EKSTEP: OPEN SOURCE SPEECH RECOGNITION IN INDIC भाषाएँ மொழிகள் ভাষা

What is Speech Recognition?

What is Speech Recognition?

- Getting a computer to understand spoken language
- By understand we mean:
  - Convert input speech to text
- Famously known as Automatic Speech Recognition (ASR) or Speech to text (STT)

Objective of Ekstep Project

Objective of Ekstep Project

- Focus on 23 Indic Languages.
- Create data to build models in 23 Indic Languages.
- Open Source SOTA models for all the Indic Languages.
- In addition to models, open source data used to create models as well.

Why Open Source?

Why Open Source?

- Open source trained models, datasets & tools to encourage other technologists, to research & develop further products in local languages.
- Cloud services like Google and Azure offer speech to text services for few Indic languages. But the ecosystem is very closed.

Problems with Speech Recognition

Problems with Speech Recognition

- Domain and Speaker Agnostic.
- Ex: there are 70 Million people who speak Tamil, create a system that is Speaker Independent and works for all.
- Different dialects, pitch, volume.
- Different recording environment with different background noises and microphone equipment.
- It is very difficult to constraint the problem.

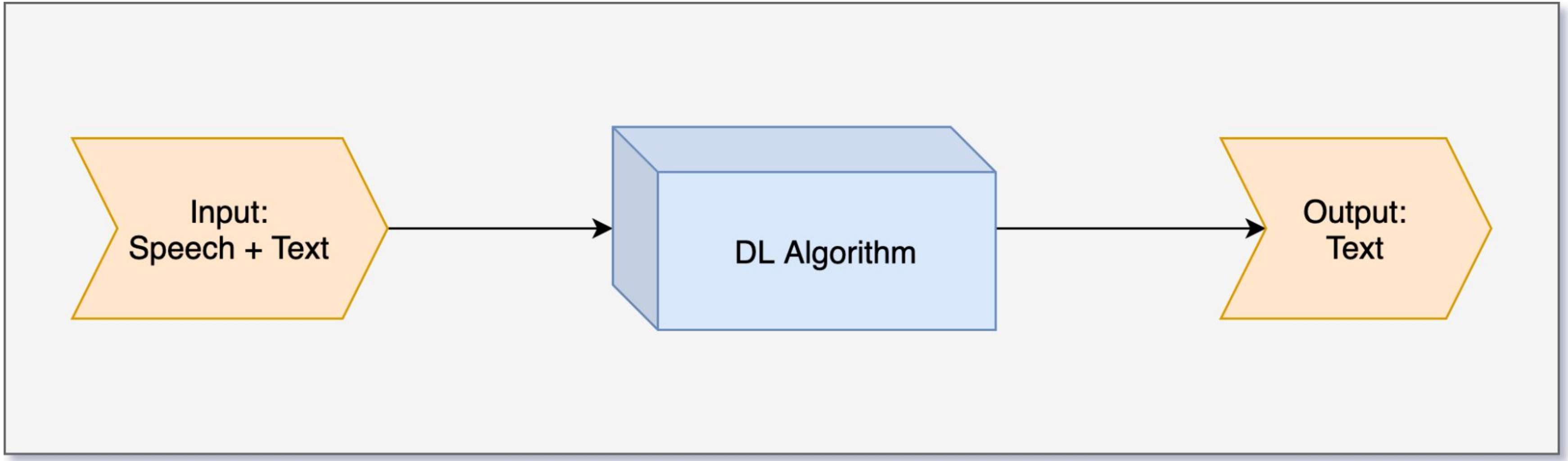
How to frame it as a Data Science Problem?

How to frame it as a Data Science Problem?

- Every Data Science Problem has 3 things:
    - Input Data
    - Algorithm
    - Output Expected

How to frame it as a Data Science Problem?

If we treat this as a supervised learning problem:



Approach 1

Treat this as a supervised learning problem

Approach 1

- DL based approach as traditional Kaldi based approaches do not scale very well.
- As a first POC, we decided to create ASR model for the most spoken language in India i.e. Hindi.
- Close to 57% of the population of India speaks Hindi.

DL Algorithm

Deep Speech 2: End-to-End Speech Recognition in English and Mandarin

Baidu Research – Silicon Valley AI Lab*
Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, Zhenyao Zhu

Abstract

We show that an end-to-end deep learning approach can be used to recognize either English or Mandarin Chinese speech—two vastly different languages. Be-

Input Data

Since we are using supervised learning we need to have labelled data. For example an audio clip and the corresponding transcript

test.wav

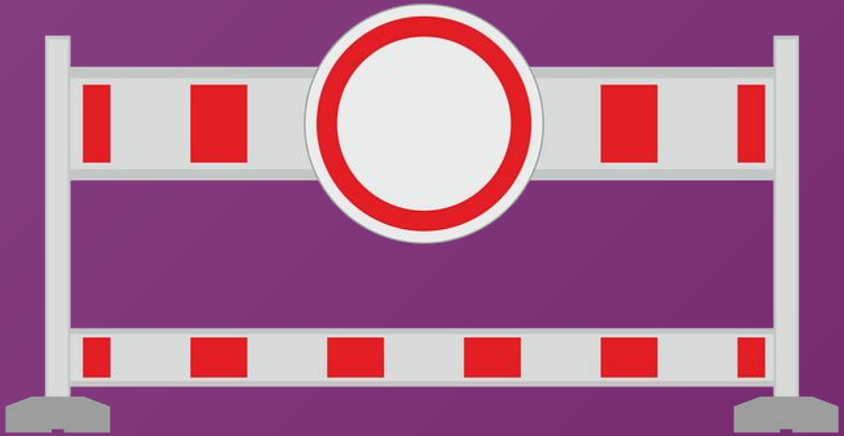
This is a sample speech text.

But how much data?

How much data?

- According to our research we found out we need 10,000 hours of labelled data to have a good production grade system.
- And that 10,000 hours have some characteristics:
    - Balanced on gender diversity
    - Have sufficient speaker diversity (maximum 30 minutes per speaker)
    - Have perfect transcription quality

ROADBLOCK



Roadblock

- We never had 10,000 hours of labelled data as per our requirements even in Hindi.

Solution?

Create our own Data.

Objective while creating data

- Find open source audio data from variety of different sources.
- Have a balanced gender ratio i.e. female to male.
- Have sufficient speaker diversity i.e. maximum 30 minutes from one speaker targeting 20,000 speakers in 10,000 hours of data.
- Have utterances with maximum duration of 15 seconds. Due to hardware limitations and model requirements the input audio at any point can be maximum of 15 seconds only.
- Have utterances with no/little background music/noise.
- Create near perfect transcript for the utterance.

