Natural Language and Speech Processing

Lecture 14: Some advanced topics in language and speech technology

Tanel Alumäe

Contents

- Weakly supervised training for speaker identification (Martin Karu's MSc thesis)
- Speaker identification using large language models (Oleksanda Zamana’s MSc thesis)
- Collar-aware speaker change detection (Joonas Kalda's PhD work)
- Hybrid spoken language identification (my own work)

Different kinds of supervision in machine learning

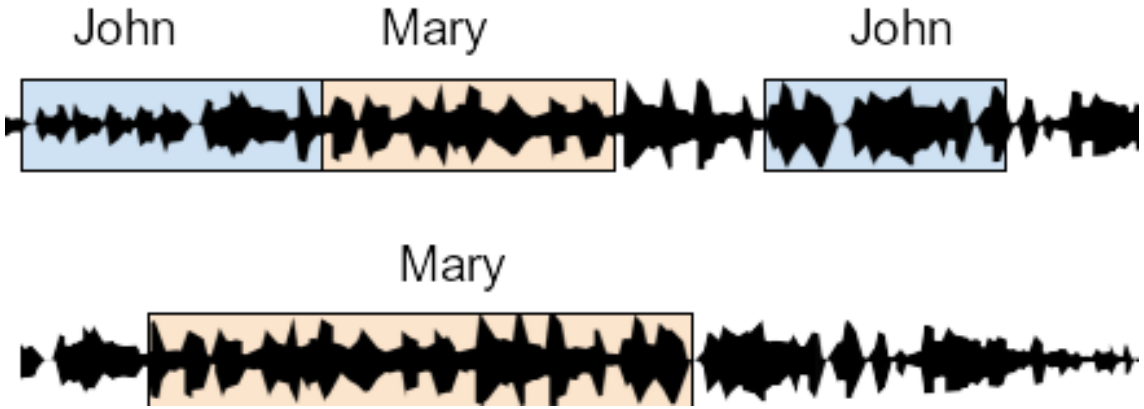
- Fully supervised
  - All training examples are labeled
  - Goal: train a model that can predict the labels for unseen samples
- Unsupervised
  - There are no labels at all, only data
  - Goal: find structure (e.g. clusters) in the data
- Semi-supervised
  - There is some labeled data, and (a lot of) additional unlabeled data
  - Goal: train a model that can predict labels for unseen samples, but try to make use of additional data
- Weakly supervised, distantly supervised

Weakly supervised training

- Weak supervision is used when full data labeling would be very expensive but weak labels come cheap
- Many different types of weak supervision, e.g.
  - **Candidate labels:** each training sample is assigned many labels, but only one of them is correct
  - **Probabilistic labels**: each label is assigned to each instance, with a given probability
  - **Incomplete labeling:** each training sample belongs to multiple classes, but only a partial set of the classes are labeled for each sample
  - **Crowd annotation**: labeling is done by non-expert and cheap labor, thus the labels are not very trustworthy
  - **Label proportions**: we know the proportions of the labels in a set of instances, but we don't know which ones precisely correspond to each label

Supervised training for speaker recognition

- Speaker recognition: identify a person based on the voice
- Training speaker identification models usually requires **hand-segmented** training data
- Producing such data is costly
- Difficult to cover a wide range of speakers
  - E.g., thousands of politicians for media monitoring purposes
- Difficult to keep up-to- date, enroll new speakers
  - Requires hand-labelling additional data

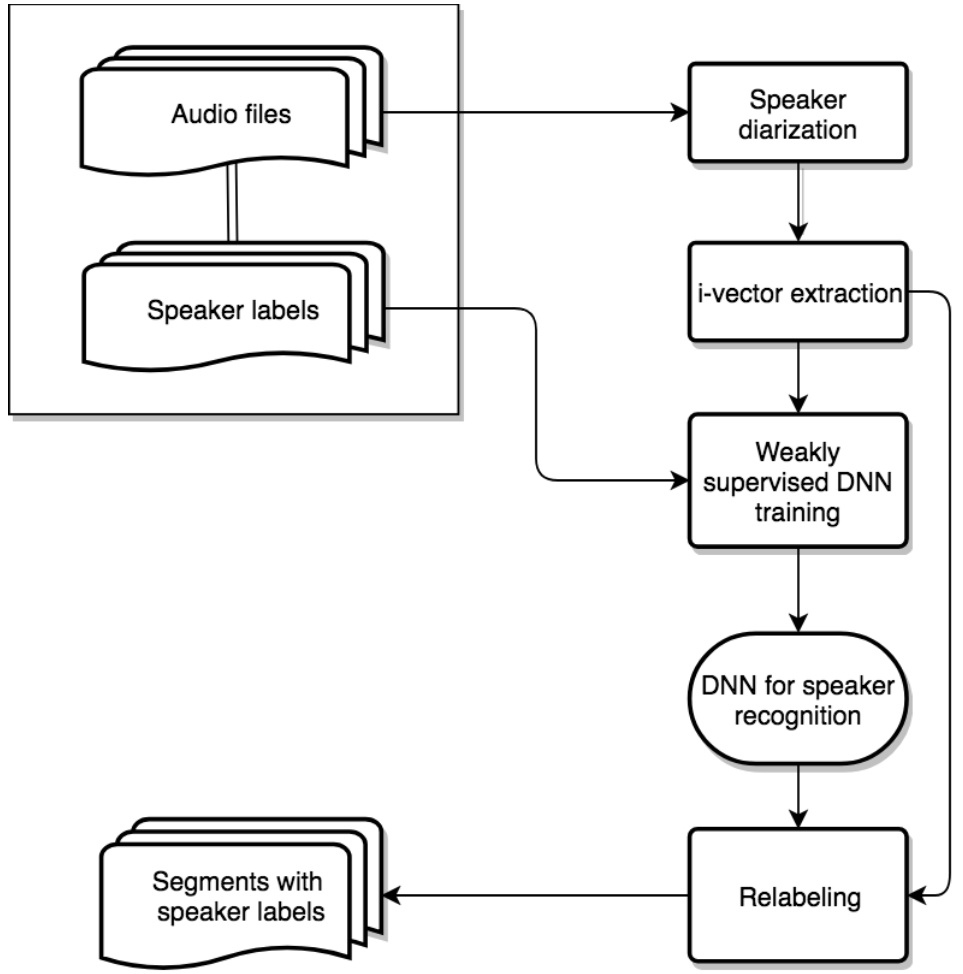


Weakly supervised training for speaker ID

- This method requires only recording-level labels
- Set of speakers who appear somewhere in the recording
- Time-based annotation is not required
- Producing such data is much easier

