**Diarization und Overlap Detection für Arzt-Patienten-Gespräche**

# Gliederung

1. Einleitung
   1. Motivation
   2. Aufgabenstellung
   3. Zielsetzung und Abgrenzung (Überschneidung mit 1.2?)
   4. Aufbau der Arbeit

* Paperübersicht, Modellübersicht?
  + Auswahlverfahren
* Beschreibung des Paper Ansatz Horiguchi
* Verwendete Corpora AMI, ICSI
  + Beschreibung der Daten
  + Wenn die medizinischen: wie annotieren?
* Zieldaten/-domäne
  + Gespräche im Raum
  + Wahrscheinlich nicht im Öffentlichen
  + Wahrscheinlich nur zwei Sprechende, möglich dass mehr
    - Auswahl (Vorgabe?) der Trainingscorpora AMI, ICSI
* Vorgehensweise: Corpora in das Modell einarbeiten
* Grundlagen der automatischen Spracherkennung? Diarization, Segmentation, Sprecheridentitäten, GMM, HMM, SAD/VAD, MFCC
  + 🡪 typische Herangehensweise?
* Evaluation:
  + Standard mit DER (JER)
  + Diskussion über spezielle Anforderungen:
    - Gesonderte Betrachtung von Overlap
    - Erwischt man alle Sprecher
    - Welche Wort-/Satzkategorien
    - Backchannel oder „echtes“ Ins-Wort-Fallen
* Onlinefähigkeit?
* Speaker Change? Overlap Detection?
* Fokus auf Overlap?
* Technischer Kontext:
  + Kaldi: Aufbau? Egs? Recipes?
  + Python
  + Serverdaten?
* Ergebnisse
  + DER-Tabellen, andere Evaluations-Tabellen, Berechnungsdauer
  + Baseline
* Weiterführende Arbeiten
  + Kompletten Korpus annotieren
  + (Hyperparameteroptimierung 🡪 wird wrsl zum Teil schon automatisch gemacht 🡪 ja wird mit adam)
* Fazit

BAs von school of engineering: Talkalyzer und co

Paperauswahl konkretisieren, technische Begründung der Auswahl bezüglich der Ziele?

1. Einleitung
   1. Motivation
   2. Aufgabenstellung
   3. Zielsetzung und Abgrenzung (Überschneidung mit 1.2?)
   4. Aufbau der Arbeit

* Paperübersicht, Modellübersicht?
  + Auswahlverfahren
* Beschreibung des Paper Ansatz Horiguchi
* Verwendete Corpora AMI, ICSI
  + Beschreibung der Daten
  + Wenn die medizinischen: wie annotieren?
* Zieldaten/-domäne
  + Gespräche im Raum
  + Wahrscheinlich nicht im Öffentlichen
  + Wahrscheinlich nur zwei Sprechende, möglich dass mehr
    - Auswahl (Vorgabe?) der Trainingscorpora AMI, ICSI
* Vorgehensweise: Corpora in das Modell einarbeiten
* Grundlagen der automatischen Spracherkennung? Diarization, Segmentation, Sprecheridentitäten, GMM, HMM, SAD/VAD, MFCC
  + 🡪 typische Herangehensweise?
* Evaluation:
  + Standard mit DER (JER)
  + Diskussion über spezielle Anforderungen:
    - Gesonderte Betrachtung von Overlap
    - Erwischt man alle Sprecher
    - Welche Wort-/Satzkategorien
    - Backchannel oder „echtes“ Ins-Wort-Fallen
* Onlinefähigkeit?
* Speaker Change? Overlap Detection?
* Fokus auf Overlap?
* Technischer Kontext:
  + Kaldi: Aufbau? Egs? Recipes?
  + Python
  + Serverdaten?
* Ergebnisse
  + DER-Tabellen, andere Evaluations-Tabellen, Berechnungsdauer
  + Baseline
* Weiterführende Arbeiten
  + Kompletten Korpus annotieren
  + (Hyperparameteroptimierung 🡪 wird wrsl zum Teil schon automatisch gemacht 🡪 ja wird mit adam)
* Fazit

https://arxiv.org/pdf/2005.09921.pdf

End-to-end neural network-based speaker diarization model: **EEND (SA-EEND -> self attentive)**

* Direkt geeignet für overlapping speech, als auch Minimierung von DER
* Multi-label classification (wie SED)
  + Permutationsfrei
* Vorteile:
  + Kann implizit mit overlap umgehen
  + Keine Module notwendig für:
    - SAD
    - Speaker identification
    - Source seperation
    - clustering

FEATURES: T-length log-scaled Mel-filterbank

* Bi-directional long short-term memory **BLSTM OR**
* **Tranformer Encoder**
* Embedding of each T-length slot:
* Linear tranformation : element-wise sigmoid 🡪 **Posteriors**

**of S speakers at time slot t** (was bedeutet die Notation: kommaseparierte Werte in (eckigen) Klammern hoch S? These: Tupelschreibweise, aber warum unterschiedlich [ ] und ( ) 🡪 **TODO** nachprüfen)

Training phase: optimized using PIT 🡪 loss is calculated between posterior and labels with

with , perm = set of all possible permutations of speakers with binary cross entropy: binary cross entropy:

(^Latexschreibweise haut wohl in Word nicht 100% hin 🡪 unbekannte commands?)