But how to create data?

How to create data? Data Pipelines

1 — 2 — 3

Find Open Source Audio Data

Scrolling through Internet

Label metadata

Metadata like number of speakers, gender

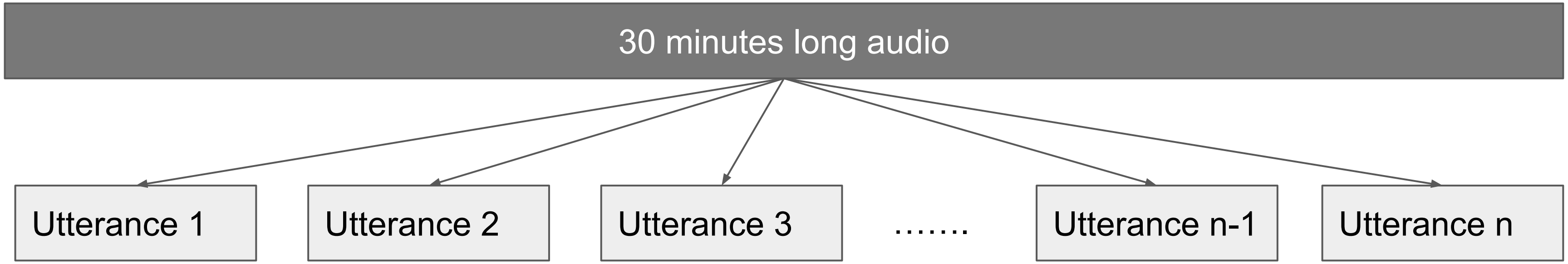
Data pipeline

Pass data through data pipeline to get it according to our needs

Data Pipeline: Step 1 - VAD

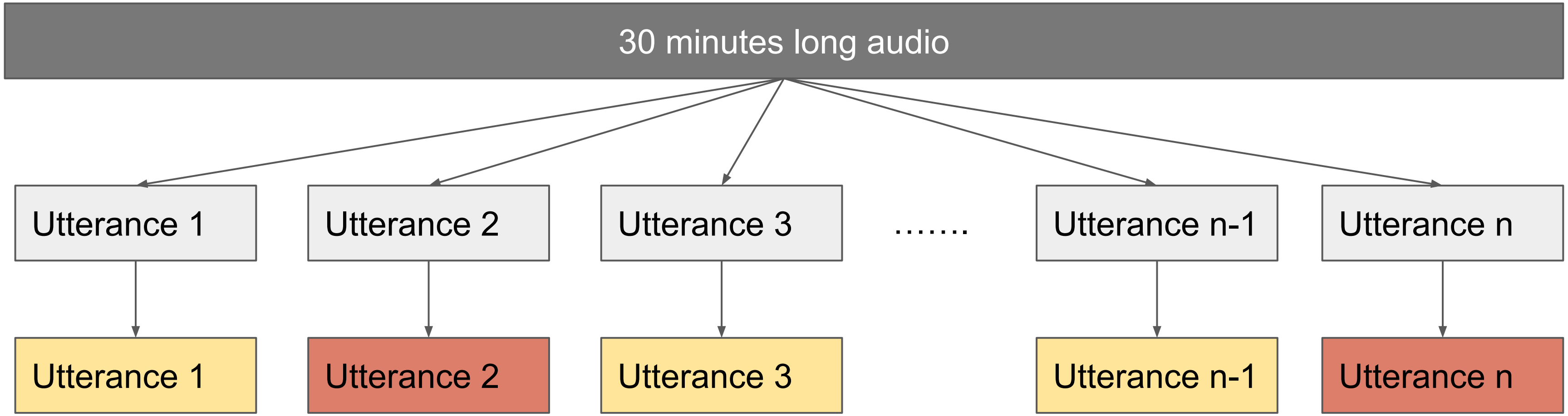
30 minutes long audio

Data Pipeline: Step 1 - VAD

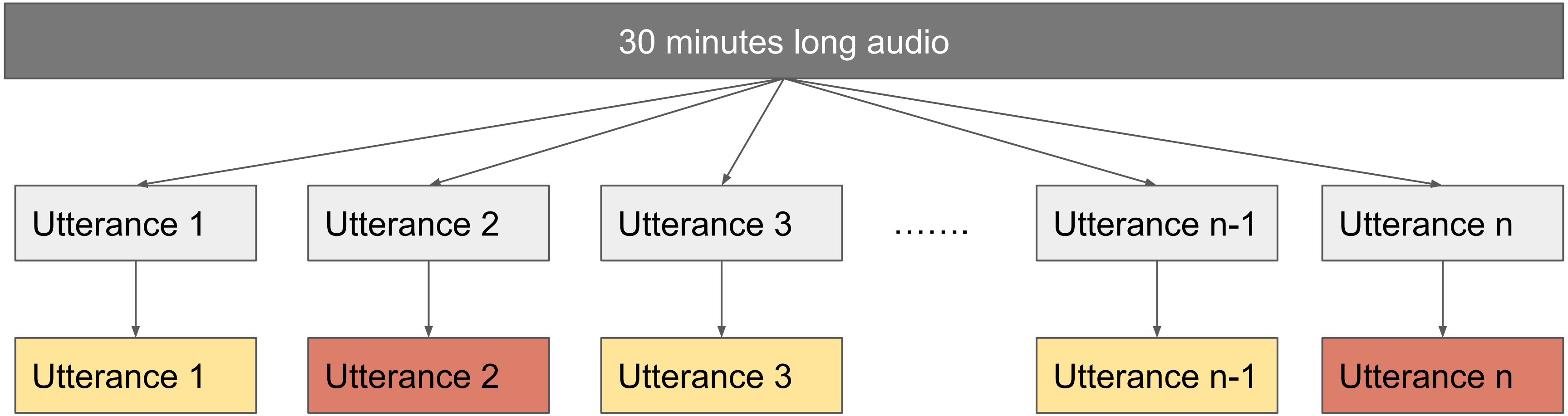


VOICE ACTIVITY DETECTION
Breaks Audio on Silences due to 15 second restriction.