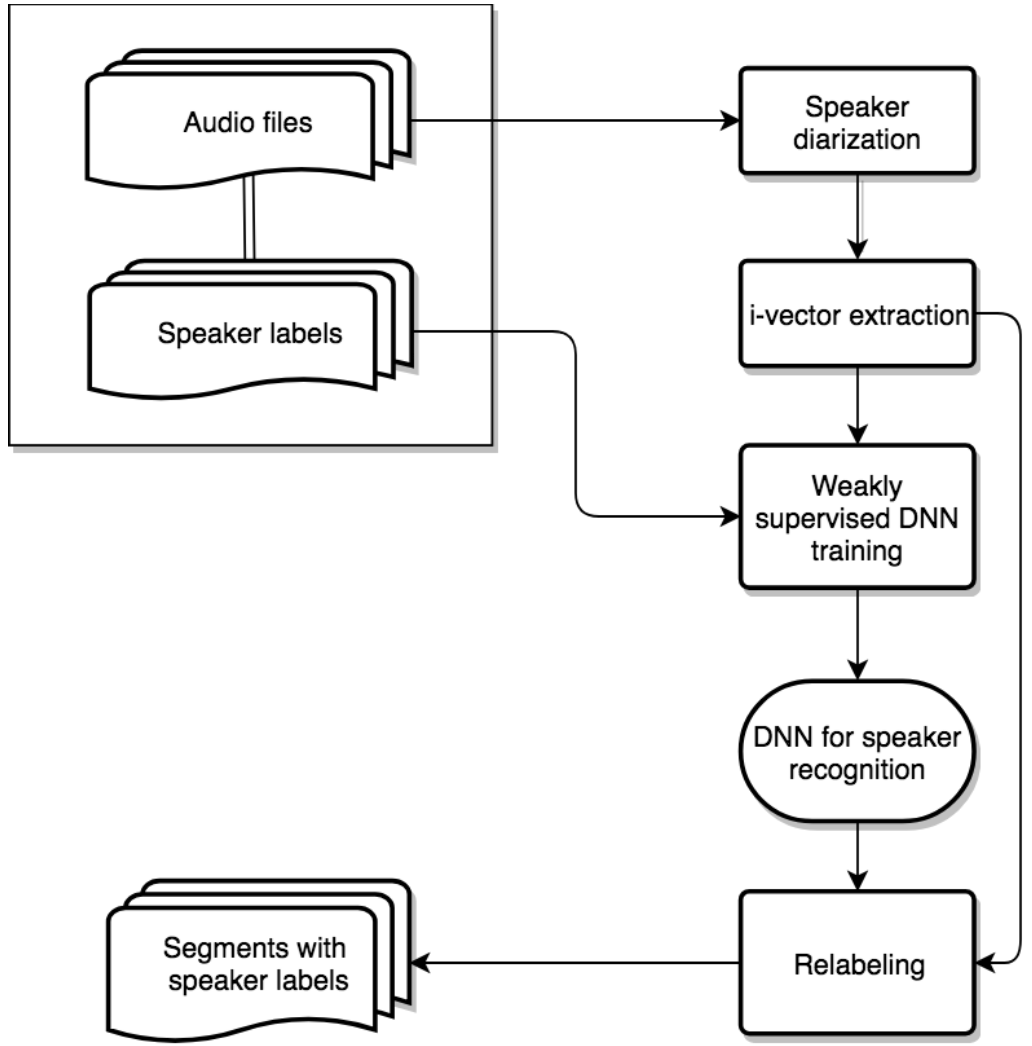
Method overview

- Training data: list of audio files, and the corresponding sets of speakers
- Training:
  1. Speaker diarization
  2. Embedding (i-vector, x-vector) extraction
  3. Train a DNN for classifying embeddings to speakers (including unknown speaker), using a special cost function
  4. Relabel the segments of the training data, using the trained DNN
  5. Apply any speaker recognition method (e.g., x-vectors + LDA/PLDA)



Training data

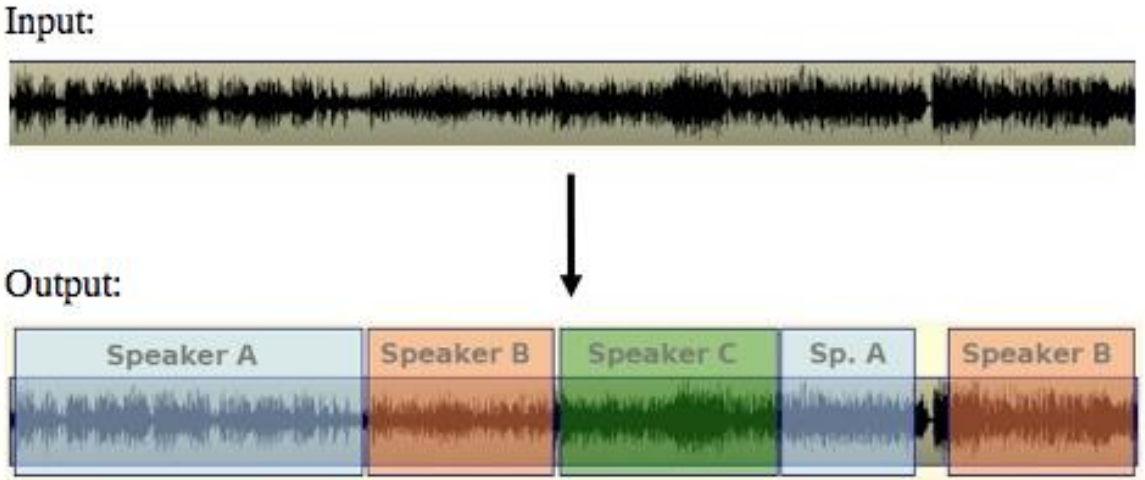
- Many audio files
  - Speaker labels for each file: a set of persons (who we want to cover with our model) who speak there
  - There can be other speakers who do not interest us
- Each person-of-interest (POI) should appear in several recordings (the more the better)
- Two persons should not appear always in the same recordings (no way to distinguish them)



Speaker diarization

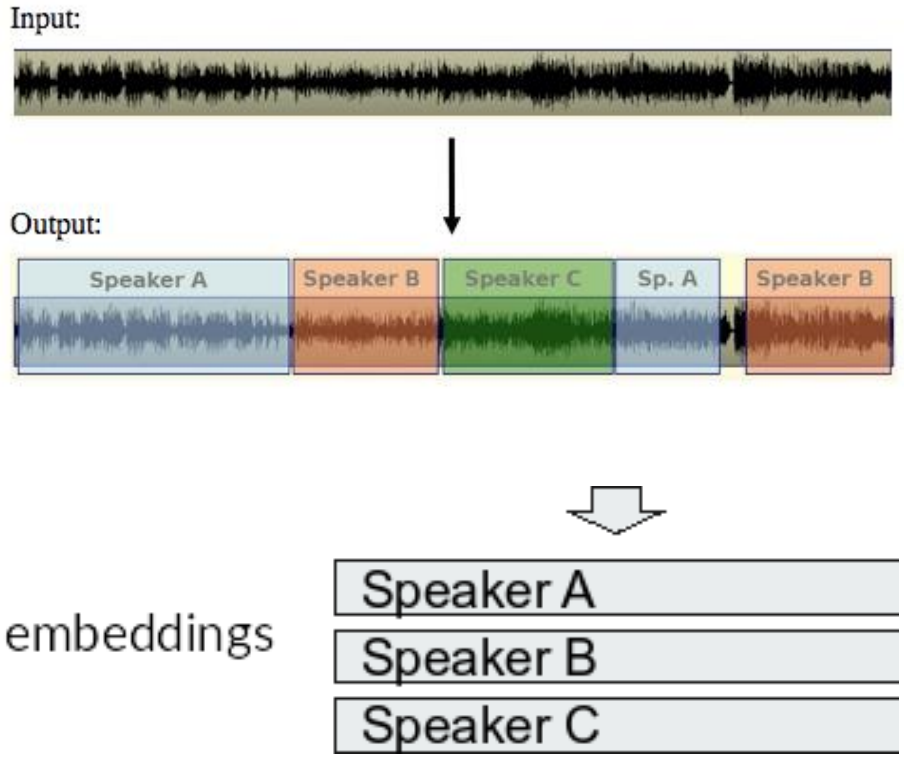
Bottom-up speaker diarization:

1. Detect speaker change points, using a sliding window
   - This results in segments where the speaker is homogeneous
2. Fine-tune segment boundaries to low energy regions (to avoid having a boundary in the middle of a word)
3. Remove segments containing non-speech (e.g., music, noise, silence)
4. Cluster resulting segments, using hierarchical agglomerative clustering