EEND only works with fixed number of speakers during inference

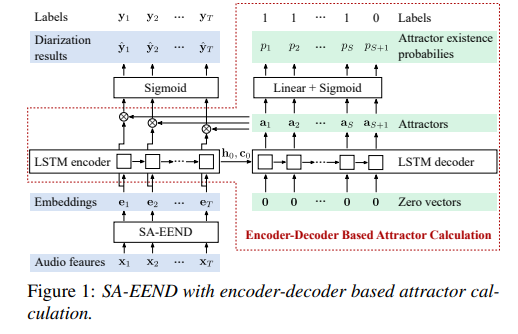
* End-to-end trainable 🡪 Encoder-Decoder based Attractor **EDA** 🡪 **determining attractors from embedding sequence**
* With **LSTM**-based encoder-decoder
  + **Unidirectional LSTM encoder**
  + **Unidirectional LSTM decoder**
  + Theoretically infinte numbers of attractors can be calculated 🡪 when to stop? via fully connected layer with sigmoid function
  + ! output attractors depend on order of input embeddings
    - Shuffle = one specific permutation
* Training phase
  + groundtruth labels: with (heißt das: 4 speaker 🡪 l = [1, 1, 1, 1, 0] ?)
  + attractor existence loss via binary cross entropy: Between label and estimated probabiliy (heißt das:

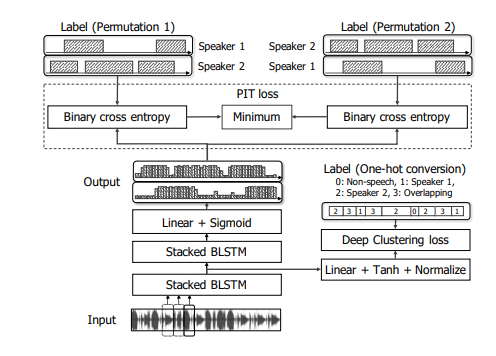
345-dim log scaled Mel-filterbank feature 🡪

4x stacked SA-EEND Tanformer encoder 🡪

256 dim embeddings 🡪

EDA: attractor calc





|  |  |  |
| --- | --- | --- |
| Cvmn.csp | Cvmn from mfcc |  |
| Feats.scp | Mfcc |  |
| **Reco2file\_and\_channel** |  |  |
| **Segments** | Utterance, recording-id, start end |  |
| **Spk2utt** | Speaker to utterance |  |
| **Text** | Transcript of each utternance |  |
| Utt2spk | Utterance to speaker (from spk2utt) |  |
| **Wav.scp** | Wav files (lambdas) |  |

Text:

|  |  |
| --- | --- |
| **Utt\_id (speaker-id\_utternance\_id)** | utternance |
| sw02001-A\_000098-001156 | HI UM YEAH I'D LIKE TO TALK ABOUT HOW YOU DRESS FOR WORK AND |
| sw02001-A\_001980-002131 | UM-HUM |
| sw02001-A\_002736-002893 | AND IS |

Wav.scp:

|  |  |
| --- | --- |
| **Recording-id** | **Filename (lambda)** |
| sw02001-A | /home/dpovey/kaldi-trunk/tools/sph2pipe\_v2.5/sph2pipe -f wav -p -c 1 /export/corpora3/LDC/LDC97S62/swb1/sw02001.sph | |

Segments:

|  |  |  |  |
| --- | --- | --- | --- |
| **Utt-id** | **Recording-id** | **Segment-begin** | **Segment-end** |
| sw02001-A\_000098-001156 | sw02001-A | 0.98 | 11.56 |

Utt2spk:

|  |  |
| --- | --- |
| **utterance-id** | **Speaker-id** |
| sw02001-A\_000098-001156 | 2001-A |

Everything must be sorted (export LC\_ALL=C)

**CREATED FROM THOSE FILES:**

Spk2utt:

utils/utt2spk\_to\_spk2utt.pl data/train/utt2spk > data/train/spk2utt

|  |  |  |
| --- | --- | --- |
| Speaker-id | Utterance-id1 | Utterance-i2… |
| 2001-A | sw02001-A\_000098-001156 | … |

Feats.scp

steps/make\_mfcc.sh --nj 20 --cmd "$train\_cmd" data/train exp/make\_mfcc/train $mfccdir

cvmn.scp

steps/compute\_cmvn\_stats.sh data/train exp/make\_mfcc/train $mfccdir

|  |  |
| --- | --- |
| Speaker-id | Cmvn-feats |
| 2001-A | /home/dpovey/kaldi-trunk/egs/swbd/s5/mfcc/cmvn\_train.ark:7 |

**Baseline**:

MFCC(mel-bins = 40, num\_ceps =40, f\_min = 20, f\_max = 7600 (8000 – 400, why -400) 🡪 MFCC Compute cepstral mean and variance statistics per speaker CMVN stats (norm\_means=true, norm\_vars=false) 🡪 CMVN sliding window cmn\_window=300

ASpIRE SAD Model / A TDNN trained in the egs/aspire/s5 for speech activity detection. / MODEL TYPESpeech Activity Detection (SAD), TDNN + Stats pooling / ASpIRE SAD Model 1a

x-vector extraction pretrained nn 🡪

plda

**[STAGE 3]**

**make\_mfcc**: mel-bins: 40, mel\_ceps: 40, f\_min=20, f\_max=7600, data/{train|valid|dev}

steps/make\_mfcc.sh --mfcc-config conf/mfcc\_hires.conf --nj $nj --cmd "$train\_cmd" data/$dataset

* mfcc to data/$dataset/data 🡪 splitting segments for parallel –> extract segments, compute mfcc 🡪 frame\_shift = 0.01 = 10ms

compute-mfcc-feats.cc

* MFCC over wav.scp: MFCC per utt

**compute\_cmvn\_stats**: compute cepstral mean (and variance statistics) per speaker

data/{train|valid|dev}

steps/compute\_cmvn\_stats.sh data/$dataset

require: feats.scp, spk2utt

cmvn-dir = data/train/data

compute-cmvn-stats --spk2utt=ark:$data/spk2utt scp:$data/feats.scp ark,scp:$cmvndir/cmvn\_$name.ark,$cmvndir/cmvn\_$name.scp

compute-cmvn-stats.cc

cmvn.scp: speaker\_id | data/cmvn\_train.ark:#####

cmvn\_train.ark: speaker\_id | mean?