Data Pipeline: Step 2 - SNR



Data Pipeline: Step 2 - SNR

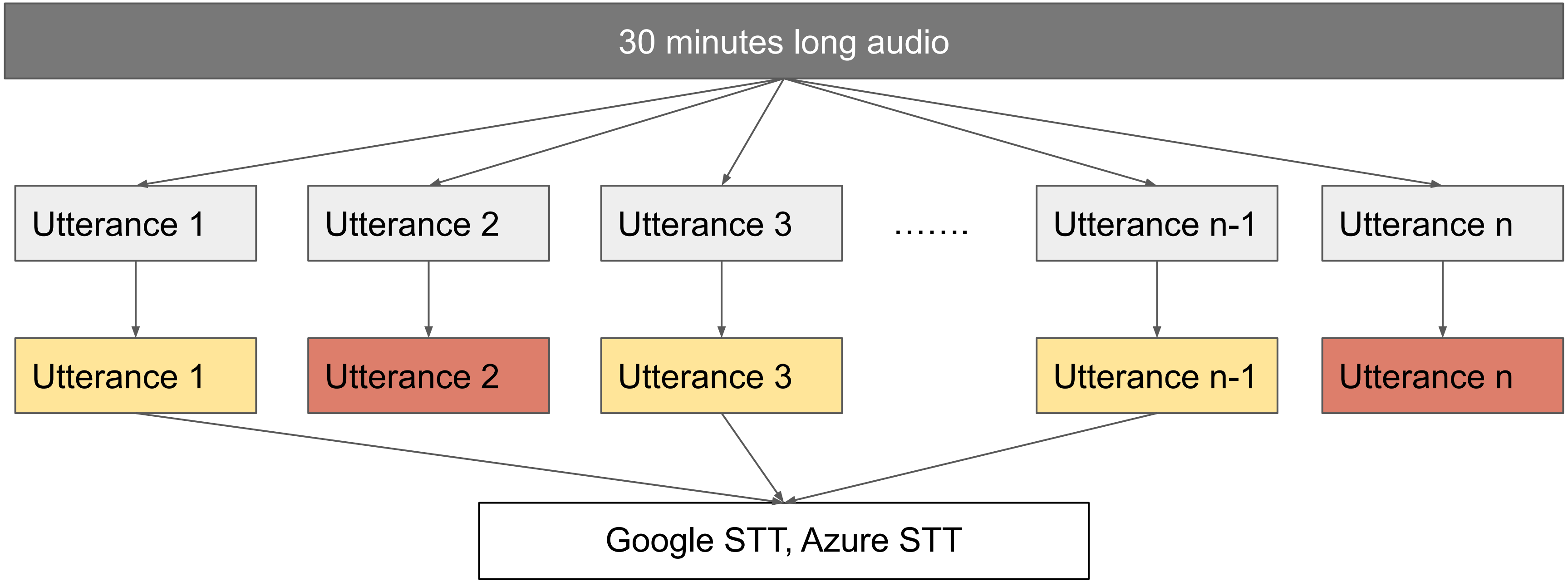


SNR (Signal to Noise Ratio)
Based on the threshold we can find noise.

