

# USM-SCD: MULTILINGUAL SPEAKER CHANGE DETECTION BASED ON LARGE PRETRAINED FOUNDATION MODELS

Guanlong Zhao, Yongqiang Wang, Jason Pelecanos  
Yu Zhang, Hank Liao, Yiling Huang, Han Lu, Quan Wang

Google LLC, USA

{guanlongzhao, yqw, pelecanos, ngyuzh, hankliao, yilinghuang, luha, quanw}@google.com

## ABSTRACT

We introduce a multilingual speaker change detection model (USM-SCD) that can simultaneously detect speaker turns and perform ASR for 96 languages. This model is adapted from a speech foundation model trained on a large quantity of supervised and unsupervised data, demonstrating the utility of fine-tuning from a large generic foundation model for a downstream task. We analyze the performance of this multilingual speaker change detection model through a series of ablation studies. We show that the USM-SCD model can achieve more than 75% average speaker change detection F1 score across a test set that consists of data from 96 languages. On American English, the USM-SCD model can achieve an 85.8% speaker change detection F1 score across various public and internal test sets, beating the previous monolingual baseline model by 21% relative. We also show that we only need to fine-tune one-quarter of the trainable model parameters to achieve the best model performance. The USM-SCD model exhibits state-of-the-art ASR quality compared with a strong public ASR baseline, making it suitable to handle both tasks with negligible additional computational cost.

**Index Terms**— Speaker change detection, foundation model

## 1. INTRODUCTION

Speaker change detection (SCD) [1] is the process of identifying the speaker turn points in a multi-speaker audio stream. SCD has broad applications in enhancing speaker diarization accuracy [2, 3], improving Automatic Speech Recognition (ASR) quality [4], generating line breaks in captions to boost readability and accessibility [5], and augmenting textual prompts for multi-modal large language models (LLMs) [6].

Conventionally, SCD is achieved by using a neural network to map acoustic features or speaker embeddings [7–9] to a frame or segment level yes/no speaker change prediction. The neural network is generally trained by minimizing the binary cross entropy loss between the ground-truth SCD labels and the predictions. These conventional approaches have various limitations. First, they require accurate timing information of the speaker change point, which is difficult to obtain since marking speaker change timestamps is a highly subjective process for human annotators. Second, the methods that use purely acoustic information ignore rich semantic information in the audio. Third, the methods that use speaker embeddings utilize sensitive biometric information that can be exploited for unintended purposes and are sub-optimal from a privacy point of view [10].

A few recent studies [2, 11, 12] explore using ASR-based approaches to detect word-level speaker changes to mitigate the aforementioned issues with conventional models. Xia *et al.* [2] propose an

SCD model using a Transformer-Transducer (T-T). Specifically, they augment the text transcription of the spoken utterance with a special speaker turn token `<st>`, and then train the model to output both regular text tokens and the special speaker turn token. This model does not need accurate timestamps for training since the T-T model is trained in a seq2seq fashion and does not need forced-alignment to provide training targets. The model also utilizes both acoustic and linguistic information in the input audio. As a follow up of that work, in [11] we propose a training loss that penalizes speaker change false acceptance and false rejection errors in the N-best hypotheses to further enhance performance. Wu *et al.* [12] add an additional SCD module on top of an existing T-T ASR network to optimize the SCD and ASR tasks separately.

Recent advances in self-supervised learning have ushered in a new era for speech tasks. Large pretrained foundation models [13] have led to significant performance improvement in various downstream speech tasks including emotion recognition [14], language identification [15], voice activity detection [16], and mispronunciation detection [17]. In this work we take advantage of the recent Google Universal Speech Model (USM) [18] framework to build an SCD model that is capable of recognizing speaker changes in 96 languages. In addition, the performance of prior ASR-based models is limited by the quantity of supervised SCD data available for individual languages, leading to lower performance. We explore the benefit of using a large quantity of unsupervised and supervised multilingual ASR data for model pretraining. The major contributions of this paper include (1) a 96-language SCD model that significantly outperforms the previous monolingual baseline; (2) detailed ablation studies of the proposed multilingual SCD model.

## 2. METHOD

First, we build a pretrained model as the foundation model. We then fine-tune the foundation model with data annotated with speaker changes.