SEEMS like mean / speaker\_id

accumulating cmvn stats for utterances

**[STAGE 4] Preparing AMI training data for PLDA model**

Local/nnet3/xvector/prepare\_feats.sh -jobs -cmd data/train data/plda\_train exp/plda\_train\_cmn

**prepare\_feats:** applies sliding window CMVN and writes features to disk

In-data-dir = data train; out-data-dir = data/plda\_train; feat-dir=exp/plda\_train\_cmn

Cmn-window = 300

apply-cmvn-sliding --norm-vars=false --center=true --cmn-window=$cmn\_window \

  scp:${sdata\_in}/JOB/feats.scp ark:- \|

<https://ozlemkalinli.com/PCMN_2019.pdf>

**apply-cmvn-sliding**: apply sliding window cepstral mean normalization per **UTTERANCE**

data-out/ **feats.scp** + exp/plda\_train\_cmn/xvector\_cmvn\_feats:\*: **= MEAN NORMALIZED**

**[STAGE 5] Extracting x-vector for PLDA training data**

 diarization/nnet3/xvector/extract\_xvectors.sh --cmd "$train\_cmd --mem 10G" \

    --nj $nj --window 3.0 --period 10.0 --min-segment 1.5 --apply-cmn false \

    --hard-min true $model\_dir \

    data/plda\_train $model\_dir/xvectors\_plda\_train

extract\_xvectors.sh: assumes trained x-vector DNN, segments file, features. Segment files created by SAD for speech segments. This script does: **SUBSEGMENTATION**: splits speech segments into very short overlapping subsegments (1.5s, 0.75s overlap). X-vectors are extracted for each subsegment.

**AFTER**: PLDA computes similarity between all pairs of x-vectors in recording

Window=3.0s (sliding window length), period=10.0s (period of sliding windows), min\_segment=1.5s (minimum segment length per xvector), **apply-cmn=false**(apply sliding window cepstral mean norm to features), hard-min=true (removes segments less than min-segment),

src\_dir = model\_dir=exp/xvector\_nnet\_1a

data = data/plda\_train

dir = exp/xvector\_nnet\_1a/xvectors\_plda\_train

nnet = nnet3-copy –nnet-config=[output-node name=output input=tdnn6.affine]

nnet final.raw = x-vector DNN, trained on Voxceleb with augmentation (reverberation from simulated room impulse responses)

The x-vector architecture:

@inproceedings{snyder2018xvector,

title={X-vectors: Robust DNN Embeddings for Speaker Recognition},

author={Snyder, D. and Garcia-Romero, D. and Sell, G. and Povey, D. and Khudanpur, S.},

booktitle={2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)},

year={2018},

organization={IEEE},

url={http://www.danielpovey.com/files/2018\_icassp\_xvectors.pdf}

}

Diarization with x-vectors:

@article{sell2018dihard,

title={Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge},

author={Sell, G. and Snyder, D. and McCree, A. and Garcia-Romero, D. and Villalba, J. and Maciejewski, M. and Manohar, V. and Dehak, N. and Povey, D. and Watanabe, S. and Khudanpur, J.},

journal={Interspeech},

year={2018}

}

Nnet from <https://kaldi-asr.org/models/m12>, CHIME 6 Baseline Diarization system, models/12/0012\_diarization\_v1.tar.gz

**[[stage 0]] sliding window subsegments:**

segments=pruned\_segments (only min. 1.5s)

**get\_uniform\_subsegments.py** = overlap-duration = window - period = **-7s**, max-segsment-duration=3.0s, max-remaining-duration=1.5s, constant-duration=True

alle 10s 3s segmente, wenn weniger als 10s, dann 3s bis max, wenn weniger als 3s bei periode, dann rest, also:

0 – 3; 10 – 12,3 ||| 0-3; 10-13; 16,3-19,3

Subsegment file:

|  |  |  |  |
| --- | --- | --- | --- |
| **Subsegment-id = {**  **Utt\_id-rel\_start(8digits)-rel\_end(8digits)**  **}** | **Utterance-id** | **Start-time** | **End-time** |
| EN2001a\_002809\_004005-00001000-00001196 | EN2001a\_002809\_004005 | 10.000 | 11.960 |

**Subsegment\_data\_dir.sh**: segments file + text-file, combining into new subsegmented-output directory for all files in them (like wav.scp etc):

Srcdir = data = data/plda\_train, subsegments-file=exp/xvector\_nnet\_1a/xvectors\_plda\_train/subsegments, destdir=subdata=exp/xvector\_nnet\_1a/xvectors\_plda\_train/subsegments\_data

**[[stage 1]] xvector extractions**

nnet3-xvector-compute --use-gpu=no --min-chunk-size=$min\_chunk\_size --chunk-size=$chunk\_size \

      "$nnet" "$feats" ark,scp:${dir}/xvector.JOB.ark,${dir}/xvector.JOB.scp

**[[stage 2]]** **combining xvectors across jobs**

* dir = exp/xvector\_nnet\_1a/xvectors\_plda\_train/xvector.scp

**[[stage 3]] computing mean of xvectors**

ivector-mean scp:$dir/xvector.scp $dir/mean.vec

dir/mean.vec

**[[stage 4]] computing whitening transform (if true)**

**[STAGE 6] train PLDA models**

Compute the mean vector for centering the evaluation xvectors: already happened? Mean.vec

**TRAIN PLDA model**

ivector-compute-plda ark:$model\_dir/xvectors\_plda\_train/spk2utt \

    "ark:ivector-subtract-global-mean scp:$model\_dir/xvectors\_plda\_train/xvector.scp ark:- |\

     transform-vec $model\_dir/xvectors\_plda\_train/transform.mat ark:- ark:- |\

      ivector-normalize-length ark:-  ark:- |" \

    $model\_dir/xvectors\_plda\_train/plda

Ivector-compute-plda.cc: computes a plda object from a set of vectors. Uses speaker information from a sp2utt file to compute within and between class variances

Usage: ivector-compute-plda [oitions] spk2utt-rspecifier ivector-rspecifier plda-out

**[STAGE 7] evaluation**

**[OVERVIEW: DATASETS and intermediate results]**

test\_sets=dev test

Train\_set=train

model\_dir=exp/xvector\_nnet\_1a

**STAGE 2:** creating data sets

Data/dev

Data/test

Data/train

**STAGE 3:** feature extraction

Data/{dev,test,train} 🡨 steps/make\_mfcc.sh 🡪 feats.scp

Data/{dev,test,train} 🡨 steps/compute\_cmvn\_stats.sh 🡪 cmvn.scp

**STAGE 4:** prepare features for xvector extraction (cmvn over feats)

exp/plda\_train\_cmn 🡨 local/nnet3/xvector/prepare\_feats.sh data/train 🡪intermediate results