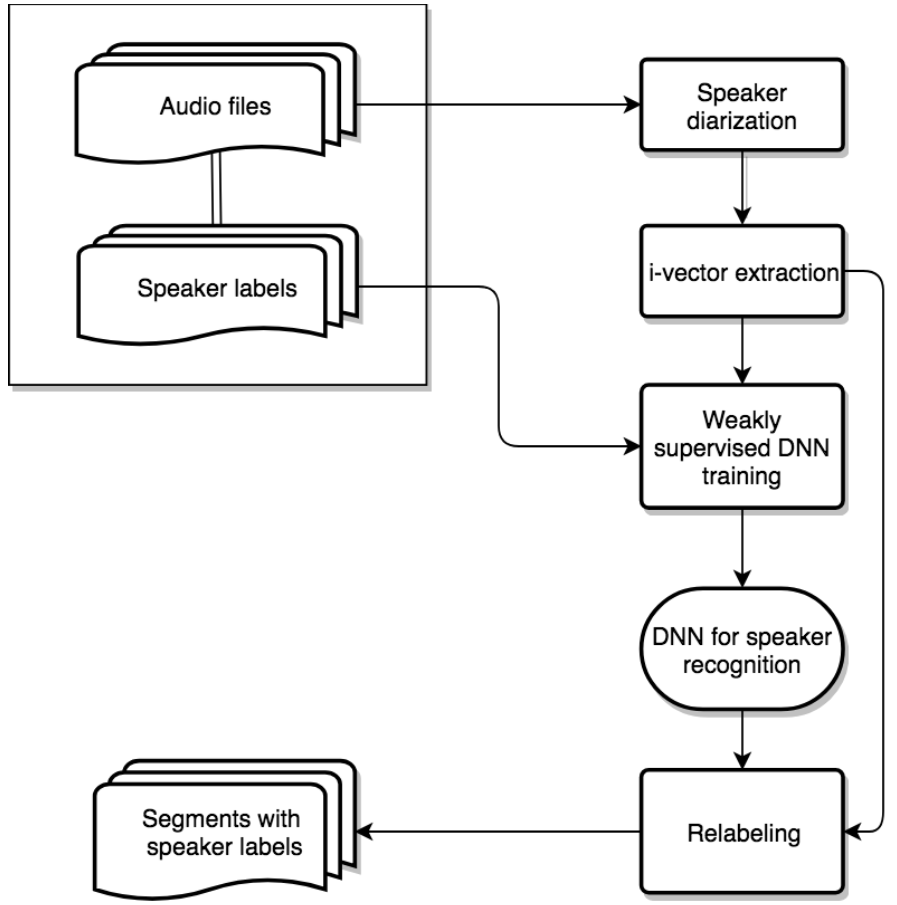
Extraction of speaker embeddings

- A method for mapping variable-length speech utterances to a fixed-dimensional vectors
  - Typically 400..600 dimensional
- Similar utterances (i.e., those from the same speaker) have similar embeddings
- Examples: i-vectors, x-vectors, ResNet-based embeddings



Weakly supervised training of DNN

- Goal: train a DNN that maps embeddings to persons
- Training data: a set of recordings, each of which consists of
  - a set of persons-of-interests that speak there
  - a set of embeddings
- However: we don't know which embedding corresponds to which speaker!
- Also, there could be more embeddings than speaker labels (since we only have labels for persons-of-interest)



Training algorithm

Data:

- N diarized speech recordings, each with
  - M_i embeddings X_{i1} .. X_{ij}
  - K_i speaker names Y_{i1} .. Y_{ik}
- Result:
  - DNN that maps embeddings to speakers

while has not converged:

- shuffle training recordings
- for n=1..N:
  - Compute DNN posterior probabilities for all embeddings X_{ij} of recording n
  - **Average** the predictions over the recording
  - Compute expected average:

$$\bar{\tilde{p}}_n(y_i) = \begin{cases} \frac{1}{|X_n|}, & \text{if } y_i \in Y_n \\ max(0, 1 - \frac{|Y_n|}{|X_n|}), & \text{if } y_i = \langle unk \rangle \\ 0, & \text{otherwise} \end{cases}$$

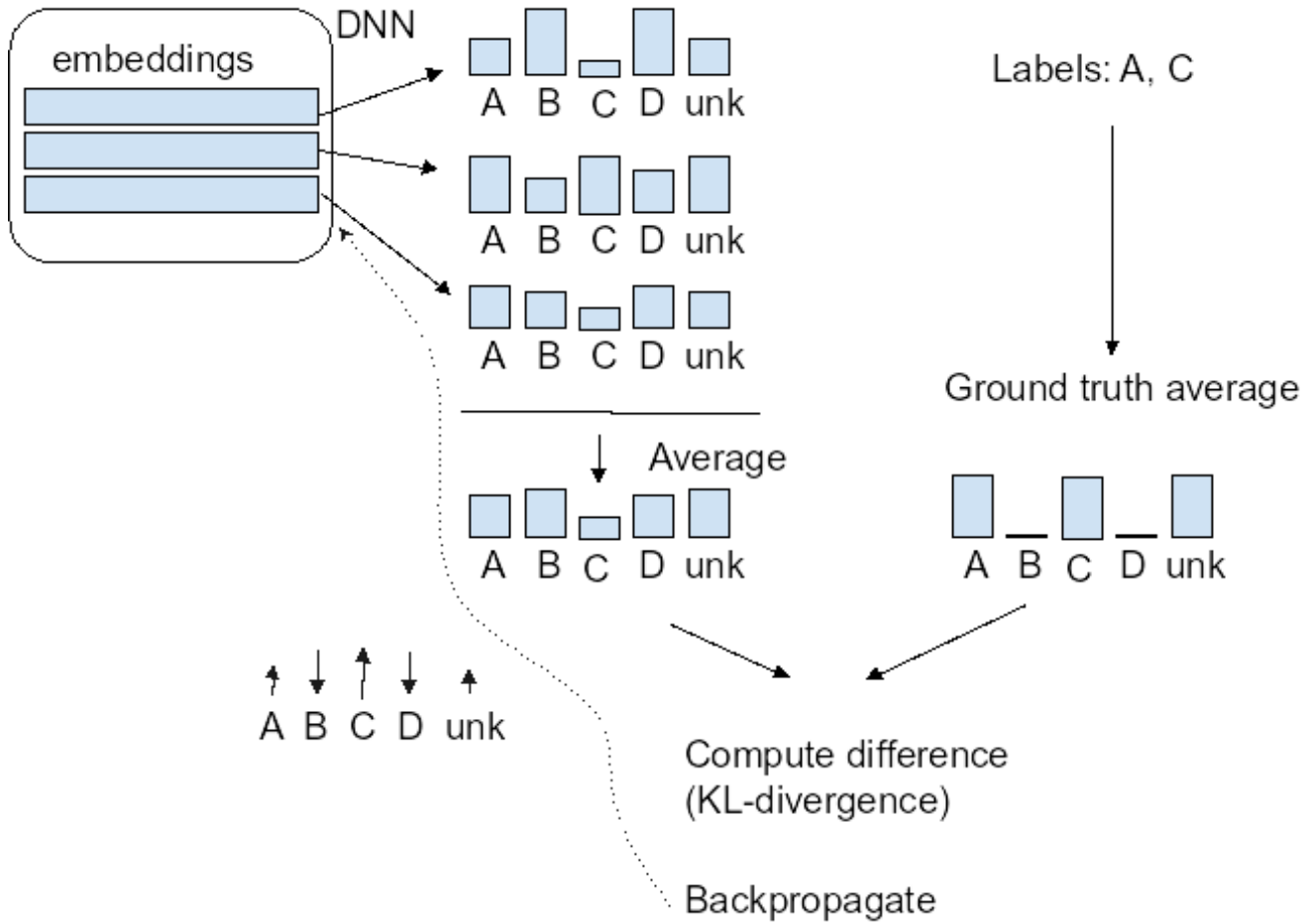
  - Compute **KL-divergence** between the predicted and expected average distributions
  - Compute gradients and update the model