### 2.1. Backbone model

At a high level, the backbone model architecture used in this work consists of a Conformer encoder [19] and a Connectionist Temporal Classification (CTC) [20] decoder. The inputs are mel-spectra features and a one-hot vector representing the language of the utterance.

We pass the input features through mean variance normalization, SpecAugment [21] (only for training), and multiple 2D-convolution layers (denoted as the *feature encoder*) to reduce the input frame rate, similar to the setup in wav2vec 2.0 [22]. We then append the features with a one-hot language embedding. The concatenated features are then projected by a linear input projection layer to match

the dimension of the Conformer encoder, which takes the projection layer outputs as its inputs. The Conformer encoder is trained with chunk-wise attention [18]. The output of the Conformer encoder is passed to a linear projection layer, outputting logits that correspond to WordPiece tokens. The model is trained with the CTC loss. We do not use the RNN-T paradigm [23] in this work due to its slow training speed as a result of its auto-regressive nature, which is especially prevalent when training large models with billions of parameters.

## 2.2. Pretraining

There are various pretraining techniques. In this work, we explore both supervised and unsupervised pretraining methods.

### 2.2.1. BEST-RQ pretraining

We select BEST-RQ [24] as the unsupervised method to pretrain our networks. BEST-RQ provides a simple framework with a small number of hyperparameters for unsupervised training on large-scale unlabeled audio data. BEST-RQ applies a random-projection quantizer to map speech signals to discrete labels to enable BERT-style pretraining for ASR encoders. The quantizer randomly initializes a matrix and a codebook, and uses the matrix to project the input speech signals and the codebook to find the nearest vector, where the index of the vector serves as the label. The pretraining process masks the speech signals and feeds them to the ASR encoder that learns to predict labels of the masked segment. The random projection performs dimension reduction for the speech signals while the random codebook provides an approximated discrete representation of the data distribution. Both the randomly initialized matrix and codebook are fixed during the pretraining process. In this study, the encoder in the BEST-RQ system employs the same model architecture as the Conformer encoder described in Sec. 2.1.

### 2.2.2. ASR pretraining

For supervised pretraining, we initialize the Conformer encoder’s weights from the BEST-RQ model’s encoder and fine-tune it on ASR data to predict text from audio.

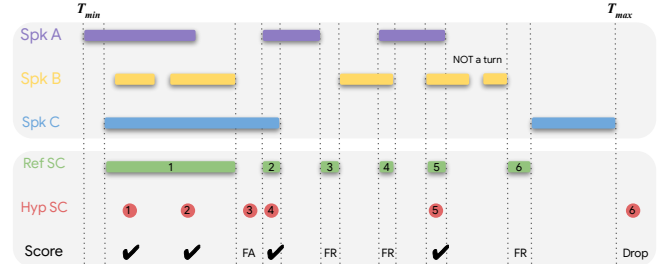
## 2.3. SCD fine-tuning

For the SCD task, we fine-tune the pretrained model with speaker change data, and we refer to this type of model as **USM-SCD**.

We warm start the backbone model’s Conformer encoder from a pretrained model’s encoder. The decoder projection layer is always randomly initialized. The training targets are WordPiece tokens augmented with speaker change annotations. To create training targets, we add a special speaker change token  $\langle st \rangle$  between two different speakers’ transcripts (e.g. “hello how are you  $\langle st \rangle$  I am good  $\langle st \rangle$ ”) to model speaker changes during training. Compared with audio-only SCD models [8], this model may more directly utilize the language semantics as a signal for speaker segmentation. For inference, we perform an ASR decoding with the SCD model, and identify the speaker change tokens. We use the timestamps of the predicted speaker turn tokens in the evaluation.

## 2.4. Speaker change token posterior scaling

The speaker change tokens are relatively scarce in the training data. To encourage the model to output speaker change tokens, we can apply a scaling factor to the posterior probability of the speaker change token  $p(\langle st \rangle | \mathbf{X})$  during decoding, where  $\mathbf{X}$  is the model input. Assuming *greedy* decoding (see section 3.2 of [20]) during inference,



**Fig. 1:** Illustration of the SCD scoring mechanism for computing the precision, recall, and F1. “Spk A-C” stands for speaker annotations on a conversational utterance. “Ref SC” is the reference speaker change intervals. “Hyp SC” is the predicted speaker change. “Score” shows the scoring decision of each prediction and reference.