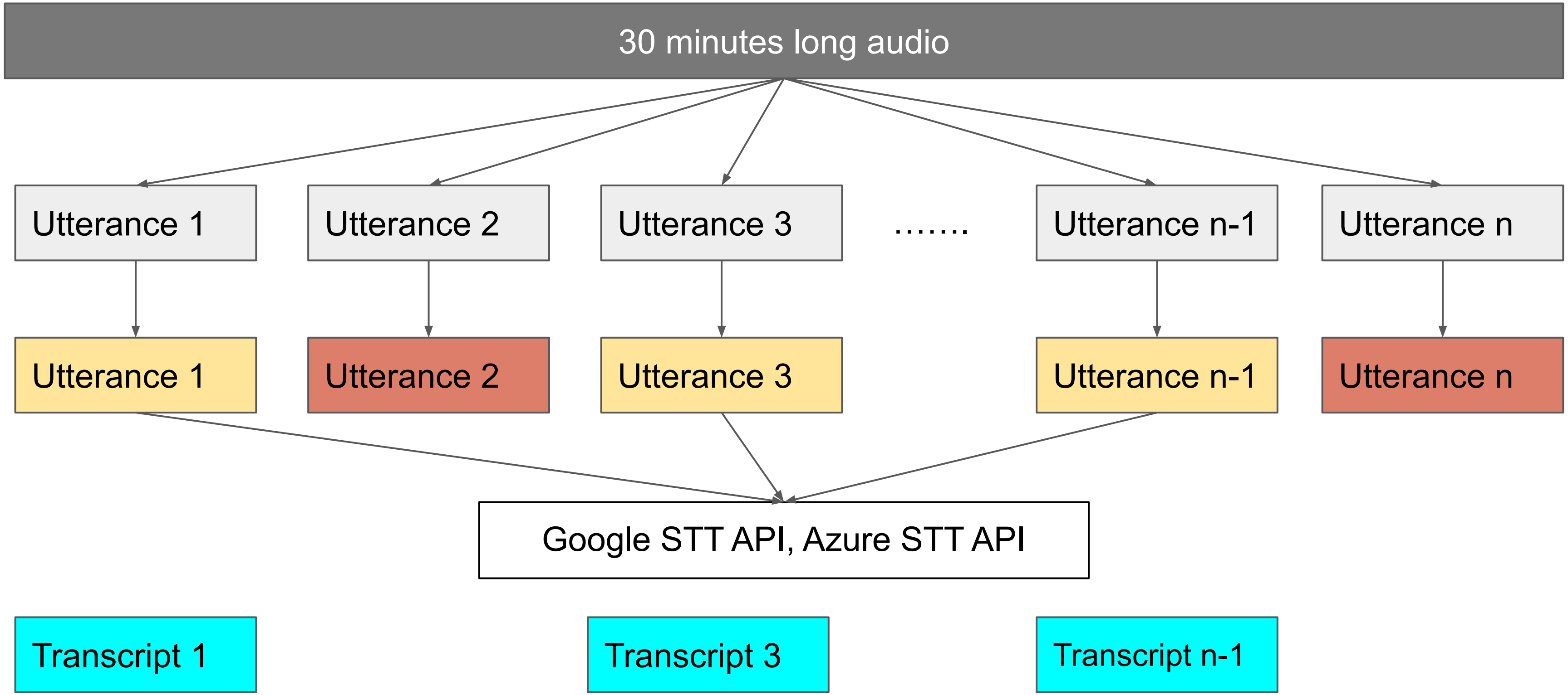
Not Noise

Noise

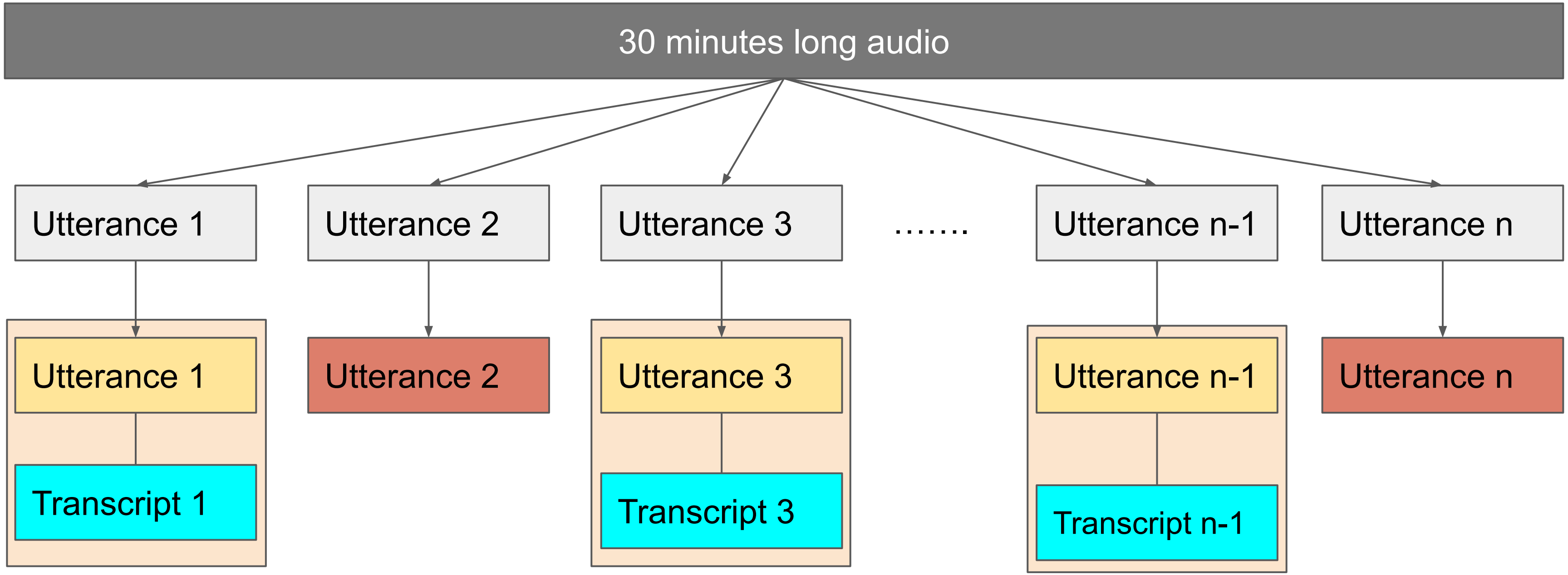
Data Pipeline: Step 3- STT



Data Pipeline: Step 3- STT



Data Pipeline: Step 3- STT



Note: We incur a Data loss here.

How to create data? Crowdsourcing

- We created an online portal called Vakyansh where people can come and donate their voice.

Modelling Data

Modelling Data

- We were able to create 4500 hours of Hindi Labelled data in 4 months using the Data Pipeline.
- We trained a model using Open Source Pytorch implementation of Deepspeech.
- We were able to get a WER of 35 on test set while WER of google ASR was around 30 on the same set.

But what is WER

WER

- WER is the metric to understand how good your speech recognition is working. (Lower is better)
- WER stands for Word Error Rate which gives the number of insertions, deletions and substitutions required to make two strings equal.
- It is based on Edit Distance where cost of each error is one.
- For ex:
  - Original: This deck is for the global community call
  - Predicted: This deck is for global comunity a call
  - WER: 3 (deletions -> the, substitutions -> community, additions -> a)

Approach 1 Summary

Approach 1 Summary

- We created 4000 hours of data and our deepspeech model was good enough to compare that with google.
- We deployed the model by creating a flask API to test out.
- Hurray!
- But?

Realizations from Approach 1

SCALE

Realizations from Approach 1

- We realized we will not be able to scale this approach to other 23 languages because:
  - It is very hard to find 10,000 hours of data in a language with all the constraints on speaker/gender diversity.
  - Some languages are very low resource languages
  - Manually finding data and labelling metadata is a very difficult task.
  - Not all languages have Google STT or Azure STT available.

Realizations from Approach 1

- We realized we will not be able to scale this approach to other 23 languages because:
    - It is very hard to find 10,000 hours of data in a language with all the constraints on speaker/gender diversity.
    - Some languages are very low resource languages
    - Manually finding data and labelling metadata is a very difficult task.
- So two steps that we needed to improve on:
    - Make the data pipeline as much as automated as possible. So that data discovery, metadata entry is automated.
    - Change our Deep learning algo so that learnings from one language can be transferred to other languages much like transfer learning in vision tasks.

Approach 2

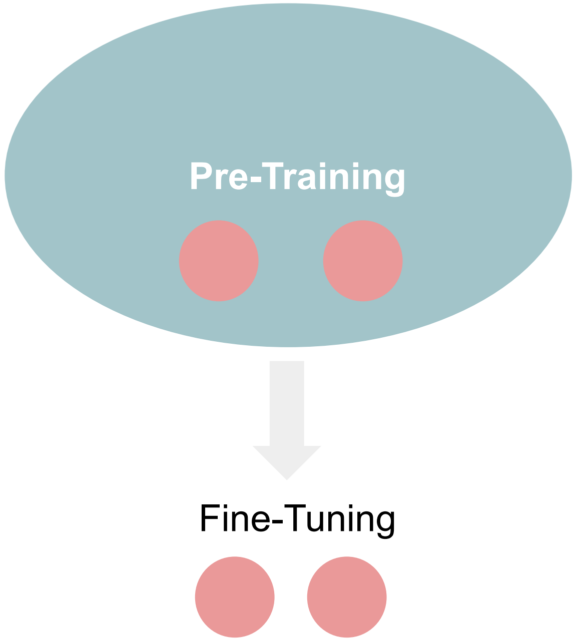
Changing Data Pipelines and Modelling Approach