Data/plda\_train 🡨 combining exp/plda\_train\_cmn into 🡪 feats.scp

**STAGE 5:** extracting embeddings as xvectors from nnet

exp/xvector\_nnet\_1a/xvectors\_plda\_train 🡨 diarization/nnet3/xvector/extractr\_xvectors.sh(data/plda\_train) 🡪 parallel extraction: xvector.{#}.{scp,arc}

exp/xvector\_nnet\_1a/xvectors\_plda\_train 🡨 combining xvector.{#}.{scp} 🡪 xvector.scp

**STAGE 6:** train PLDA

exp/xvector\_nnet\_1a/xvectors\_plda\_train 🡨 ivector-mean(xvector.scp) 🡪 mean.vec

[ivector-subtract-global-mean xvector.scp] (no intermediate result)

exp/xvector\_nnet\_1a/xvectors\_plda\_train 🡨 intermediate\_result of transform-vec 🡪 transform.mat

[ivector-normalize-length transform.mat] (no intermediate result)

exp/xvector\_nnet\_1a/xvectors\_plda\_train 🡨 ivector-compute-plda(last cached result) 🡪 plda

exp/xvector\_nnet\_1a 🡨 cp plda

exp/xvector\_nnet\_1a 🡨 cp transform.mat

exp/xvector\_nnet\_1a 🡨 cp mean.vec

**STAGE 7:** evaluation

Model\_dir = exp/xvector\_nnet\_1a

Data\_in = data/{dev,test}

Out\_dir = exp/{dev,test}\_diarization\_{spectral,vbx}

[**diarize-scripts]**

Name = {dev,test}

[**[stage 1]]**: computing features for x-vectors extractor

Data/{dev,test}\_cmn; exp/{dev,test}\_cmn

**[[stage 2]]:** extracting x-vectors

Exp/{test,dev}\_diarization\_{spectral,vbc}/xvectors\_{test,dev} 🡨 computed xvectors (no cmn, window=1.5, period=0.75, min-seg=0.5, apply-cmn=false)

**[[stage 3]]**: perform cosine similarity scoring on all pairs of segments for each recording

Diarization/score\_cossim.sh out\_dir/xvectors\_{test,dev}/cossim\_scores

**[[stage 4]]:** perform spectral clustering using cosine similarity scores

Diarization/scluster.sh 🡪 write rttm into out\_dir

**[diarize-scripts]** **end**

md-eval.pl

data/{test,dev}/rttm.annotation

exp/{test,dev}\_diarization\_{spectral,vbx}/rttm

typische:

hamming-window 🡪 features 🡪 SAD via mehrere stufen, zb : energy based detector (stille) , dann GMM based detector (nicht-sprache töne) 🡪 speaker segmentation module (z.B. Bayesian Information Criterion BIC) 🡪 clustering von segmenten gleicher sprecher (bottom – up: hierarchical agglomerative clustering) 🡪 i/d/x vectros: extraktion erzeugt gleich große segmente, durch pretrained GMM oÄ 🡪 wird in PLDA gefüttert

1. Titel ist nicht mehr zutreffend

Datenschutzbestimmungen nicht arbeiten darf

1. Deckblatt: welches abgabedatum?
2. Welche Tiefe soll alles haben? Z.B. Grundlagen, Datenbeschreibung

2 seiten: frequenz 🡪

Ab sprache, 1,2 mehr sprecher 🡪 ich möchte diarization machen 🡪 overlap 🡪 evaluieren, pro frame/zeit 🡪 klassifikation

Voxceleb

1. Kleine Beispiele hier und da zur Veranschaulichung?
2. Frist für Präsentation? Bewertungsanteil, Benotung?

||||||||||||

Frage für mich nachher: abgabetag!

141.75.89.4

|  |  |  |
| --- | --- | --- |
| Nach google 2018 icassp |  |  |
| Speech detection: VAD to remove noise and non-speech |  |  |
| Speech segmentation: extract short segments, each containing a single speaker |  |  |
| Embedding extraction: generate a compact representation for each segment (i-/d-/x-vecotr) |  |  |
| Clustering: determine number of speakers, assign each speech segment to a single speaker |  |  |
| Resegmentation (optional): smooth the diarization results |  |  |

i-/d (deep, neural networks)-/x-vectors (later end 2 end): fixed dim speaker-discriminative features from audio. Same speaker –> similar embeddings

d-vector: different ways to extract: e.g. loss function

training with end-to-end loss on 3-layer lstm

clustering: spectral clustering with eigen-gap to find number of clusters then sequence of refinement operations on affinity matrix

Hungarian Algorithm for evaluation: speaker ground truth vs hypothesis:

A B B A C

C A A C B

* equivalent

Pro/contra end to end diarization

+ ONE neural net for everything:

* speaker embedding extraction
* segmentation
* clustering
* VAD, resegmentation etc

+ loss function directly based on DER

* Permutation-invariant loss

<https://www.youtube.com/watch?v=pGkqwRPzx9U>

clustering: order of data incoming to determine clustering

Googles fully supervised dingens

Training:

Speaker change: deterministic estimation of probabilities

Speaker assignment: gradient ascent (not descent) 🡪 maximizing probabilies, not minimizing error

Sequenze generation: gradient ascent

Mel-scale

Mel = melody

Ein Bild, das Text, Himmel, Angeln enthält.

Automatisch generierte Beschreibung

Delta auf mel scale wird gleich wahrgenommen. 100 – 200 hz klingt nach einem größeren abstand als 1600 – 1700 hz. 500 – 1000 mel klingt ähnlich wie 1500 – 2000. Empirisch erarbeitet durch psychologische experimente. :