this can be achieved by multiplying  $p(\langle st \rangle | \mathbf{X})$  with a constant factor  $\lambda > 1$ , i.e.,  $p'(\langle st \rangle | \mathbf{X}) = \lambda \cdot p(\langle st \rangle | \mathbf{X})$ . Effectively this increases the posterior probability of the  $\langle st \rangle$  token. *Greedy* decoding simplifies the process since we do not need to redistribute the rest of the probability mass as a result of the scaling. In practice, we operate on the log posterior probability rather than on the raw posterior probability to avoid numerical issues, hence we have  $\log(p'(\langle st \rangle | \mathbf{X})) = \log(\lambda) + \log(p(\langle st \rangle | \mathbf{X}))$ .

## 2.5. SCD evaluation metrics

For SCD evaluation, we compute precision (percentage of model predictions that are true speaker changes), recall (percentage of ground-truth speaker changes that are correctly predicted by the model), and F1 score (the harmonic mean of the precision and recall). We treat the F1 score as a more comprehensive quality metric than the precision/recall rate alone. To compute these metrics, we align predicted and ground-truth speaker changes based on their timestamps, i.e., correct predictions should overlap with the ground-truth labels. Please refer to Fig. 1 for an example. For a detailed description of these metrics, please see Sec. 3 of [11].

# 3. EXPERIMENTAL SETUP

## 3.1. Data

We use various supervised and unsupervised short/long-form data across model training and evaluation. All internal datasets are collected according to Google’s Privacy Principles [25] and abide by Google AI Principles [26].

### 3.1.1. Training

**YT-56-U:** This dataset is built by first randomly collecting three million hours of audio from “speech-heavy” user-uploaded YouTube videos, filtered by user-provided language tags. The three million hours of audio is then further segmented by a Voice Activity Detection (VAD) model and non-speech segments are removed. This yields approximately one million hours of unlabeled audio data. Later, we use a language identification model to select data that corresponds to 56 languages from that unlabelled audio data. *We use this dataset to pretrain the BEST-RQ model.*

**VS-SUP:** We use a Voice Search dataset consisting of 85 language locales to pretrain the ASR model. There are a total of 1.2 billion short utterances (average duration 4 seconds) from Voice Search traffic. The data is anonymized and human transcribed. *No speaker change information is available for this dataset. We use this*

**Table 1:** Statistics of additional internal and public En-US test sets.

Testset	Domain	Dur. (h)	Average	
			Turns/min	Duration/Utt. (min)
AMI [28]	Meeting	9.1	10	34
Callhome [29]	Telephone	1.7	19	5
DIHARD1 [30]	Mixed	16.2	12	9
Fisher [31]	Telephone	28.7	13	10
ICSI [32]	Meeting	2.8	13	55
Inbound	Telephone	21.0	9	5
Outbound	Telephone	45.6	13	6

dataset for ASR pretraining because short-form supervised ASR training data is significantly larger in volume than long-form data with speaker change labels.

**YT-SUP:** This is a dataset with audio from YouTube videos that has text transcripts and speaker change labels from 96 languages. We group consecutive segments into a longer unit similar to [27]. The maximum sequence length for training is 30 seconds. The total quantity of training data is 108k hours, ranging from three hours (Paraguayan Guarani) to 4k hours (Brazilian Portuguese) across locales. We use this dataset to fine-tune the USM-SCD model.

### 3.1.2. Evaluation

**YT-96-Eval:** For all languages, we have in total 1,400 hours of internal YouTube long-form evaluation data (no overlap with **YT-56-U** or **YT-SUP**) annotated with text transcriptions and speaker changes. On average, we have 15.2h (std: 4.5h) of evaluation data per language and 5 speaker changes per minute of audio in this test set.

**En-US-Eval:** For American English (En-US), we have additional internal and public test sets, see Table 1. For the first DIHARD challenge evaluation subset (DIHARD1), we remove all YouTube-derived utterances to avoid evaluating on utterances that might have appeared during training. For Fisher, we randomly sample a subset of 172 utterances for testing<sup>1</sup>. “Outbound” and “Inbound” are vendor-provided call center telephone conversations between call center operators and customers, initiated by the call center and by customers, respectively. “Outbound” and “Inbound” were previously used in [2, 3, 11].