Loss function

$$\bar{\tilde{p}}_n(y_i) = \begin{cases} \frac{1}{|X_n|}, & \text{if } y_i \in Y_n \\ max(0, 1 - \frac{|Y_n|}{|X_n|}), & \text{if } y_i = \langle unk \rangle \\ 0, & \text{otherwise} \end{cases}$$

- Let's say a recordings has 5 embeddings (=5 diarized speakers) and 2 speaker labels (John and Mary)
- |Xn|=5 (number of detected speakers)
- For John and Mary, the expected average posterior probability is ⅕
- For other speakers in the training set, the expected average posterior is 0
- For unknown speaker (kind of a background model), expected average is 1-⅖=⅗

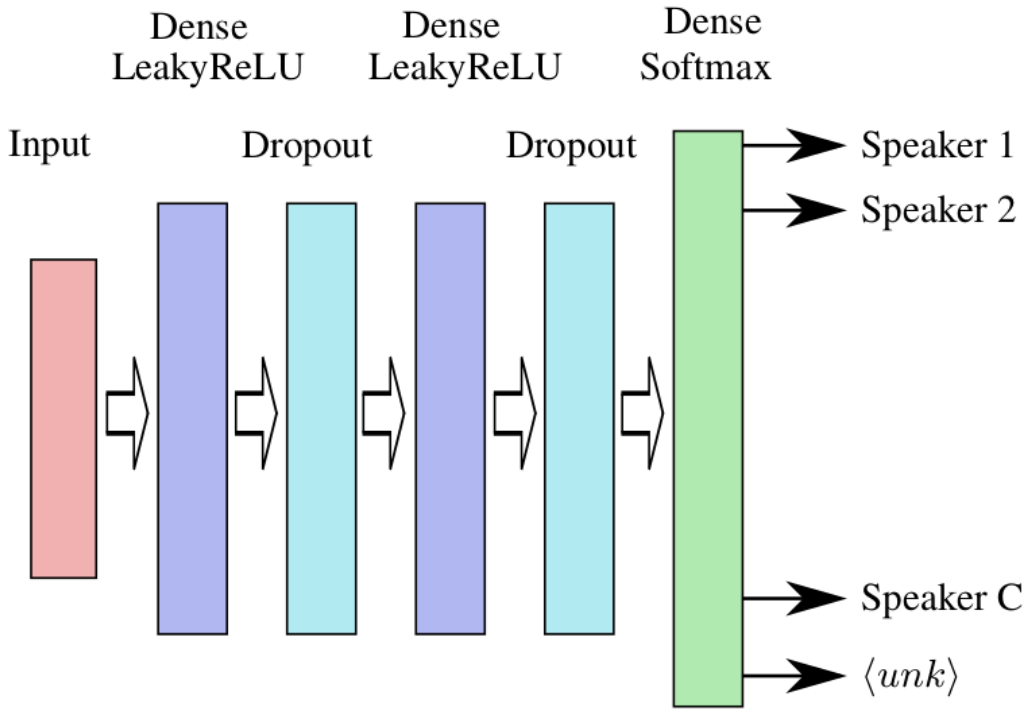
One training step example: recording with 3 speakers (from diarization), and 2 speaker labels (A, C)



DNN architecture

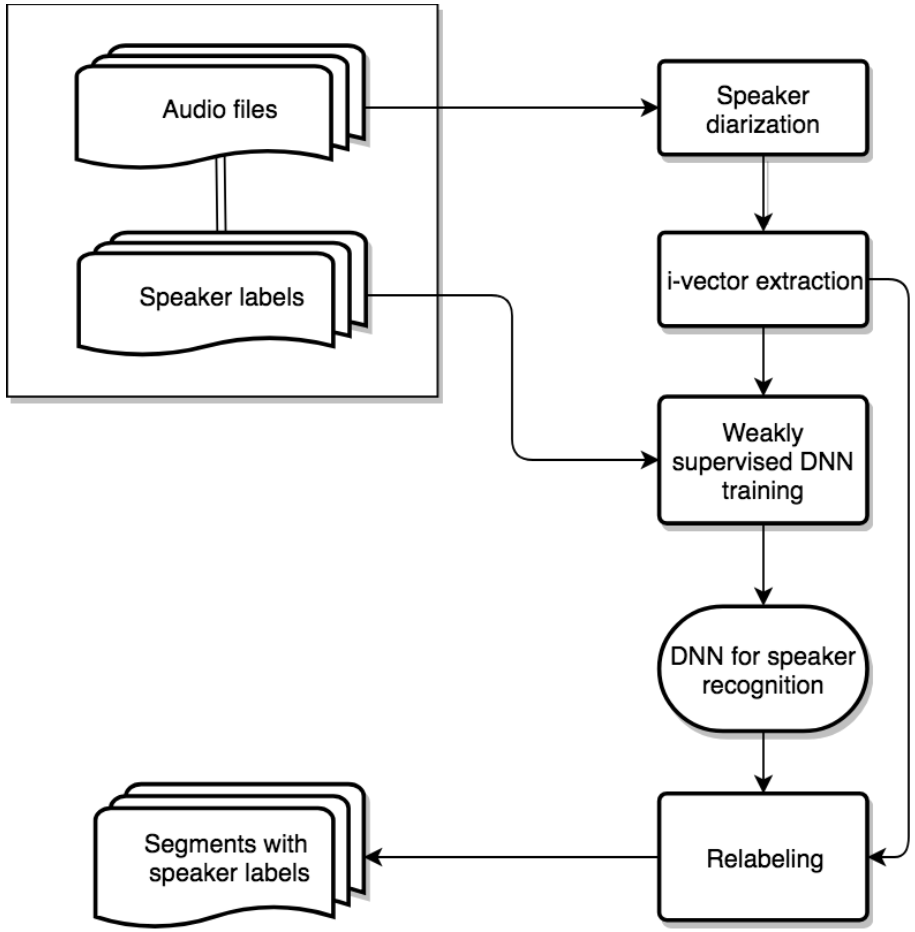
The SID DNN has a simple architecture

- Two 1024-dimensional hidden layers
- Leaky ReLU activations
- Dropout (with dropout probability 0.2) between hidden layers
- Trained for 50 epochs using a linearly decreasing learning rate