Switching from supervision to self supervision

SELF SUPERVISED LEARNING

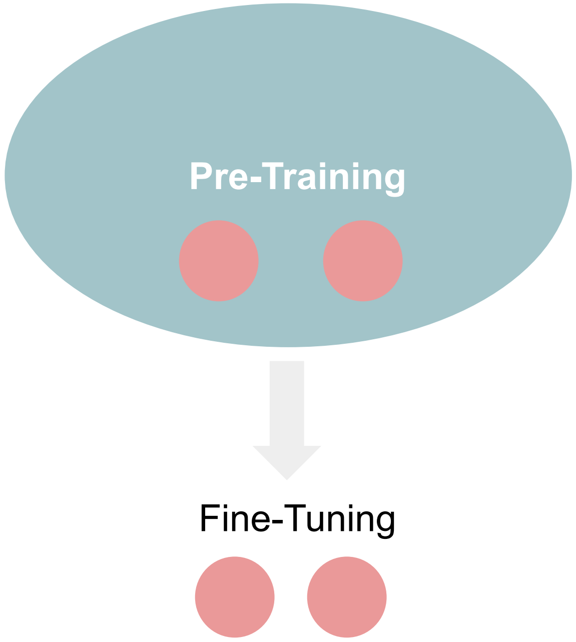
- **Self supervised learning** has been creating breakthroughs in NLP for the past 3-4 years (BERT, GPT-2, GPT-3, VIT etc).
- Self supervised learning consists of two steps in training.

  One is pretraining which is followed by finetuning.



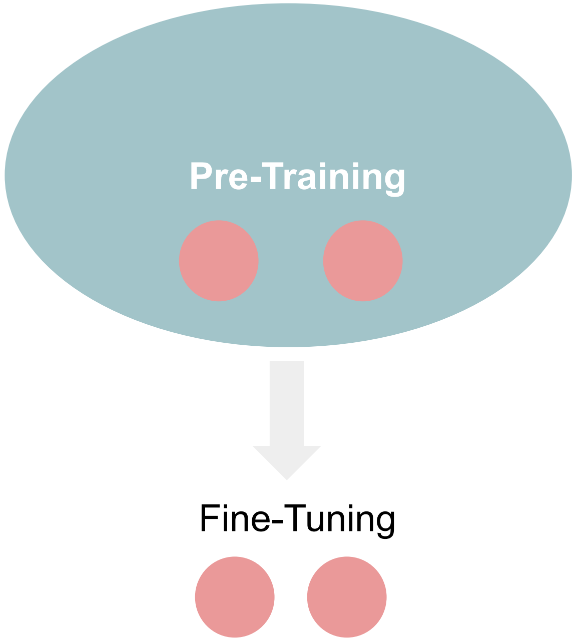
SELF SUPERVISED LEARNING

- In pretraining the model is trained on a huge corpus of unlabelled data.
- The task in pretraining is to use contrastive loss to model similar representations of data in same vector spaces.
- In finetuning the model is trained on a limited amount of labelled data to map the representations learnt during pretraining to labels that we want to predict.



WAV2VEC 2.0

- A new deep learning algorithm (developed by Facebook AI) which solely uses **unlabelled** audio data for **pretraining** to learn speech representations.
- Pretrained model is then **Finetuned** on a small amount of language specific **labelled data** to develop speech recognition models.
- So through self supervised learning, **Knowledge transfer** between languages was now possible.
- This was a new algorithm which was tested only for English, so we decided to give a try for Indic Languages.



SPEECH RECOGNITION - OUR STRATEGY

Supervised Learning			Semi Supervised Learning
Approx 10,000 hrs of labelled data	Data Magnitude	100x	Approx 100 hrs of labelled data
Approx 4,000 hrs of labelled data in 4-5 months	Data Acquisition time	7x	Approx 100 hours of labelled data in 2-3 weeks
1 Language in 1-1.5 yrs, based on data acquisition time	Go To Market / Language *accuracy at par with Google	15x	1 Language in 4 weeks

WHAT IT MEANS FOR US!

Model Training Cost

- The expensive part in model training is the pre-training step.
- If we want to train a model on a new language, no pre-training needs to be done which can bring down compute cost down by ~90%

Model Training Time

- Older techniques showed less promise for transfer learning among languages.
- Now phonetic similarity between languages can be leveraged which can reduce training times drastically.

Data Collection

- Acquiring labelled data is very difficult.
- Now, the majority chunk of data required is solely audio.
- Acquiring 100 hours of labelled data is much more easier than 10,000 hours.

Proceed to Wav2vec

Automating Manual Parts in the pipeline

AUTOMATED PIPELINE PROCESS

Data Collection	Data Validation	Data Processing & Filtering
Crawlers - Automated & License (CC)	Cleaning & Quality - Language, Songs/Music, Noise (SNR)	Cataloguing & Filtering - Speakers, Gender, Max Duration, Transcription

DATA COLLECTION PROCESS

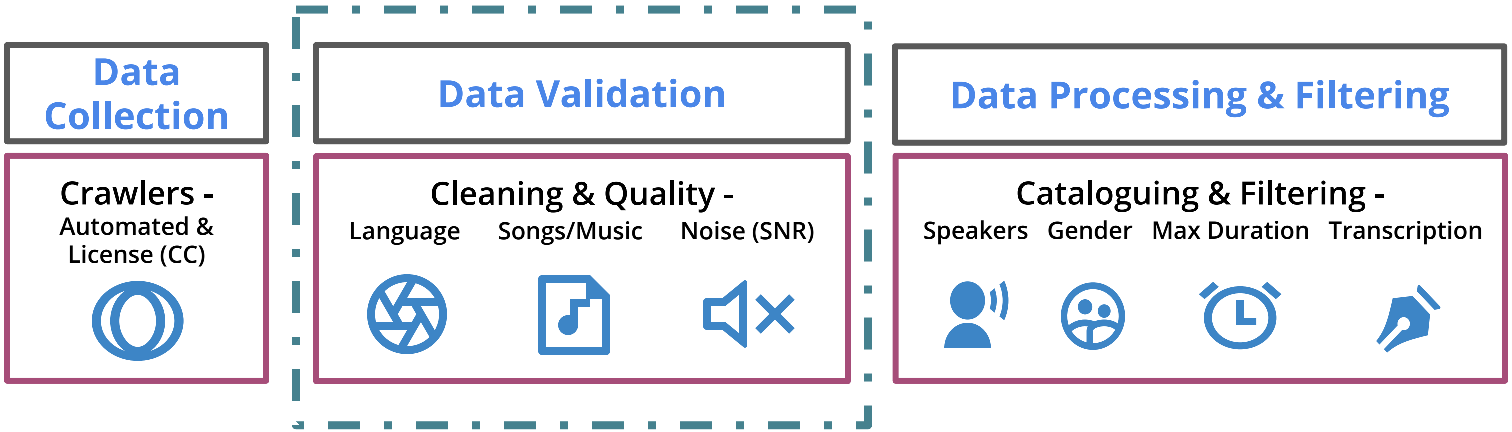
Data Collection

Crawlers - Automated & License (CC)