### 3.2. Modeling details

We extract 128-dim log-mel filter-bank energies from a 32ms window with a 10ms frame shift as the raw input feature to the model. We use a WordPiece model that has a vocabulary size of 16,384.

The feature encoder contains two 2D-convolution layers of shape  $3 \times 3 \times 1 \times 128$  and  $3 \times 3 \times 128 \times 32$  (time  $\times$  frequency  $\times$  input-channel  $\times$  output-channel), respectively. The stride size of both convolution layers is 2 on both the time and frequency dimensions. The feature encoder increases the frame rate by 4-fold (i.e., down-sample the frames by 4-fold), from 10ms to 40ms, resulting in a 1,024-dim feature vector. The multi-headed self-attention in the Conformer layers has 8 attention heads. The chunk-wise attention in the Conformer encoder has an 8s context. The convolution kernel size is 5. We run experiments on a model with 1.84 billion parameters, where we have 32 Conformer layers and each layer has 1,536 dimensions.

We use the Adafactor optimizer [33] with a transformer learning rate schedule. For fine-tuning tasks, we optimize the encoder and decoder with separate optimizers and learning rate schedules given that the encoder alone has been pretrained. For the encoder, we use a peak learning rate  $3 \times 10^{-4}$  with 6k warm-up steps, while for the

**Table 2:** Overall system comparisons on **YT-96-Eval**. The *w/ SCD* systems are fine-tuned from the corresponding pretrained models with speaker change tokens in the training target; the *w/o SCD* system is trained to perform only ASR.

		BEST-RQ Pretrain w/ SCD	ASR Pretrain w/ SCD	ASR Pretrain w/o SCD	Whisper large-v2
WER	En-US	17.1	<b>12.6</b>	<b>12.6</b>	16.2
	21-lang.	21.1	<b>16.6</b>	<b>16.6</b>	30.1
	96-lang.	34.3	30.1	<b>28.8</b>	-
SCD	Precision	80.0	<b>82.4</b>	-	-
	Recall	<b>52.6</b>	51.9	-	-
	F1	63.5	<b>63.7</b>	-	-

decoder projection layer we use a peak learning rate  $5 \times 10^{-4}$  and 2k warm-up steps. Training was done with a global batch size of 4,096 on TPUs [34]. We monitor the training process on a held-out development set. For all models, we train them for around 40k steps. Empirically [35], fine-tuning from a well-trained foundation model only requires a small fraction of training steps compared with training from scratch. In this study, we observe that the model can converge to a reasonable state with as few as 5k training steps, which takes about 6.5 hours of training time with the aforementioned setup.

## 4. RESULTS

We compute the WER (for ASR) and SCD precision, recall, and F1 rates as quality metrics. For all evaluations, unless otherwise specified, we use greedy search and aggregate the evaluation data from all 96 languages (**YT-96-Eval**) to compute the final scores. For WER, we remove speaker change tokens from the scoring.

### 4.1. Overall system comparisons on YT-96-Eval

We first study the choice of the pretrained model. The results are summarized in the first two columns of Table 2. The SCD models fine-tuned from the two pretrained models yield comparable SCD F1 scores (0.3% relative difference), suggesting that they are comparable in terms of detecting speaker change events. The SCD model fine-tuned from the ASR model has significantly better WER (30.1 vs 34.3 across 96 languages; a 12.2% relative reduction), demonstrating the benefit of ASR-pretraining on the word-level SCD task.

Next, we study the trade off between ASR and SCD. We fine-tune from the ASR-pretrained checkpoint to construct *ASR Pretrain w/o SCD* that does not have the speaker change token in the training target, resulting in a WER of 28.8%. Therefore, with the proposed approach, adding the SCD capability to the ASR model would result in a 4.5% relative WER regression.

To provide additional context, we compare the WER of the USM-SCD model with a strong publicly available ASR model Whisper [36] (large-v2, 1.55B parameters) that was trained on more than 400k hours of transcribed ASR data. We select 21 top performing languages from Whisper (which achieve WER lower than 40% on **YT-96-Eval**), and the results are shown in the last column of Table 2. We observe that although adding the SCD capability to an ASR model hurts the WER, the resulting USM-SCD model still has a better ASR performance on YouTube data compared to Whisper.