1. # mel bands 🡪 hyperparameter
2. Construct mel filter banks

Lowest/highest f that we want or data has to Mel

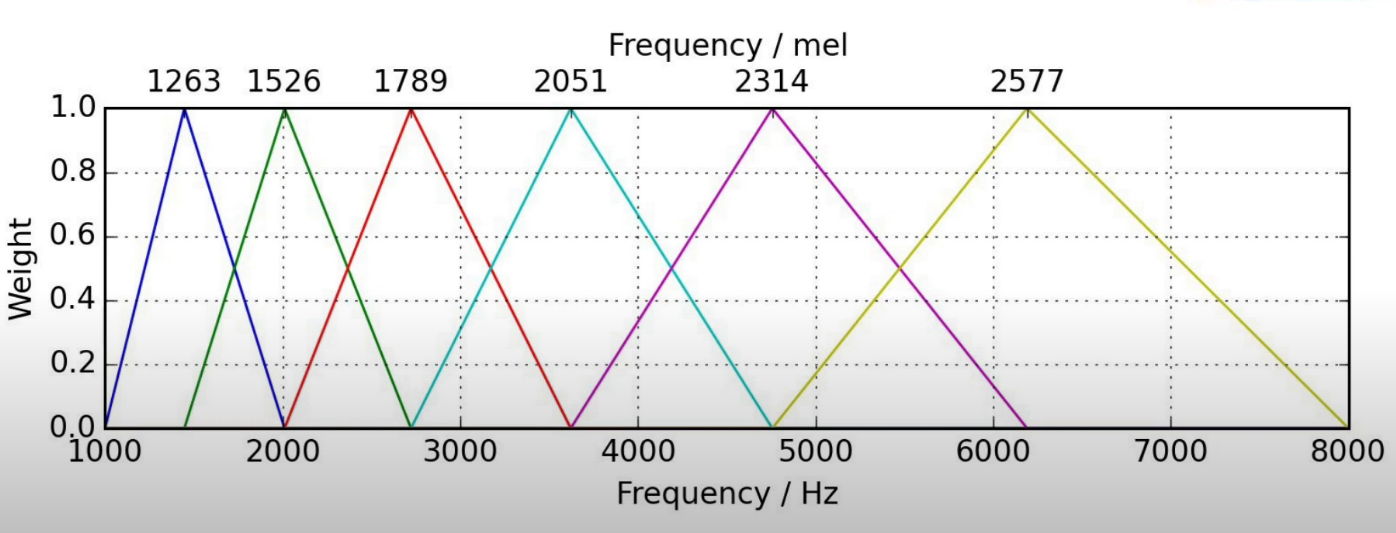
Lowest 🡪 high freq, devide into equally spaced points by #banks



Points back to Hz

Because of discrete nature round points to nearest freq bin

Make triangular filters: one mel point is peak, left is under mel peak b4, right is under mel peak next



1. Apply those banks to spectogram

Mel-matrix = (# bands, **framesize/2 + 1**)

STFT-matrix= (**framesize/2 + 1**, # frames)

Mel-matrix \* STFT-matrix = Mel-spectogram (# bands, # frames)

**Mel-frequency cepstral coefficient MFCC**

Ceps 🡪 spec backwards

Quefrency = Frequency

Liftering = Filtering

Rhamonic = Harmonic

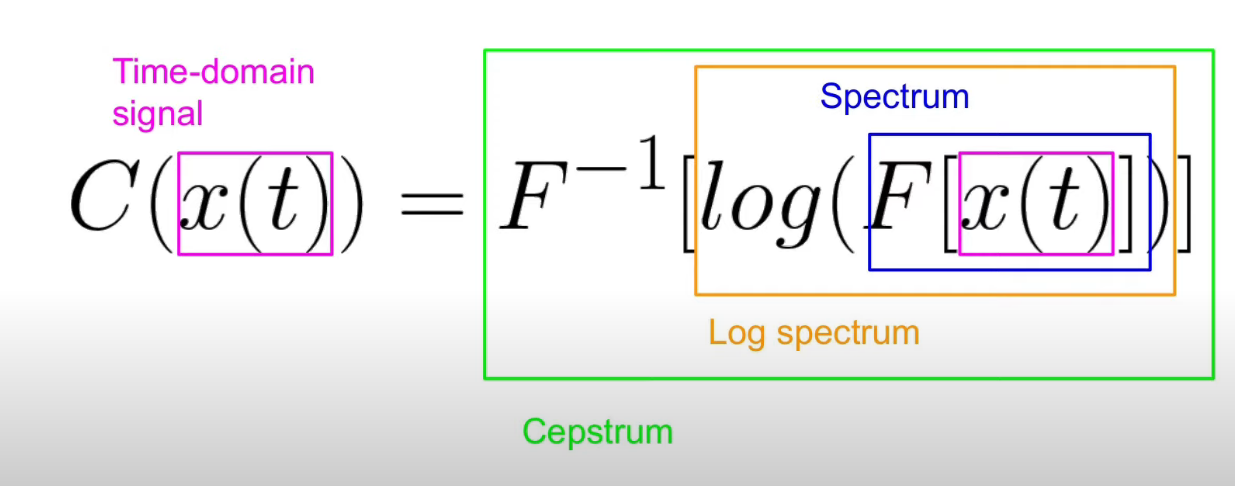
Cepstrum:

X(t) = waveform = time-domain signal

F = Spektrum = FT

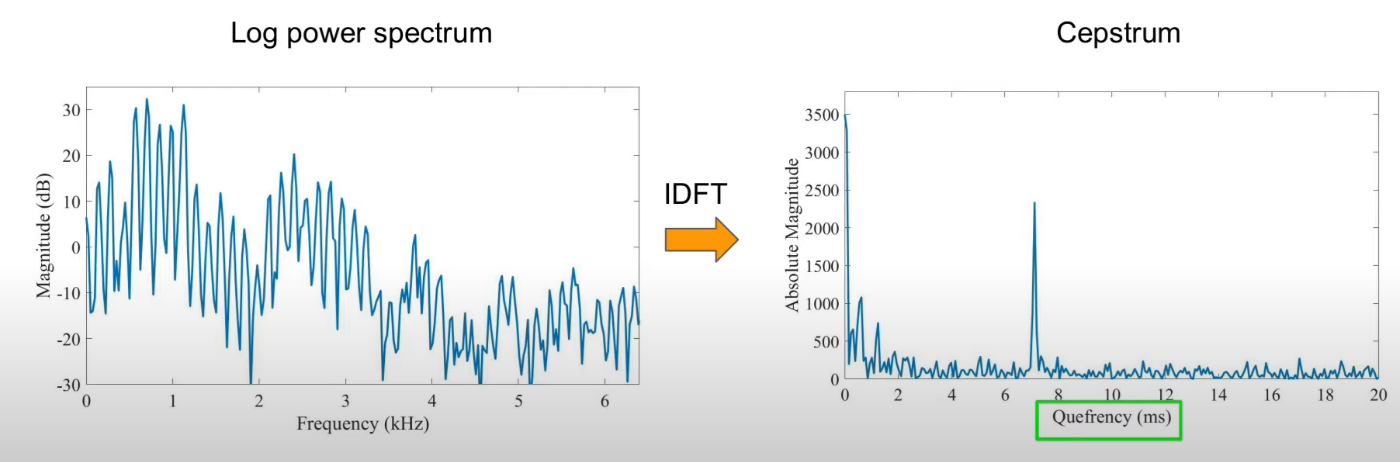
Log(F) = log-Spektrum

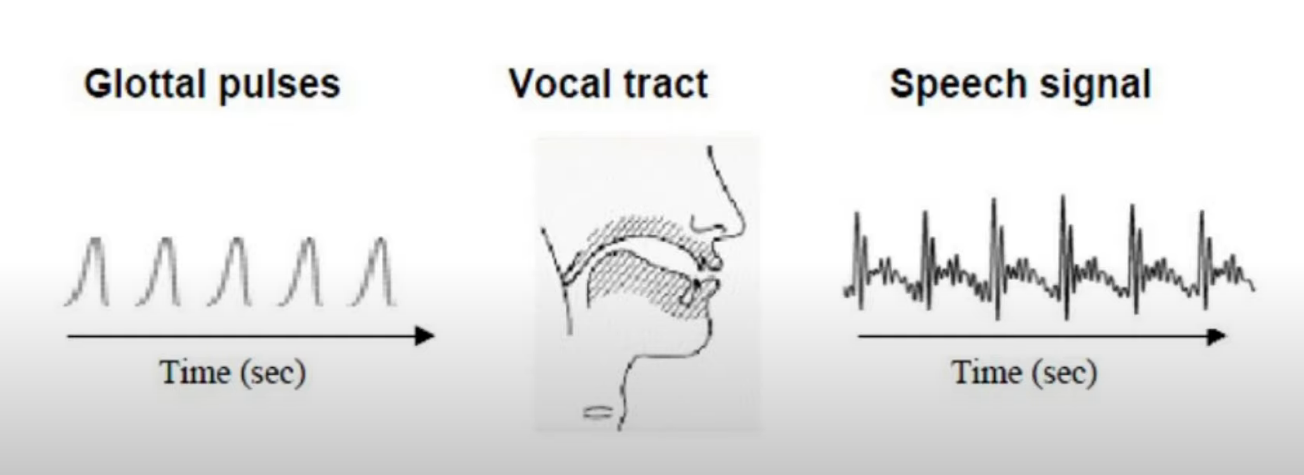
F-1 = Cepstrum = inverse FT zum log-Spektrum

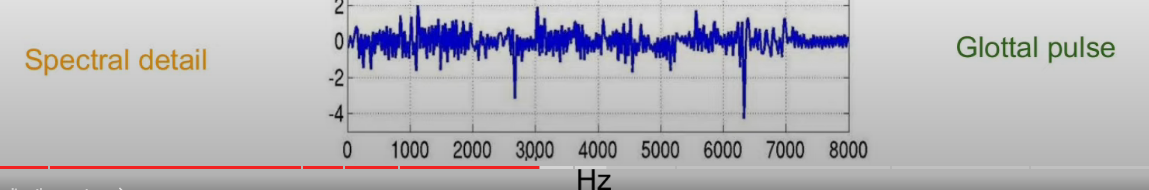
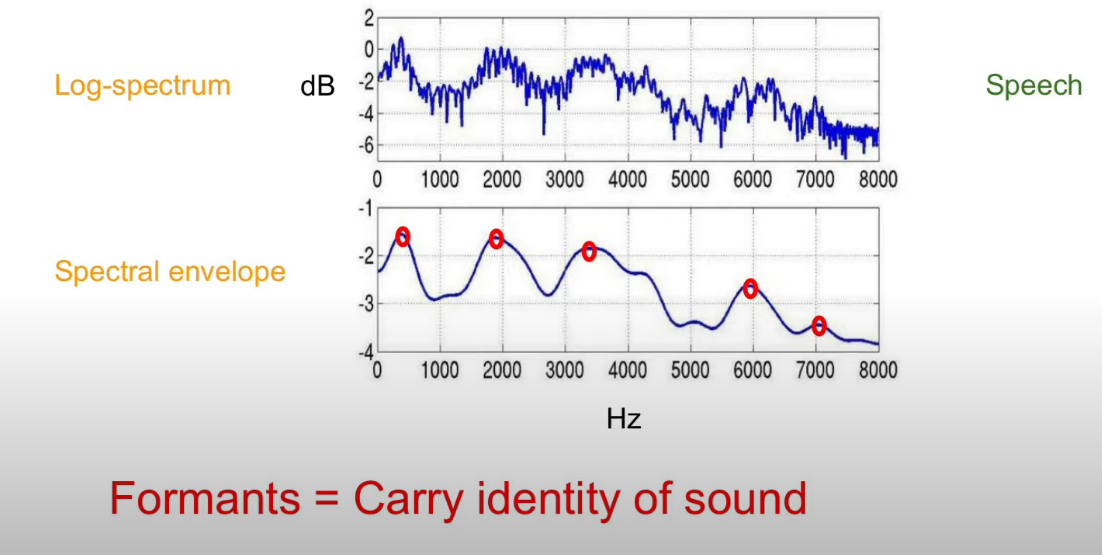


Power spectrum (frequenz diagramm nach dft)