Data Collection Pipelines

- ❑ Search Audio and Video datasets thru' **Crawlers**
  - ❑ **Creative Common** Datasets
  - ❑ **Licensed** Audios/Videos
    - ❑ Approval request to the owners for **usage permission**

DATA VALIDATION



Dataset Validation

- ❑ Data collected thru' Crawlers inherently has **lot of noise,** such as, traffic, songs or could be for a different language than intended

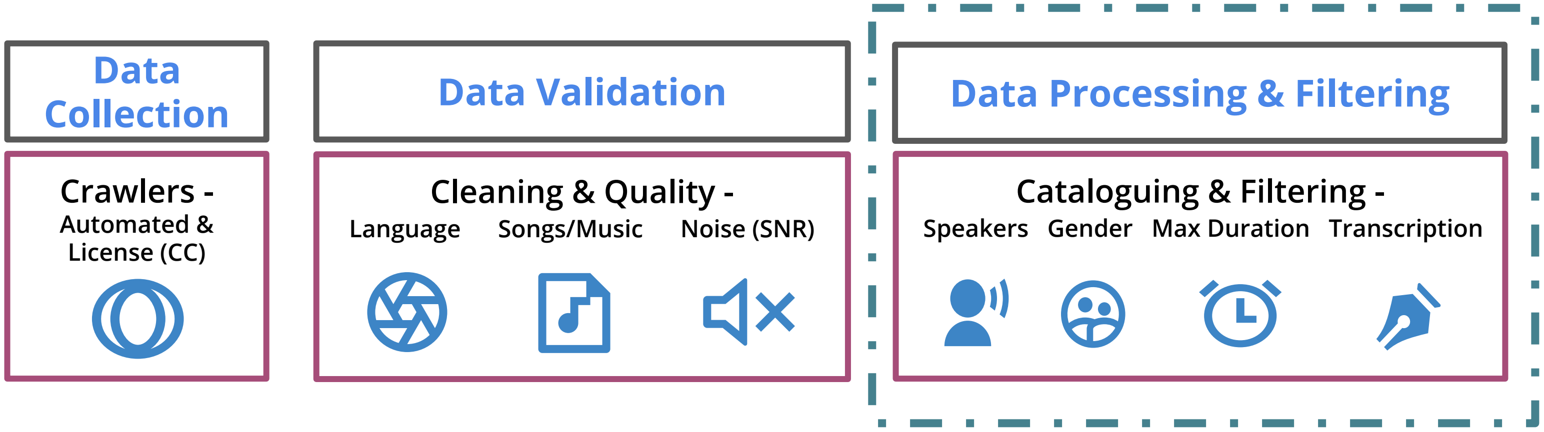
DATA VALIDATION - LANGUAGE IDENTIFICATION

- ❏ **Automate** and **expedite** the data collection process
- ❏ Train our own **deep learning model** to identify language
- ❏ Filter out audios which do not belong to the the language we want to train on.
- ❏ Power spectrogram of the audio is extracted and passed through a CNN (Resnet 18) for the classification task.

DATA VALIDATION - SONGS/MUSIC

- Songs and music have different properties from speech.
- The phonemes are stretched and add noise to training data for speech recognition.
- We are training our own models to filter out songs from speech since we don’t want songs in our training data.
- Mel spectrograms are extracted and passed through a CNN for the classification task.

DATA PROCESSING



Dataset Processing & Filtering

- ❑ An E2E ASR algorithm requires certain characteristics and diversity in the data for an effective training
- ❑ Data is filtered based on the desired diversity in gender, accent and dialects
- ❑ A balanced duration/speaker helps generalize the learning of the model

DATA PROCESSING & FILTERING

Speaker Identification

Issue: Identifying Speaker diversity in enormous amount of data is very tedious for a manual job

- ❏ The higher the speaker diversity in data, the better **generalized the model** is, for the multilingual Indian culture
- ❏ We have developed our own methods to **estimate number of speakers** in the audio data on which we will train our model.
- ❏ Speech sample is passed through a vocoder which is a deep net trained on a speaker contrastive task. Hierarchical clustering methods are then used to estimate the number of speakers.

DATA PROCESSING & FILTERING

GENDER IDENTIFICATION FROM SPEECH

- ➢ The training data should be as balanced as possible.
- ➢ If the training data is biased towards a certain speaker property the model has a high chance of being biased as well.
- ➢ Males and females have different frequency range while speaking.
- ➢ We use a vocoder based to encode speech sample into a vector which is trained on a speaker contrastive task.
- ➢ Fitting a classifier on those embeddings gave us a good way to differentiate between males and females.

DATA PROCESSING & FILTERING

SPEECH TO TEXT (STT)

- ➢ A training dataset comprises of
  - ❑ **Audios**
  - ❑ Corresponding **transcription**
- ➢ Transcription can be generated thru' multiple ways :
  - ❑ **Human Transcription** has higher accuracy
  - ❑ **Commercial STT products** for supported language
    - ❑ The **hybrid of various STTs, could avoid biases** in our model towards any particular commercial STT

OVERALL MODELING PROCESS

Data Collection	Data Validation	Data Processing & Filtering	Experiments
Crawlers - Automated & License (CC)	**Cleaning & Quality -** Language, Songs/Music, Noise (SNR)	**Cataloguing & Filtering -** Speakers, Gender, Max Duration, Transcription	**Training & Inference** Pre-Training & Fine Tuning

EXPERIMENTS

Model Training

- Pre-training
    - Phase that learns phonemes of a language thru' audio data only
    - Dataset requires diverse of environment but not essentially demographically balanced since the emphasis is on learning speech representations
- Fine tuning
    - Requires Labelled data in small amounts
    - Balanced data that ensures diversity in Gender, speaker's accent and dialects and a healthy duration/speaker

Model Inference & Testing

- Validate the Model on unseen dataset with predictable expected results
- Compute WER
- User testing by folks proficient in the language, on Inference website