### 4.2. Effect of sub-components to fine-tune

We now study which model parameters to fine-tune. For this experiment, we always fine-tune from the ASR pretrained model given the results in Sec. 4.1. Given that we are using a different data source (i.e., **YT-SUP**) for SCD training and we modify the training targets, we always fine-tune the *feature* encoder, input projection,

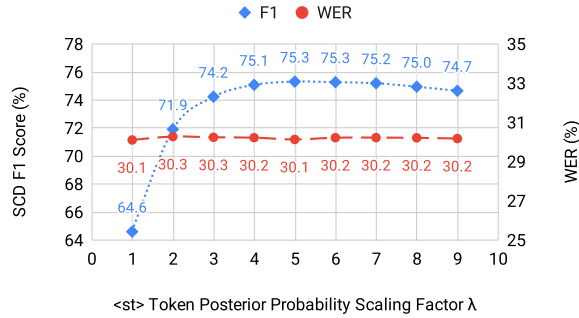
<sup>1</sup><https://github.com/google/speaker-id/blob/master/publications/ScdLoss/eval/fisher.txt>

**Table 4:** En-US results based on **En-US-Eval**. DIHARD1 and In/Outbound do not have ground-truth text transcripts. The last column shows the evaluation metrics computed by pooling all test sets together.

Metrics	System	AMI	CallHome	DIHARD1	Fisher	ICSI	Inbound	Outbound	Pooled data
WER	SCD loss	39.8	57.3	-	30.6	46.1	-	-	34.3
	USM SCD	25.7	44.2	-	18.4	31.5	-	-	<b>21.6</b>
Precision	SCD loss	79.4	82.0	78.8	82.6	77.8	72.8	75.1	77.6
	USM SCD	91.6	84.6	92.9	94.7	90.2	94.4	91.9	<b>90.8</b>
Recall	SCD loss	68.1	59.1	52.4	75.7	58.7	79.2	58.7	65.2
	USM SCD	75.3	90.8	81.7	76.5	82.7	70.1	87.3	<b>81.4</b>
F1	SCD loss	73.3	68.7	62.9	79.0	66.9	75.9	65.9	70.9
	USM SCD	82.6	87.6	86.9	84.6	86.3	80.5	89.5	<b>85.8</b>

**Table 3:** Effect of the choice of model parameters to fine-tune. The decoder and input processing layers are always fine-tuned (27M parameters). Evaluated on **YT-96-Eval**.

Fine-tuned Enc. layers	# Params Trained	WER	Precision	Recall	F1
First 4	254M	35.9	83.8	35.6	50.0
Last 4	254M	30.4	82.2	44.6	57.8
First 4 & last 4	480M	<b>30.1</b>	<b>84.0</b>	<b>52.5</b>	<b>64.6</b>
All	1.84B	<b>30.1</b>	82.4	51.9	63.7



**Fig. 2:** SCD token <st> posterior probability scaling results on **YT-96-Eval**.

and decoder projection layers, which consist of 27M trainable parameters. A preliminary experiment suggests that *only* fine-tuning the feature encoder, input projection, and decoder projection layers does not converge well. Therefore, we selectively fine-tune certain layers of the Conformer encoder and freeze the rest of the parameters. All models are trained for 40k steps. The results are in Table 3. We observe that optimizing the last 4 layers is significantly better than optimizing the first 4 layers both in terms of WER and SCD metrics. Interestingly, optimizing both the first 4 and last 4 layers (i.e., 8 of 32 layers) gives the best ASR and SCD performance, which only accounts for  $\sim 26\%$  of the trainable parameters.

#### 4.3. Effect of the speaker change token posterior scaling

Next, we study the effect of the speaker change token posterior scaling factor (cf. Sec. 2.4). Based on the results in Sec. 4.2, we use the model that is only fine-tuned on the first 4 and last 4 layers of the Conformer encoder (480M trainable parameters). We run experiments (see Fig. 2) by setting the factor  $\lambda$  from 1.0 to 9.0, with a step size of 1.0. Note that this experiment does not require retraining the model since the posterior scaling happens during inference. We observe that the posterior scaling does not significantly affect the ASR quality, with the maximum WER difference being less than 0.7% *relative* (i.e., from 30.1% to 30.3%). More importantly, the scaling

factor brings large gains in terms of SCD quality. Compared with the baseline configuration where there is no SCD posterior scaling (i.e., scaling factor 1.0), the best posterior scaling factor of 5.0 increases the SCD F1 score from 64.6% to 75.3%, a 16.6% relative improvement.