Log of power/pegel/amplitude = dB





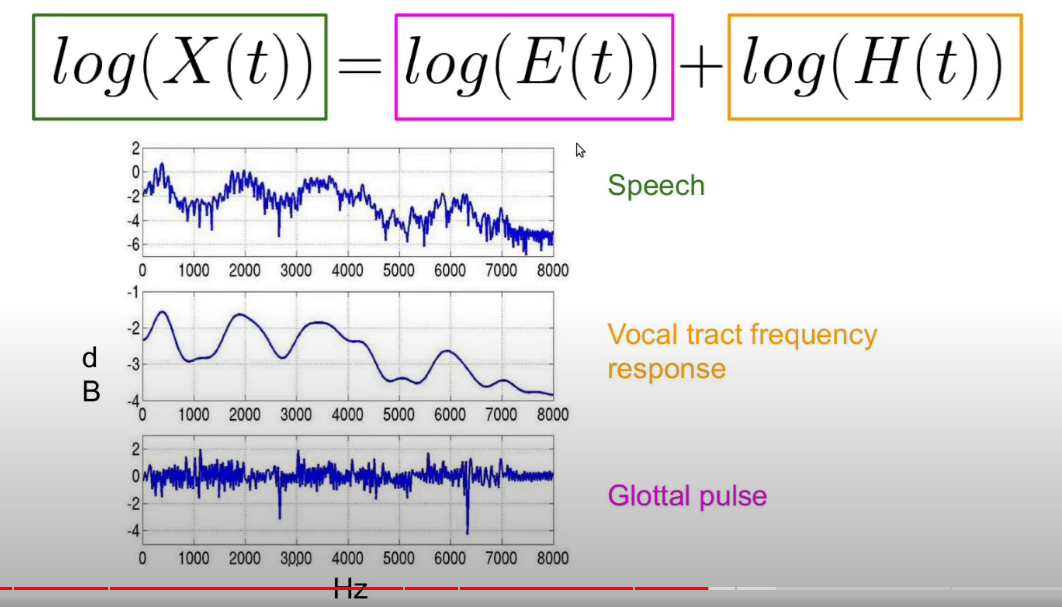


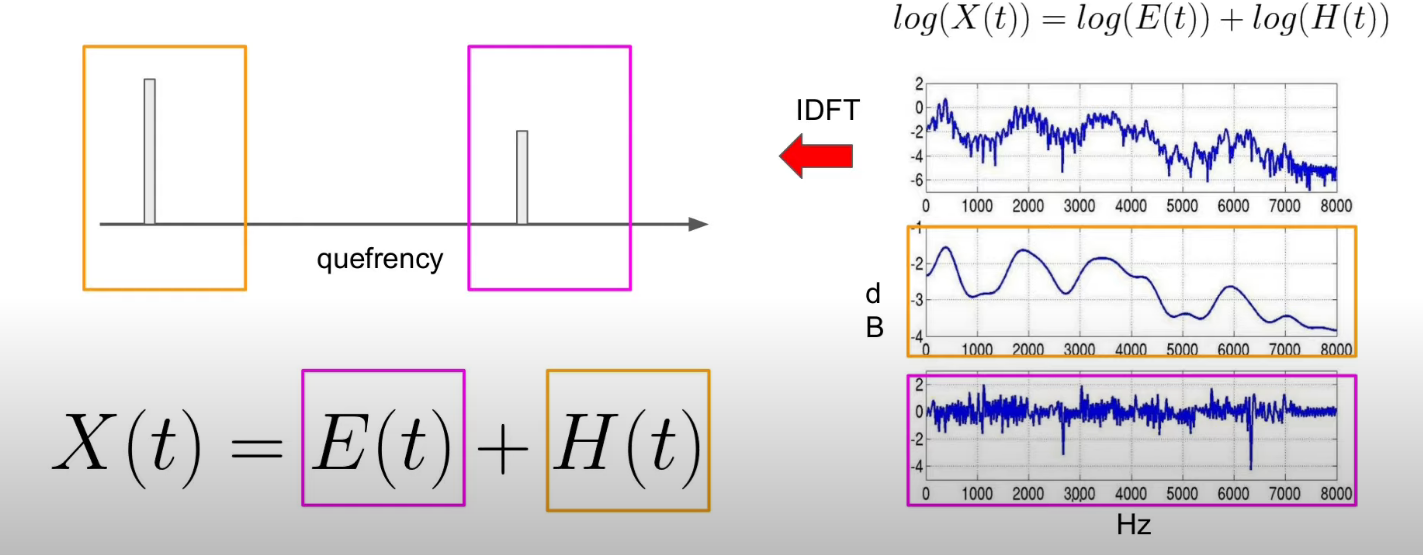
Speech = glottal pulse + vocal tract frequency response (faltung)

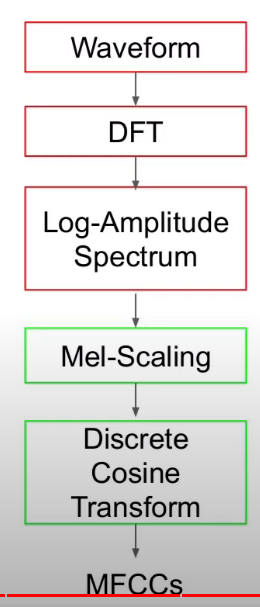
Ein Bild, das Text, Whiteboard enthält.