Experiments

Training & Inference

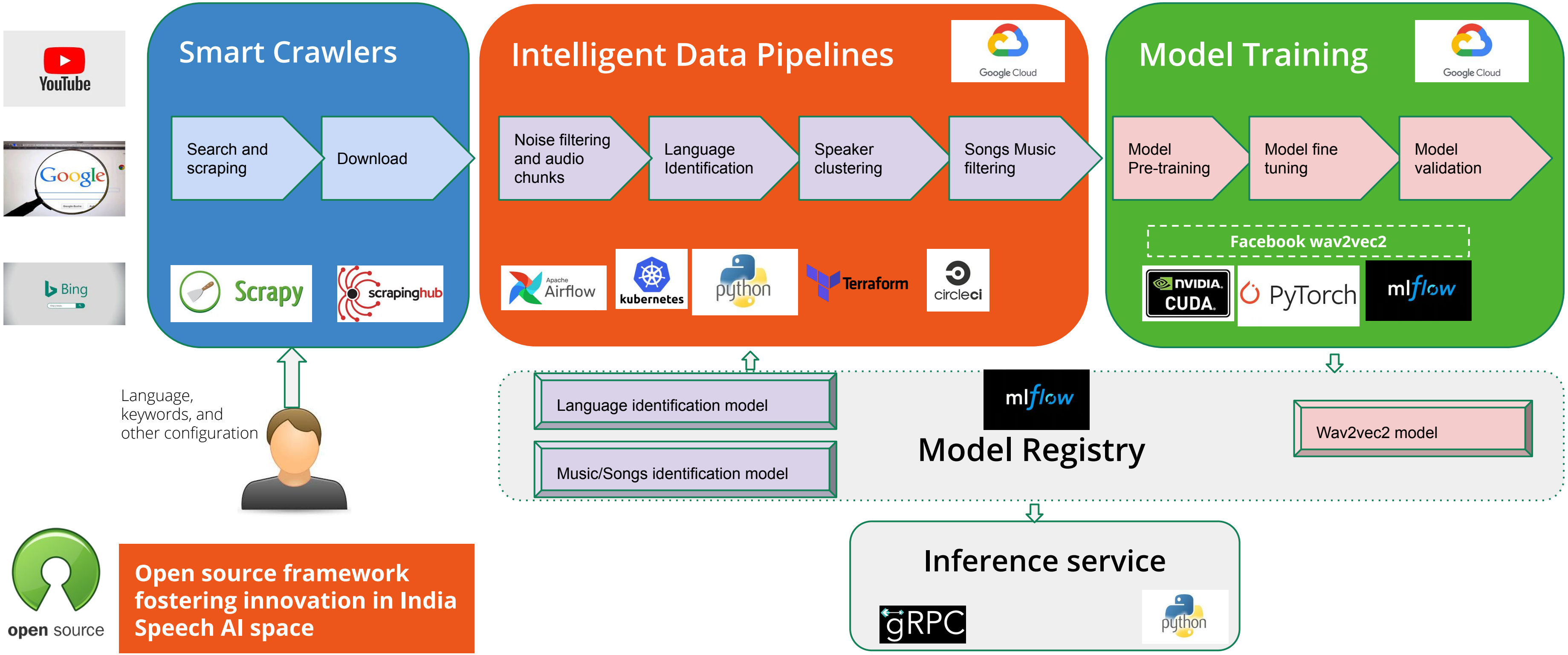
Pre-Training & Fine Tuning

Ekstep Speech Recognition Tech : One View

- Auto ingest 5000 hrs for one language in 2-3 days
- Automated catalogue building

- Language identification, Speaker Clustering, Songs/Music filtering
- End to end automation : Faster time to market

- Cross language learning and unsupervised learning leading to faster time to market



Open source framework fostering innovation in India Speech AI space

Hardware used for Training

- We use NVIDIA A-100 and V-100 GPU’s for model training.
- A-100 GPU’s are the costliest and fastest GPU’s present on the planet.
- Mostly 32xCPU machines are used
- RAM requirements range from 100GB - 200GB.

ACHIEVEMENTS

हिन्दी

ગુજરાતી

Accuracy at par
with Google/Azure

ENGLISH

தமிழ்

తెలుగు

CURRENT RESULTS

- **Pretrained** our model on **4200 hours** of **Hindi** audio data.
- **Fine-tuned** on **individual languages.**
- Deployed models for 8 languages uptil now.

Language	WER	Google's WER
Hindi	26.007	25.9
Gujarati	22.5	26.86
Tamil	28.3	27.41
Telugu	29.35	31.51

ACHIEVEMENTS - THE FIRSTS

Lowest GTM Time for a language

Automated Intelligent Data pipeline open sourced

Onboarding languages with Zero dataset / STT support

Vakyansh - India’s answer to Mozilla Voice

Open Source Model, Utilities and datasets

Achievements

First in English ASR Challenge

- Hosted by Indian Institute of Technology, Madras Speech laboratory. This challenge was a part of the National Language Translation Mission funded by MeitY, India.
- The task was to transcribe the NPTEL lectures and we used Vakyansh and wav2vec2 to create ASR models for Indian Accented English in addition to mathematical symbols like alpha, beta, gamma etc in the transcription too.
- We beat already established players like Samsung, Jio

NPTEL EVALUATION SET

Open Task

Position	Team name	Best Score(WER %)	# Submissions	Best Approach
1🥇	Ekstep_Thoughtworks	5.84	3	Wav2vec2 Pretrained on English Training Data + Extra data + Finetuned on the training data + 5 gram LM from training and extra data.
2🥈	BUT	9.78	7	TDNN_V2
3🥉	Scribetech	10.12	6	Kaldi Chain + Transfer Learning + RNNLM (Train+Dev)
4	IIT_Hyderabad	10.85	2	Kaldi_Chain with CPC_features extracted from pretrained models + RNNLM
5	CDOT	11.6	5	bigger lm, baseline am , 4 gram model
6	NITAP_Cognizyr	13.33	10	Transfer Learning from 62 phones with extended vocabulary related to domain

Closed Task

Position	Team name	Best Score(WER %)	# Submissions	Best Approach
1🥇	Ekstep_Thoughtworks	5.79	5	Wav2vec2 Pretrained on English Training Data + Finetuned on the training data + NO LM Present.
2🥈	Samsung_R_and_D_Bangalore	8.39	5	Espnet transformer with LSTM LM, Data augmentation using speed and volume perturbations
3🥉	Sayint_Zen3_Info_Solutions	8.97	4	Kaldi Chain Subword Model
4	IIT_Hyderabad	10.32	1	Kaldi_Chain+ RNNLM
5	BUT	10.33	1	Mono.graph.my_v1.rnnlm
6	Armsoftech_a	11.27	5	Kaldi Chain model with RNNLM
7	IIIT_Dharwad	11.27	7	RNNLM
8	CDAC_Pune_b	11.46	5	Kaldi chain