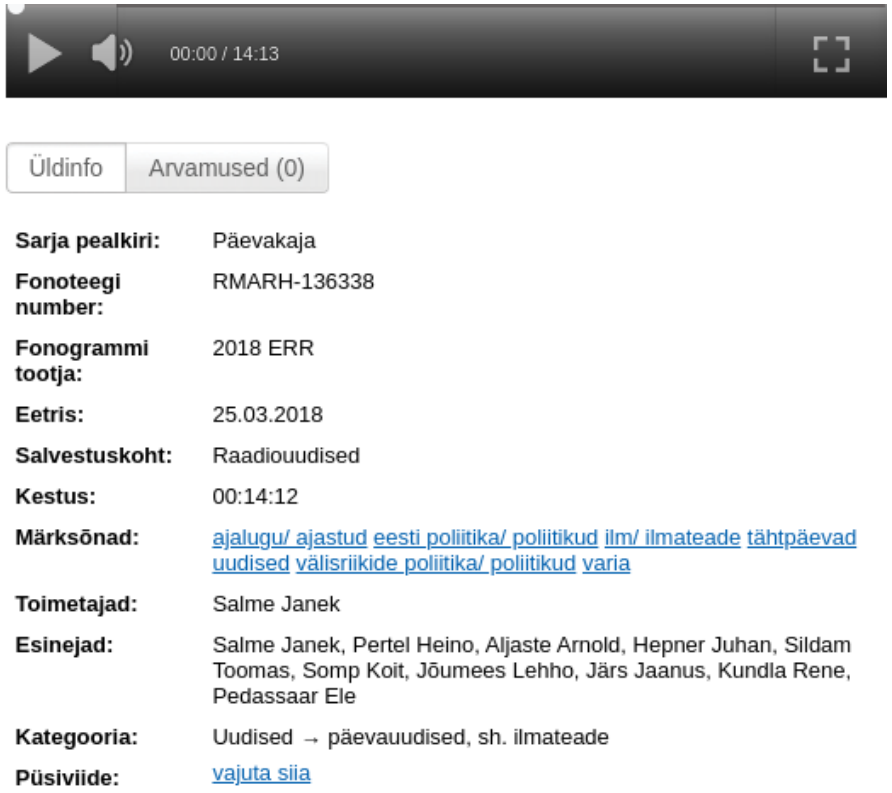
Training data relabeling

- Weakly supervised training results in a DNN that maps embeddings to persons-of-interest (or unknown person)
- This DNN can be used directly for speaker recognition
- Alternatively, it can be used to convert the original training data with recording-level labels into segmented training data
- Now, any conventional speaker recognition method can be used
- The resulting segmented training data can be used for e.g. training a speaker embedding system (x-vectors)



Experiment: Estonian Broadcast News

- The online archive of Estonian Public Broadcasting (ERR) contains news programme recordings
  - All speakers who appear in each show are manually annotated
  - No time information
- Can we use this data to train speaker recognition models for public figures?



Estonian Broadcast News: Data

Training data:

- **6619 radio news shows** (2004-2016)
- Most of the shows between 10 to 24 minutes in duration
- Accompanied with metadata, that includes a set of all speakers appearing in each show, e.g. https://arhiiv.err.ee/vaata/paevakaja-nr-20783
- 13771 unique persons
- About 5000 occur more than once
- **1529 occur five or more times**

Development and test data

- 16 shows from 2017
- Total speaking time: 2h05 (dev), 2h38 (test)
- Hand-segmented and annotated with person names
- Number of speakers: 99 (dev), 121 (test)
  - Anchors: 24 (dev), 32 (test)
  - Non-anchors: 75 (dev), 89 (test)
- Average number of speakers per show:
- 16 (dev), 19 (test)

Results: Estonian Broadcast News

- Open-set speaker ID task
- Baseline: supervised i-vector based speaker ID system trained on about 200h manually transcribed data
  - 814 speakers
  - Covers about 28% speakers in the dev set
- Proposed weakly supervised system:
  - 4939 speakers (all speakers in the weakly-supervised training data that occur more than once)
  - Covers 59% of speakers in the dev set
- Both systems use i-vector + LDA/PLDA (classical speaker recognition system)

Speaker identification accuracy, based on manually segmented dev/eval data

- Thresholds tuned to 95% precision

System	Dev		Eval	
	Precision	Recall	Precision	Recall
Supervised	100%	15%	100%	17%
Proposed	95%	41%	95%	45%

Recording-level labels for speaker recognition: summary

- Weakly supervised training allows to quickly create new speaker recognition systems cheaply and quickly
- Does not require time-consuming segment-level labeling
- Often, existing metadata about recordings can be exploited
- Experiments:
  - Works very well for speaker identification in broadcast speech

Speaker identification with LLMs

- Can we use large language models for speaker identification?
- Oleksanda Zamana's MSc thesis experiments with two approaches:
  - Using BERT-like model for classification
  - Feeding the whole recording (e.g. radio news, talkshow) transcript to GPT-4 and asking it to do speaker ID

BERT-like model for speaker identification

- Idea
  - Sometime audio-based model makes mistakes
  - Can we use the contents of the speech for making speaker ID more accurate
  - E.g., if the speaker usually speaks about health issues, then it's unlikely that a speech segment talking about computation linguistics corresponds to her
- How: let's train a BERT-like text classification model on data from the radio (which has weak speaker labels, that are mapped to segments using the weakly trained model)
- Model: XLM-RoBERTa

Results

Table 3: Speaker identification precision (P) and recall (R) rates of different models on Estonian test sets.

	News		Talkshows		Op. festival	
	P%	R%	P%	R%	P%	R%
Audio-based	98.4	71.7	94.7	64.2	96.8	26.7
Text-based	81.8	20.8	85.8	16.2	11.1	0.3
Audio + text PLDA	98.5	71.8	94.7	64.8	98.9	26.4

P = Precision: % of model's predictions (with probability exceeding a hand-tined threshold) that are correct
R = Recall: % of speakers in the recording who were identified

- Example: speech segment from radio news, interview:
  - *Algusest peale soov korraldada Hiiumaal kõrgetasemelist klassikalist muusika, festivali, mis ei ole nii-öelda ainult Hiiumaa festival, vaid selline nagu maailma mastaabis, et siia tulla tahaks kuulama ka nii-öelda välisriikidest. Teiseks, et Erkki, Sven Tüüri looming ja eriti tema orkestrilooming kindlasti kõlaks tema kodusaarel. Ja kolmandaks on siis see, et toetada Hiiumaa vanimat kirikut, pühalepa kirikut ja seal sees olevat orelit.*
- Audio:based model: This is Olari Elts (Estonian composer), probability 0.33
- Such low probability hypothesis would be rejected, based on the the audio model alone
- However, the hybrid model (that combines predictions from acoustic and text-based models) confirms that it is indeed Olari Elts

Speaker identification with LLMs

- Idea: speaker diarization + speech recognition gives us speech transcript with speaker codes
  - Speaker codes are prepended to utterances of each speaker
- Can we simply ask LLM (e.g. GPT-4) to infer the mapping from speaker codes to real names, based on the contents?
  - Assumes speakers are introduced (or introduce themselves), which usually happens on the radio
- Yes, we can, and it works really well!

Model input

You are an expert in Estonian public figures. You will be given an automatic transcription of the news or talk show, complete with speaker codes. Try to guess which persons are speaking in the program and also find the connection between the speaker's codes and names. Output the result using JSON. Example of JSON format: {"code: "name"}. If the name is unknown, write "Unknown" instead of the name.

S1: It's six o'clock, the newsroom is summarizing Sunday the twenty-fifth of October. I am Uku Toom. There is a second round of parliamentary elections in Lithuania, ...
S4: Criminals have broken into a psychotherapy center, the computer system, and obtained highly personal information about patients.
S1: A state of emergency was declared in Spain to fight the corona pandemic ...
S7: Organizing a referendum requires a decision of the Riigikogu, but
...

Model output

```
{
   "S1": "Uku Toom",
   "S4": "Unknown",
   ...
}
```

Results

- GPT4 is really good in inferring speaker names, even on Estonian!
- Best: audio + GPT4
  - Use audio-based models' predictions
  - For unidentified speakers, use GPT-4's predictions
- The main benefit: now we can also identify speakers who are not in our training data!