Automatisch generierte Beschreibung

Log um die einzelnen parts unabhängig voneinander zu betrachten

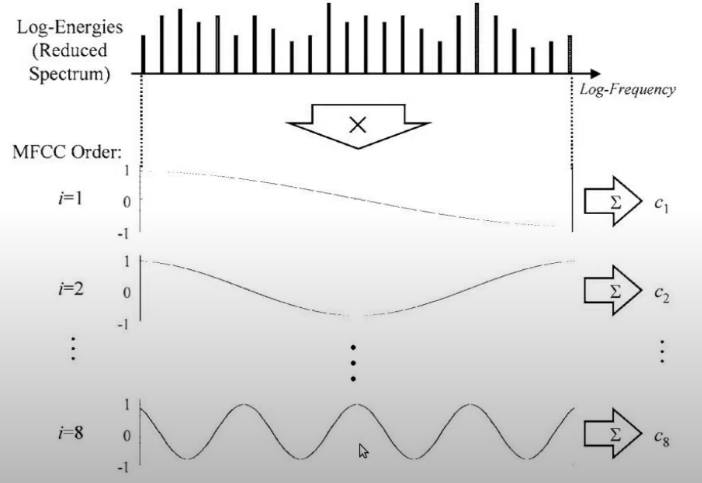


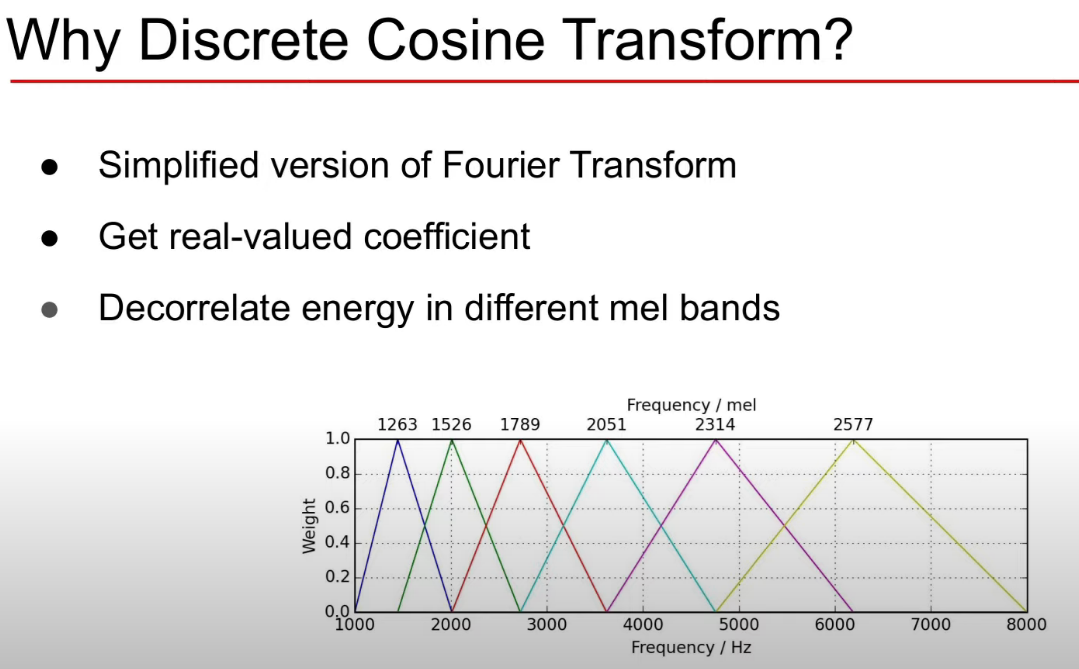




Log-amplitude spektrum: menschen nehmen lautstärke nicht linear wahr

Mel-scaling: log frequenz: menschen nehmen frequenzen nicht linear wahr





Alle dreiecke bei mel bands überschneiden, also korrelieren. Ml profitiert von features, die möglichst nicht korrelieren

DCT reduziert auch dim

Wie viele koeffizienten? Ersten 12-13? Haben die meisten informationen, z.B. formants, spectral envelope

Deltas, deltadelta MFCC 🡪 39 koeffizienten

Ein Bild, das Text, Elektronik, Anzeige, Screenshot enthält.

Automatisch generierte Beschreibung

Soll die kleinen Strukturen des Spektrums ignorieren (noise, unwichtig) (pitch not important)

Die großen Strukturen des Spektrums beachten

* Hochgradig experten/menschen bias, maschinen finden nicht selbst heraus, was ideal wäre
* Anfäliig auf noise
* Keine inverse, nur approximation

Mehr pitch invariant

In code:

Kaldi\_data.load\_wav(): return data, samplerate

KaldiDiarizationDataset:

\_\_init\_\_():

From train.yaml:

Sampling\_rate = 8000, subsampling = 10, Frame\_shift = 80, frame\_size = 200

Chunk\_size = num\_frames = 500, context\_size = 7

For rec in data.wavs:

Data\_len = int(reco2dur[rec] \* rate / frame\_shift) # 20s \* 8000 / 80 = 2000

Data\_len = int(data\_len / self.subsampling) # 200 / 10 = 200

St, ed = gen\_frame\_indices(200, 500, 500, subsampling=10) = (0, 500)

Chunk\_indices.append(rec, st \* subsampling = 0 \* 10 = 0, ed \* subsampling = 500 \* 10 = 5000

Get\_example:

Rec, st, ed = chunk\_indices[i] # „ES2010b“, 0, 5000

Y, T = feature.get\_labeledSTFT(kaldi\_data, rec, st, ed, frame\_size = 200, frame\_shift = 80)

Feature:

get\_labeledSTFT: returns Y: STFT (n\_frames, n\_bins), T: labels (n\_frames, n\_speakers)

(rec = „ES2010b“, start = 0, end = 5000, frame\_size = 200, frame\_shift = 80)

data, rate = kaldi\_obj.load\_wav(rec, start \* frame\_shift, end \* frame\_shift) # ES2010b, 0, 400000 🡪 return vector 400000 entries, 8000 rate

Y = stft(data, frame\_size = 200, frame\_shift = 80)

Stft(data, frame\_size = 200, frame\_shift = 80):

Fft\_size = 1 << (frame\_size – 1).bit\_length() # 199.bit\_length() = 8, „1“ 8 mal nach links (an die neunte Stelle) : 100000000 = 256 (next biggest 2er potenz

Len(data) = 400000 % frameshift = 80 == 0 # true:

Librosa.stft(data, n\_fft=fft\_size = 256, win\_length=frame\_size=200, hop\_length=frame\_shift=80).T (transponiert)

# Encoder Decoder Transformer Attention Networks

Feature extraction 🡪 sad 🡪 speaker change detection | overlapped speech detection 🡪 speech turn representation 🡪 speech turn clustering

* Feature extraction 🡪 vad 🡪 overlapped speech detection | speacker change detection 🡪 speaker embedding 🡪 clustering 🡪 resegmentation

# Verschiedenes

## SOX

$ sox IS1009a.Mix-Headset.wav -t wav -r 8000 -b 8 - remix - | cat > IS1009a.Mix-Headset\_manual\_1.wav

Eval2000: aus stm RTTM erstellen (mit overlap ?)

/mnt/md0/tools/kaldi/tools/sctk-2.4.10/src/stm2rttm/stm2rttm.pl stm -e rt05s > rttm\_rt05s

1. This file then contains not needed speaker information, get rid of it via
2. reformat

awk '$1 ~ /SPEAKER/' rttm\_rt05s | awk '{ $2=$2"-"$3; $3=1; $8=$2; $4=sprintf("%7.2f", $4); $5=sprintf("%7.2f", $5); print}' > rttm\_with\_overlap.annotation