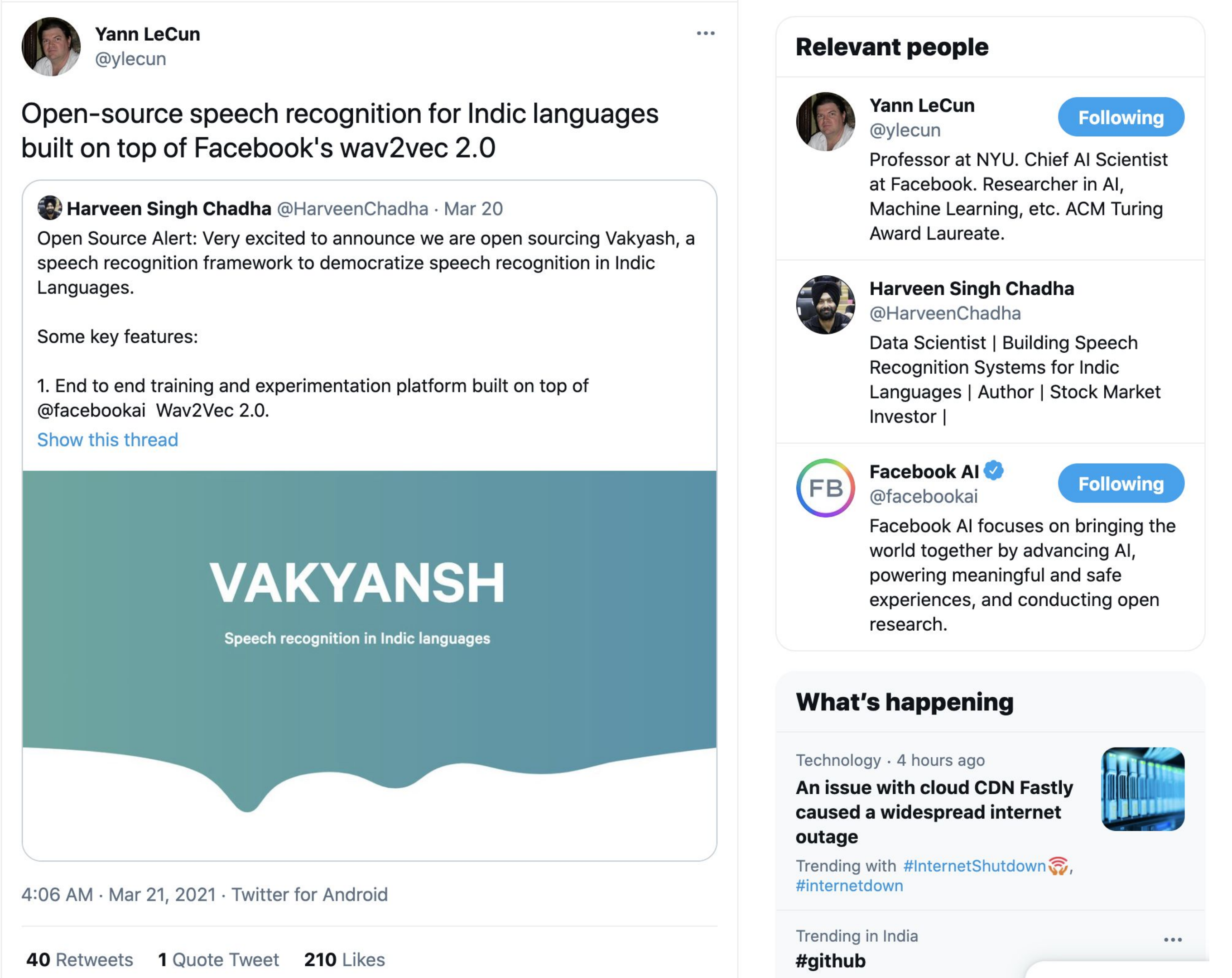
Achievements

Top 5 in Interspeech 2021 Hosted by Microsoft

- The task was to build a multilingual ASR which supports 5 Indic languages (Hindi, Odia, Gujarati, Tamil & Telugu) i.e. a single model or a single system that is capable to detect the audio language and produce transcript in the respective audio language.
- No team was able to beat our result in Hindi.

#	Team Name	Hindi (% WER)	Oriya (% WER)	Tamil (% WER)	Telugu (% WER)	Gujarati (% WER)	Average (% WER)
1	CSTR	14.33	25.34	23.16	21.88	20.59	21.06
2	Bytedance-SA	16.59	17.81	28.59	25.37	21.3	21.93
3	EthereumMiner	17.54	19.99	28.52	26.08	20.11	22.45
4	Lottery	17.81	17.74	30.69	27.67	23.62	23.51
5	Ekstep	12.24	27.1	27.2	22.43	30.65	23.92
6	Uniphore	22.79	29.55	18.8	28.69	22.79	24.52
7	GOT-HIM	17.72	29.14	27.94	26.36	22.62	24.76
8	GoVivace	21.77	29.05	28.92	26.5	21.22	25.49
9	IITM-SMT-Lab	17.8	32.21	27.12	28.11	29.8	27.01
10	TCS-SpeechNLP	19.77	35.21	26.26	26.82	28.53	27.32
11	TUTU	19.93	34.18	27.69	30.25	25.34	27.48
12	Dialpad	21.49	32.13	28.6	28.03	34.57	28.96
13	IIITHSPL	31.11	37.19	35.03	17.0	26.94	29.45
14	ScribeTech	27.78	34.57	33.01	30.08	28.22	30.73
15	Sayint	28.01	35.21	35.76	32.14	28.09	31.84
16	HAL101	21.42	34.66	37.92	33.92	34.37	32.46
17	SRI-B	30.84	49.8	26.07	28.34	27.61	32.53
18	Jio Speech	35.53	38.55	33.69	31.14	24.79	32.74
19	Baseline	37.2	38.46	34.09	31.44	26.15	33.47
20	Nuronics	38.02	48.4	34.89	33.11	29.68	36.82
21	IITM Speech Lab	23.79	37.95	52.27	43.98	41.86	39.97

- This tweet has half a million impressions



What are we doing currently?

We have pretrained a model specific for Indic Languages by creating 10,000 hours of unlabelled data.

We are now in a phase where we have started finetuning models for all the 23 Indic Languages.

Real Time Demo

THANK YOU