#### 4.4. En-US quality analysis

For En-US, there are additional internal and public datasets that have speaker change labels (Table 1). We evaluate the *USM-SCD* model fine-tuned from ASR on these datasets. We only fine-tune the first 4 and last 4 Conformer encoder layers, and the SCD posterior scaling factor is set to 5.0 during inference. The per-testset results are summarized in Table 4. We also include the best performing system from [11] (denoted as *SCD loss*, 27M parameters monolingual En-US model) as a comparison. The *SCD loss* system is trained with an SCD-optimized training loss on a super-set of the En-US portion of **YT-SUP**, with 2k hours of additional training data from other domains. We observe that the *USM-SCD* system performs much better than the *SCD loss* system, achieving 21% relative F1 score improvement. The precision and recall rates increase by 17.0% and 24.8% relative, respectively.

## 5. DISCUSSION AND CONCLUSION

In this work we propose a multilingual SCD model that supports 96 languages. We take advantage of recent advances in large speech foundation models to construct this USM-SCD model and study its properties through a series of ablation studies. We find that ASR-pretraining is crucial to model performance. We observe that we only need to fine-tune roughly one-quarter of the trainable parameters to achieve the best overall performance compared to fine-tuning all parameters. We also show that an inference-time SCD token posterior scaling that requires no additional computation can result in a 16.6% relative improvement in the SCD F1 score. Finally, compared with our previous monolingual En-US SCD model, the USM-SCD model outperforms it by 21% in terms of SCD F1 score. Based on benchmarks on TPU v5e [37], the USM-SCD model can run inference at 60x faster than real-time (batch size 1), demonstrating the application potential of this model. Possible future directions include replacing the CTC architecture with a fast RNN-T implementation and applying the token-level training loss proposed in [11] to further boost model quality. It is also interesting to explore multi-output RNN-T joint networks [38] to decouple the ASR and SCD tasks.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank Wei Han for the Whisper model evaluation setup, and Olivier Siohan, Parisa Haghani, Ignacio Lopez Moreno, and Pedro Moreno Mengibar for reviewing this work.



