline models on the GigaSpeech 2 test set.
- Indonesian outperforms Whisper large-v3 by WER relative reduction of 25.51% (Indonesian, Row 7 vs. Row 1, GigaSpeech 2 TEST).
- Vietnamese outperforms Whisper large-v3 by WER relative reduction of 28.48% (Vietnamese, Row 7 vs. Row 1, GigaSpeech 2 TEST).
- Nearly 10% parameters compared to Whisper large-v3 (151.9 M vs. 1542 M).

Model	#Params (M)	CER / WER GigaSpeech 2	Common Voice	FLEURS
Thai				
Whisper large-v3	1542	20.44	6.02	11.55
Whisper large-v2	1541	22.47	8.79	15.50
Whisper base	72	46.47	32.59	42.28
MMS L1107	964	31.75	14.49	23.07
Azure Speech CLI 1.37.0†	-	17.25	10.20	13.35
Google USM Chirp v2†	-	49.70	14.75	63.35
GigaSpeech 2 (proposed)	151.9	**12.46**	**4.15**	**10.54**
Indonesian				
Whisper large-v3	1542	20.03	7.43	7.85
Whisper large-v2	1541	21.44	8.93	8.95
Whisper base	72	39.37	34.70	33.76
MMS L1107	964	35.27	20.72	24.49
Azure Speech CLI 1.37.0†	-	18.07	10.33	11.18
Google USM Chirp v2†	-	19.63	9.70	**7.23**
GigaSpeech 2 (proposed)	151.9	**14.92**	13.83	13.77
+ Common Voice + FLEURS	151.9	14.95	**7.33**	12.74
Vietnamese				
Whisper large-v3	1542	17.94	13.74	**8.59**
Whisper large-v2	1541	18.74	18.00	10.26
Whisper base	72	39.88	44.07	40.41
MMS L1107	964	46.62	43.88	55.35
Azure Speech CLI 1.37.0†	-	**11.86**	**10.21**	11.88
Google USM Chirp v2†	-	13.28	12.46	11.75
GigaSpeech 2 (proposed)	151.9	12.83	14.43	11.59
+ Common Voice + FLEURS	151.9	12.39	11.47	9.94

Comparison to Existing ASR Systems (3/3)

Indonesian and Vietnamese demonstrates degraded performance compared to commercial ASR systems on the Common Voice and FLEURS test sets

- Be attributed to domain mismatch.
- Performance leap after adding Common Voice and FLEURS training data into GigaSpeech 2 (Indonesian & Vietnamese, Row 7 vs. Row 8).

Model	#Params (M)	CER / WER GigaSpeech 2	Common Voice	FLEURS
Thai				
Whisper large-v3	1542	20.44	6.02	11.55
Whisper large-v2	1541	22.47	8.79	15.50
Whisper base	72	46.47	32.59	42.28
MMS L1107	964	31.75	14.49	23.07
Azure Speech CLI 1.37.0†	-	17.25	10.20	13.35
Google USM Chirp v2†	-	49.70	14.75	63.35
GigaSpeech 2 (proposed)	151.9	**12.46**	**4.15**	**10.54**
Indonesian				
Whisper large-v3	1542	20.03	7.43	7.85
Whisper large-v2	1541	21.44	8.93	8.95
Whisper base	72	39.37	34.70	33.76
MMS L1107	964	35.27	20.72	24.49
Azure Speech CLI 1.37.0†	-	18.07	10.33	11.18
Google USM Chirp v2†	-	19.63	9.70	**7.23**
GigaSpeech 2 (proposed)	151.9	**14.92**	13.83	13.77
+ Common Voice + FLEURS	151.9	14.95	**7.33**	12.74
Vietnamese				
Whisper large-v3	1542	17.94	13.74	**8.59**
Whisper large-v2	1541	18.74	18.00	10.26
Whisper base	72	39.88	44.07	40.41
MMS L1107	964	46.62	43.88	55.35
Azure Speech CLI 1.37.0†	-	**11.86**	**10.21**	11.88
Google USM Chirp v2†	-	13.28	12.46	11.75
GigaSpeech 2 (proposed)	151.9	12.83	14.43	11.59
+ Common Voice + FLEURS	151.9	12.39	11.47	9.94

Comparison to the YODAS Corpus

GigaSpeech 2 refined yield significantly better results YODAS in the GigaSpeech 2 test set for all three languages

- For Thai and Vietnamese, GigaSpeech 2 refined consistently outperform YODAS manual across all evaluation sets.
- YODAS manual overfits due to simplistic filtering rules, leading to inconsistent performance in Indonesian.

Adding YODAS automatic tends to degrade performance

- Due to inherent noise and errors in the automatic subtitles.

Training Set	#Params (M)	CER / WER GigaSpeech 2	Common Voice	FLEURS
Thai				
YODAS manual	68.6	27.34	10.71	14.19
YODAS manual	151.9	28.76	10.96	16.11
GigaSpeech 2 refined	151.9	**12.46**	**4.15**	**10.54**
Indonesian				
YODAS manual	68.6	25.77	**10.82**	14.63
YODAS manual + automatic	68.8	41.11	15.41	47.26
YODAS manual	151.9	25.11	11.05	**12.67**
GigaSpeech 2 refined	151.9	**14.92**	13.83	13.77
Vietnamese				
YODAS manual	68.6	40.35	31.07	25.68
YODAS manual + automatic	68.6	71.91	25.73	61.38
YODAS manual	151.9	40.71	32.58	29.32
GigaSpeech 2 refined	151.9	**12.83**	**14.43**	**11.59**

Training ASR Models within ESPNet and Icefall on GigaSpeech 2

Toolkit	Model	#Params (M)	CER / WER		
			th	id	vi
Icefall	Zipformer/Stateless Pruned RNN-T	151.9	12.46	14.92	12.83
ESPnet	Conformer/Transformer CTC/AED	111.8	13.70	15.50	14.60

Icefall

Zipformer Pruned RNN-T

- Zipformer-Large encoder
- Stateless decoder
- Pruned RNN-T loss
- 2000-class BPE

ESPnet

Conformer CTC/AED

- Conformer-L encoder
- Transformer decoder
- CTC & AED loss
- 2000-class BPE

Resource link

Github Repository for Automated Pipeline

https://github.com/SpeechColab/GigaSpeech2

Download GigaSpeech 2 on Hugging Face

https://huggingface.co/datasets/speechcolab/gigaspeech2

Preprint Paper Link

https://arxiv.org/pdf/2406.11546

Thank You

If you have any questions, feel free to contact me.

Email: yifanyeung@sjtu.edu.cn