	News		Talkshows		Op. festival	
	P%	R%	P%	R%	P%	R%
Audio-based model	99.6	69.9	95.9	52.2	96.8	26.7
GPT 3.5 (16k)	97.1	10.6	100.0	47.3	90.7	28.4
GPT4 (128k)	97.5	71.4	100.0	97.8	97.1	69.5
Audio + GPT4	99.0	89.9	97.8	97.8	96.9	73.6

Superhuman performance

- Sometimes GPT-4 can infer full names of speakers, although the speaker was introduced using first name and full name is never mentioned
  - Probably works on persons with considerable amount of online presence
- Such wild guesses are not always correct

Transcript:
spk2: Me tuleme nende teemade juurde täna veel korduvalt tagasi. ***Tuuli****, mis on sinu lugu, sinu taust?*
spk8: Minu lugu on järgnev, et mina olen Soomes elanud üheksa aastat. Kusjuures see aeg on nii pikk, et vahepeal ma arvasin, et ma olen juba kümme aastat, aga, aga siis mind jälle, jälle veidi parandati.
spk8: Et mina olen seene klassikaline aktivistilapse stereotüüp, tegelikult. Ma olen kasvanud seltsides. Mul on kogu aeg igas olukorras öeldud, et, et. Vabandust. Meid on nii vähe, et lihtsalt ütle, mida sa teha tahad ja sa saad selle ära teha ja me toetame sind.
spk8: Ja kui sa kasvad üles sellises toetavas keskkonnas, siis see annab väga palju julgust juurde ning hakkaski piiride kompamine, et kui palju ma siis päriselt saan Eesti eest ära teha, ise mitte seal elades.
spk8: Et aastal kaks tuhat üheksateist olin ma siis, sain selle au olla üleilmsete Eesti kultuuripäevade noortetegevuste juht.
spk8: Seal oli meil kaasas kuuskümmend kuus noordelegaati ning puhtast noorte motivatsioonist ja sellest sünergiast, mis seal tekkis, valmis nende ööde jooksul tegelikult noortedeklaratsioon.
spk8: Millest kasvas siis välja ülemaailmne Eesti noorte võrgustik ja see on see, mida ma täna siin esindan.
...

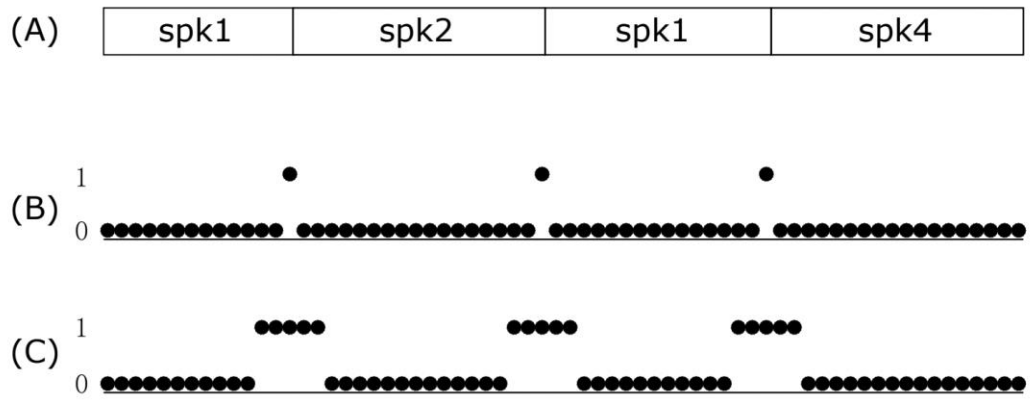
GPT4:
"spk8": "Tuuli-Emily Liivat"

Collar-aware speaker change detection

- Speaker change detection (SCD) is the task of locating precise points in the audio recording when a different speaker starts speaking
- Often is the first step of speaker diarization systems
- Postulated as a sequence labelling problem – each frame labelled as 0 (no speaker change) or 1 (speaker change)
- Data is very imbalanced with less than 1% of frames having a speaker change point
- Annotated change points are vague due to silences between speaker turns

Background

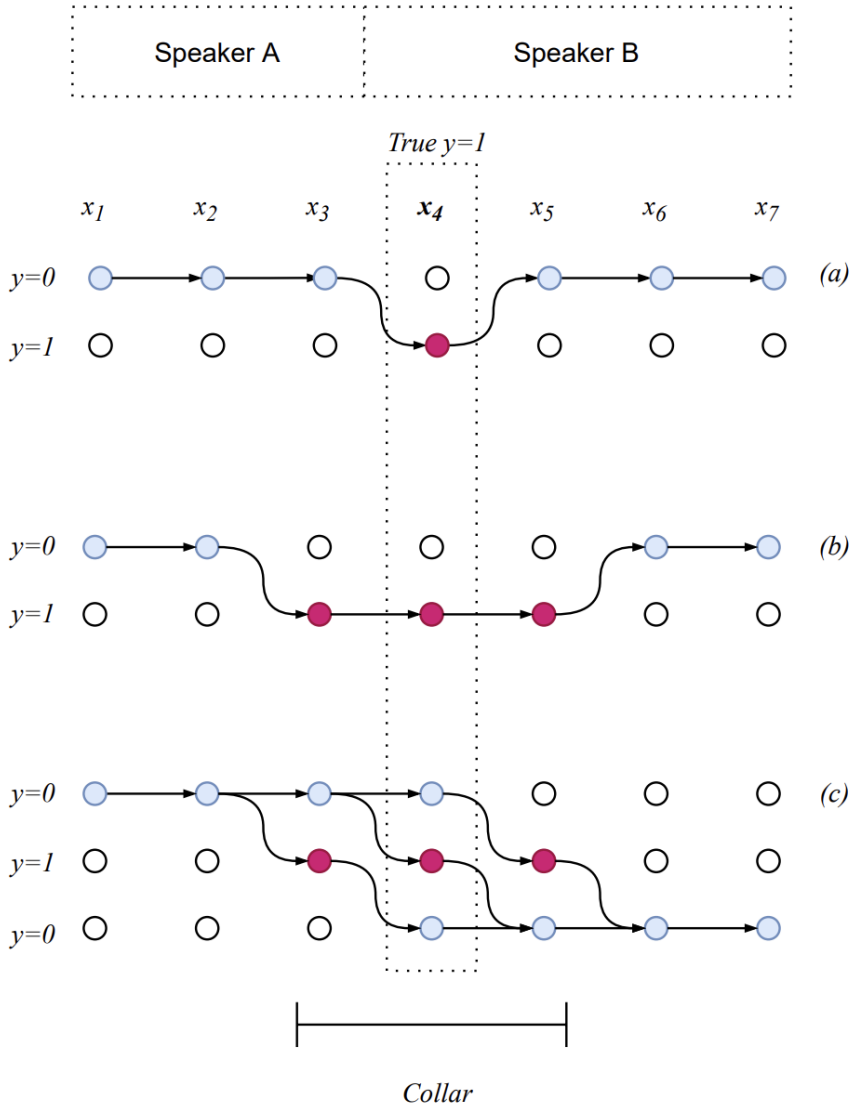
- Performance is evaluated using precision and recall with a tolerance collar (usually 250ms).
- Standard binary cross-entropy is used as the loss function for neural networks but has no relation to the tolerance collar.
- Conventional training method to tackle label imbalance is to artificially label neighborhoods around change points as positive.