## 7. REFERENCES

- [1] Jitendra Ajmera, Iain McCowan, and Hervé Boudlard, “Robust speaker change detection,” *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649–651, 2004.
- [2] Wei Xia, Han Lu, et al., “Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection,” in *Proc. ICASSP*, 2022, pp. 8077–8081.
- [3] Quan Wang, Yiling Huang, et al., “Highly efficient real-time streaming and fully on-device speaker diarization with multi-stage clustering,” *arXiv:2210.13690*, 2022.
- [4] Leda Sari, Mark Hasegawa-Johnson, and Samuel Thomas, “Auxiliary networks for joint speaker adaptation and speaker change detection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 324–333, 2020.
- [5] Gregor Donabauer, Udo Kruschwitz, and David Corney, “Making sense of subtitles: Sentence boundary detection and speaker change detection in unpunctuated texts,” in *Companion Proceedings of the Web Conference*, 2021, pp. 357–362.
- [6] Maria Tsimpoukelli, Jacob L Menick, et al., “Multimodal few-shot learning with frozen language models,” *Advances in Neural Information Processing Systems*, 2021.
- [7] Marek Hruš and Zbyněk Zajíč, “Convolutional neural network for speaker change detection in telephone speaker diarization system,” in *Proc. ICASSP*, 2017, pp. 4945–4949.
- [8] Ruiqing Yin, Hervé Bredin, and Claude Barras, “Neural speech turn segmentation and affinity propagation for speaker diarization,” in *Proc. Interspeech*, 2018, pp. 1393–1397.
- [9] Hagai Aronowitz and Weizhong Zhu, “Context and uncertainty modeling for online speaker change detection,” in *Proc. ICASSP*, 2020, pp. 8379–8383.
- [10] Sam De Silva and Anthony Liu, “Europe’s tough new law on biometrics,” *Biometric Technology Today*, vol. 2017, no. 2, pp. 5–7, 2017.
- [11] Guanlong Zhao, Quan Wang, et al., “Augmenting transformer-transducer based speaker change detection with token-level training loss,” in *Proc. ICASSP*, 2023.
- [12] Jian Wu, Zhuo Chen, et al., “Speaker change detection for transformer transducer ASR,” in *Proc. ICASSP*, 2023.
- [13] Rishi Bommasani, Drew A Hudson, et al., “On the opportunities and risks of foundation models,” *arXiv preprint arXiv:2108.07258*, 2021.
- [14] Leonardo Pepino, Pablo Riera, and Luciana Ferrer, “Emotion recognition from speech using wav2vec 2.0 embeddings,” in *Proc. Interspeech*, 2021, pp. 3400–3404.
- [15] Hexin Liu, Leibny Paola Garcia Perera, et al., “Efficient self-supervised learning representations for spoken language identification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1296–1307, 2022.
- [16] Marie Kunešová and Zbyněk Zajíč, “Multitask detection of speaker changes, overlapping speech and voice activity using wav2vec 2.0,” in *Proc. ICASSP*, 2023.
- [17] Xiaoshuo Xu, Yueteng Kang, et al., “Explore wav2vec 2.0 for mispronunciation detection,” in *Proc. Interspeech*, 2021.
- [18] Yu Zhang, Wei Han, et al., “Google USM: Scaling automatic speech recognition beyond 100 languages,” *arXiv preprint arXiv:2303.01037*, 2023.
- [19] Anmol Gulati, James Qin, et al., “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. Interspeech*, 2020, pp. 5036–5040.
- [20] Alex Graves, Santiago Fernández, et al., “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006, pp. 369–376.
- [21] Daniel S. Park, William Chan, et al., “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. Interspeech*, 2019, pp. 2613–2617.
- [22] Alexei Baevski, Yuhao Zhou, et al., “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.
- [23] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv:1211.3711*, 2012.
- [24] Chung-Cheng Chiu, James Qin, et al., “Self-supervised learning with random-projection quantizer for speech recognition,” in *Proc. ICML*, 2022, pp. 3915–3924.
- [25] “Google’s privacy principles,” <https://googleblog.blogspot.com/2010/01/googles-privacy-principles.html>, Accessed: 2023-09-13.
- [26] “Artificial intelligence at Google: Our principles,” <https://ai.google/principles>, Accessed: 2023-09-13.
- [27] Zhiyun Lu, Yanwei Pan, et al., “Input length matters: Improving RNN-T and MWER training for long-form telephony speech recognition,” *arXiv preprint arXiv:2110.03841*, 2021.
- [28] Jean Carletta et al., “The AMI meeting corpus: A pre-announcement,” in *Machine Learning for Multimodal Interaction*, 2006, pp. 28–39.
- [29] A Canavan, D Graff, and G Zipperlen, “CALLHOME American English speech LDC97S42,” LDC Catalog. Philadelphia: Linguistic Data Consortium, 1997.
- [30] Neville Ryant, Kenneth Church, et al., “First DIHARD challenge evaluation plan,” Tech. Rep., Linguistic Data Consortium, University of Pennsylvania, 2018.
- [31] Christopher Cieri, David Miller, and Kevin Walker, “The Fisher corpus: A resource for the next generations of speech-to-text,” in *LREC*, 2004, vol. 4, pp. 69–71.
- [32] “Kaldi ICSI data split,” <https://github.com/kaldi-asr/kaldi/blob/master/egs/icsi/README.txt>, Accessed: 2023-09-13.
- [33] Noam Shazeer and Mitchell Stern, “Adafactor: Adaptive learning rates with sublinear memory cost,” in *Proc. ICML*, 2018, pp. 4596–4604.
- [34] Norman P Jouppi, Cliff Young, et al., “In-datacenter performance analysis of a tensor processing unit,” in *Proc. ISCA*, 2017, pp. 1–12.
- [35] Bo Li, Ruoming Pang, et al., “Massively multilingual ASR: A lifelong learning solution,” in *Proc. ICASSP*, 2022, pp. 6397–6401.
- [36] Alec Radford, Jong Wook Kim, et al., “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, 2023, pp. 28492–28518.
- [37] “Cloud tensor processing units (TPUs),” <https://cloud.google.com/tpu>, Accessed: 2023-09-13.
- [38] Weiran Wang, Ding Zhao, et al., “Multi-output RNN-T joint networks for multi-task learning of ASR and auxiliary tasks,” in *Proc. ICASSP*, 2023.