Collar-aware training

We introduce a sophisticated loss function to deal with label imbalance and vagueness

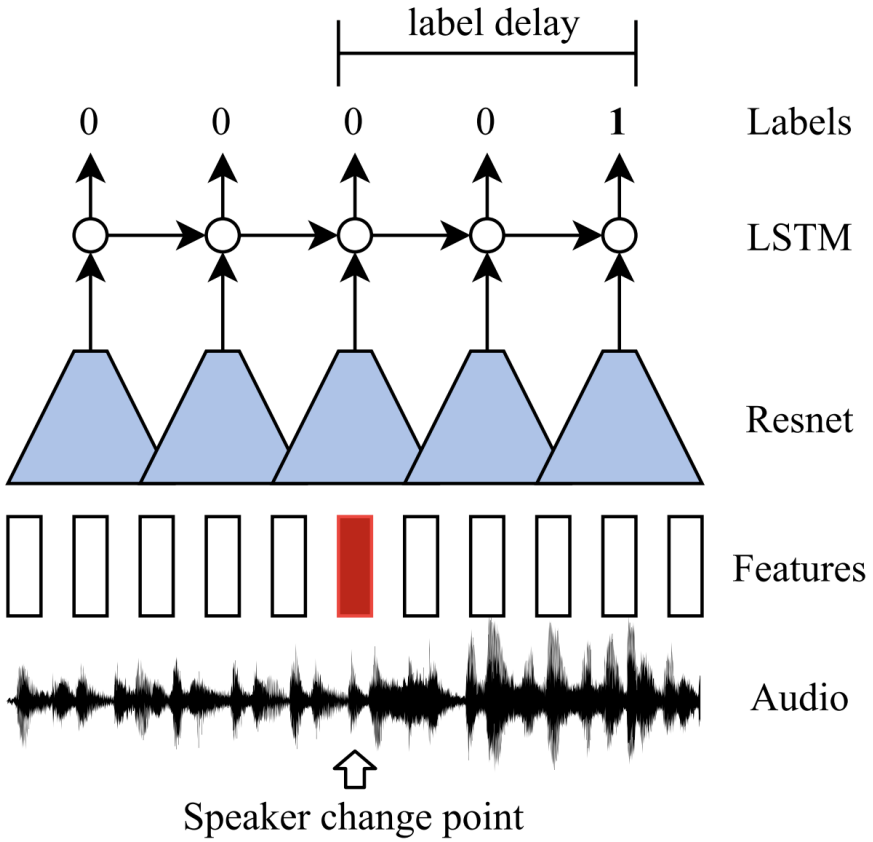
- Easiest to understand visually via comparison to the standard training method
- Marginalize over all possible subsequences that have one positive label inside the collar
- Encourages the model to predict a single positive frame in a specified collar



Model

- The first architecture was chosen to resemble [1]. MFCC-based features are fed into model made of two Bi-LSTM layers (64 and 40 dimensional) followed by a MLP with 40-, 10- and 1-dimensional layers. Used in offline mode only.

- The second model architecture uses a Resnet-based feature extractor followed by two 256-dimensional LSTM layers. 1-second label delay is used for model to see data past the current frame. For offline mode label delay is not used and LSTM layers are replaced with BiLSTM ones.

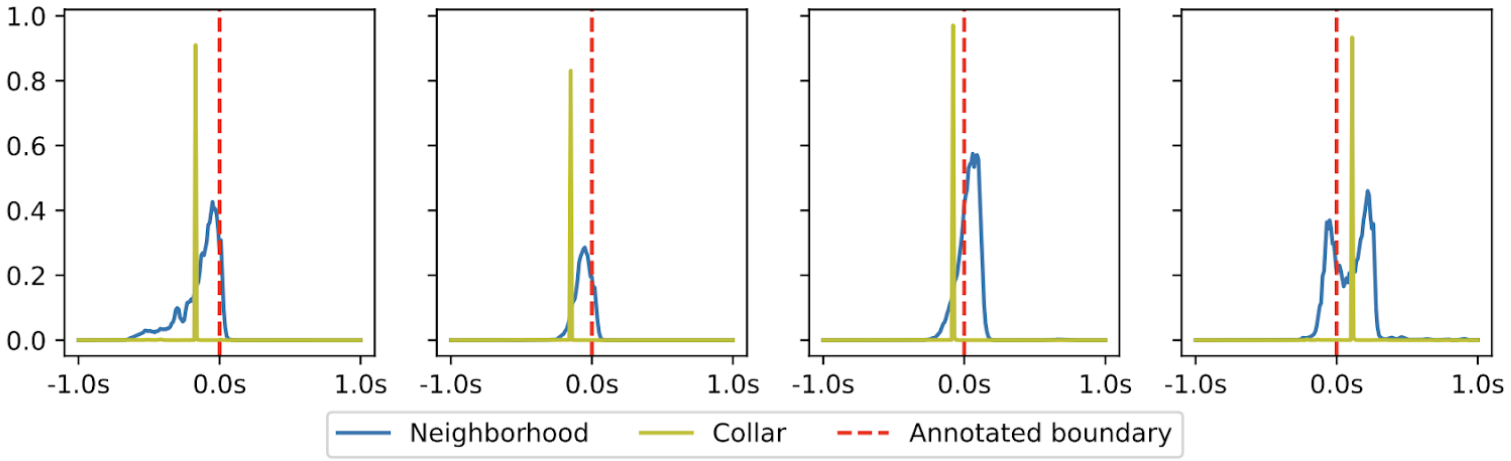


Results

	Estonian dataset						English dataset					
	collar=0.25s			collar=0.50s			collar=0.25s			collar=0.50s		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Batch-mode processing												
Pretrained speaker diarization (VBx)	0.68	0.68	0.68	0.96	0.96	**0.96**	0.48	0.64	0.55	0.67	0.88	0.76
Pretrained *pyannote.audio*	0.62	0.73	0.67	0.68	0.79	0.73	0.42	0.38	0.40	0.57	0.51	0.54
+ finetuned on the given dataset	0.82	0.82	0.82	0.89	0.89	0.89	0.60	0.49	0.54	0.73	0.59	0.65
BLSTM	0.7	0.85	0.77	0.74	0.88	0.81	0.44	0.59	0.50	0.50	0.62	0.55
+ collar aware training	0.75	0.81	0.78	0.86	0.80	0.83	0.59	0.57	0.58	0.61	0.61	0.61
Resnet + BLSTM	0.80	0.78	0.79	0.84	0.80	0.82	0.59	0.66	0.62	0.65	0.69	0.67
+ collar aware training	0.92	0.89	**0.91**	0.96	0.92	0.94	0.76	0.69	**0.73**	0.79	0.76	**0.78**
Streaming processing												
pyannote.audio with latency=1.0s	0.34	0.67	0.45	0.37	0.73	0.49	0.21	0.33	0.26	0.28	0.44	0.34
+ finetuned on our data	0.42	0.68	0.51	0.46	0.75	0.57	0.26	0.45	0.32	0.30	0.52	0.38
Resnet + LSTM	0.73	0.73	0.73	0.76	0.75	0.76	0.56	0.62	0.59	0.58	0.71	0.64
+ collar aware training	0.89	0.83	**0.86**	0.92	0.86	**0.89**	0.66	0.71	**0.68**	0.72	0.75	**0.74**

Analysis

- The outputs of neighborhood-based models are spread out over multiple frames requiring finding exact local maximum in post-processing
- Collar-aware training results in peaky outputs which do not require post-processing



Experiments

- Estonian - dataset of TV and radio broadcasts
- English - HUB4 speech dataset
  Datasets are similarly balanced with 0.04% being labelled as speaker boundaries.
- 10 recordings chosen for both test and development datasets at random.

Table 1: A comparison of lengths and the number of speaker change points in the datasets used in the experiments.

Dataset	Train	Development	Test
Estonian	497.2h / 80k	1.2h / 166	0.7h / 102
English	128.3h / 19.5k	6.1h / 893	5.4h / 893

Spoken language identification

You have already learned that spoken language identification models are very accurate on native speech
On non-native accented speech, accuracy drops dramatically
Error rates:

	VL107 dev	Estonian L1	Arabic L1	Mixed L1	Russian L1	Estonian L2	African French	Arabic L2	Mixed L2	Russian L2	Mixed L2
Wav2vec2.0-BERT	4.3	0.6	1.6	0.0	27.4	38.1	49.3	56.7	21.2	72.2	60.4

Idea: use speech recognition + text-based LID

When we hear non-native speech with a strong accent, we can often identify the language based on words and their combinations
Can we use the same approach for spoken LID?
E.g., to identify Estonian, run Estonian ASR system and check whether the output looks somewhat like Estonian
But in order to identify 100 language, we would have to run 100 ASR systems for each language
This gets computationally very expensive + need to develop many ASR models
Is there a shortcut?

Zero-shot MMS model

Zhao, Jinming, Vineel Pratap, and Michael Auli. "Scaling a simple approach to zero-shot speech recognition." ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2025.

Idea: let's train a multilingual encoder-only CTC model using a very simplified output vocabulary

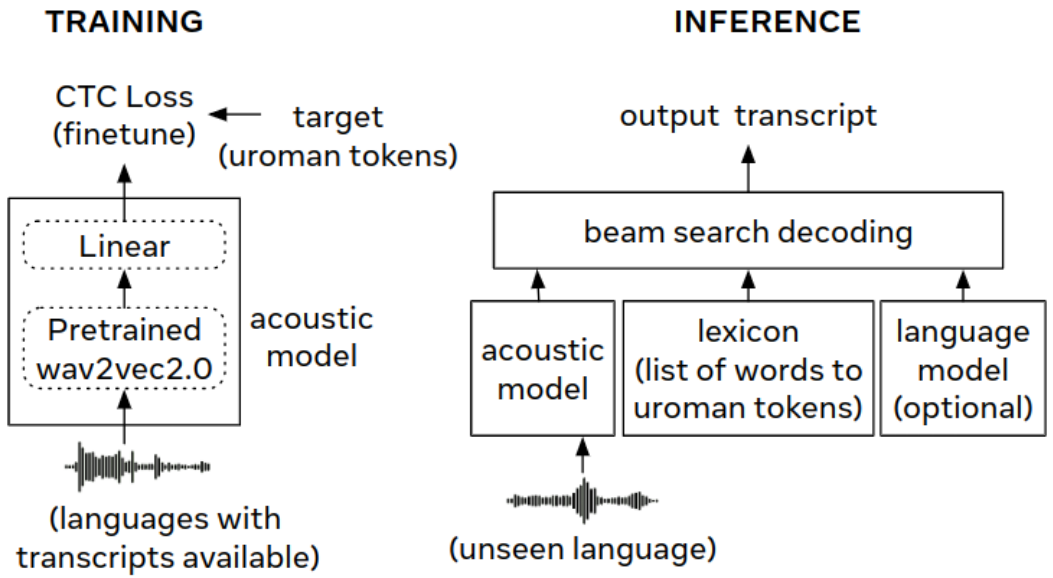
- So-called *uroman* characters: basic romanized (ASCII) letters
- Training data transcripts for all languages converted to *uroman* using the uroman library that covers almost all written languages
  - uroman.romanize_string("Tere päevast!", "est") -> 'Tere paeevast!'
  - uroman.romanize_string("Ντέιβις Καπ", "ell") -> 'Deivis Kap'
  - uroman.romanize_string("নমস্কাৰ", "asm") -> 'namaskaar'

Zero-shot MMS model

But MMS model can actually produce transcripts in the original script!
How:

- Use a language model of words (written in the original script)
- And a lexicon that maps words to *uroman "*pronunciation"
- Beam search combines CTC probabilities from MMS model with the LM and lexicon

Why important: now, given a multilingual *uroman* model, we can transcribe any language, given **only text data** and uromanizer

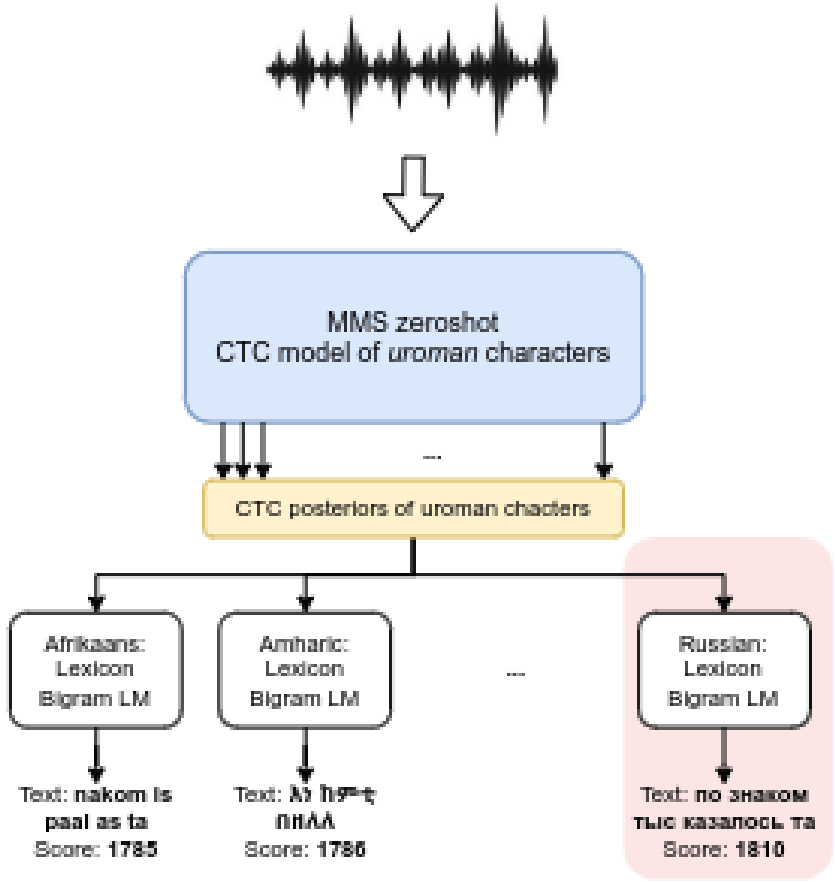


MMS zeroshot as LID system

It turns out MMS-zeroshot model, combined with LMs and lexicons in different languages, acts as a very decent language ID system

Method:

- Generate CTC posteriors using MMS-zeroshot model (requires GPU)
- Decode to words using each language's LM+lexicon (fast when using small LM, runs on CPU, can be parallelized across CPUs)
- Check, for which language the probability of the best decode is the highest



Results

LID accuracy of MMS-zeroshot is not as good as the best audio-based model
However, interpolating audio-based model' predictions with MMS-zeroshot gives nice improvements for all L2 datasets!

	VL107 dev	Arabic L1	Russian L1	Estonian L2	African French	Arabic L2	Mixed L2	Russian L2	Mixed L2
Wav2vec2.0-BERT	**4.3**	**1.6**	27.4	38.1	49.3	56.7	21.2	72.2	60.4
MMS zero-shot 10k bigram	17.0	11.5	35.3	13.9	48.9	65.1	0.9	59.3	30.9
A+B, optimized	**4.3**	**1.6**	**25.8**	**13.1**	**34.4**	**51.6**	**0.9**	**56.9**	**29.4**

THAT'S ALL!