A ZOOM FILTER FOR APPLAUSE AND LAUGHTER

Meeting 02.12.21

Project Background

Motivation

- Feedback when speaking at virtual conferences
  - Laughter and Applause

Idea

- Automatic laughter and applause detection in real-time
  - Using machine learning
- Ideally add this to an existing system as alternative to 'Mute'

Work done so far

1. Research review
   - decided to focus on laughter
     - Using model by Gillick et al.
   - decided to use ICSI corpus
2. Evaluation on whole ICSI corpus using different thresholds
   - Found recall and precision to be very low
3. Improved evaluation and investigated results
   - Results not acceptable
   - Possible data missmatch
4. Decided to retrain on ICSI corpus and try to adapt model for realtime

1 Research Review

- Why Gillick et al.'s model?
  - Recent – published Sept. 2021
    - State-of-the-art model
  - completely open source

- Why ICSI?
  - Meeting speech fits domain of project
    - Google Audio set doesn't match domain
  - Large corpus (~72h) -> enough laughter-only snippets
    - 8420 laughter only snippets
      - Average duration: 1.66s
      - Total duration: ~3.9h

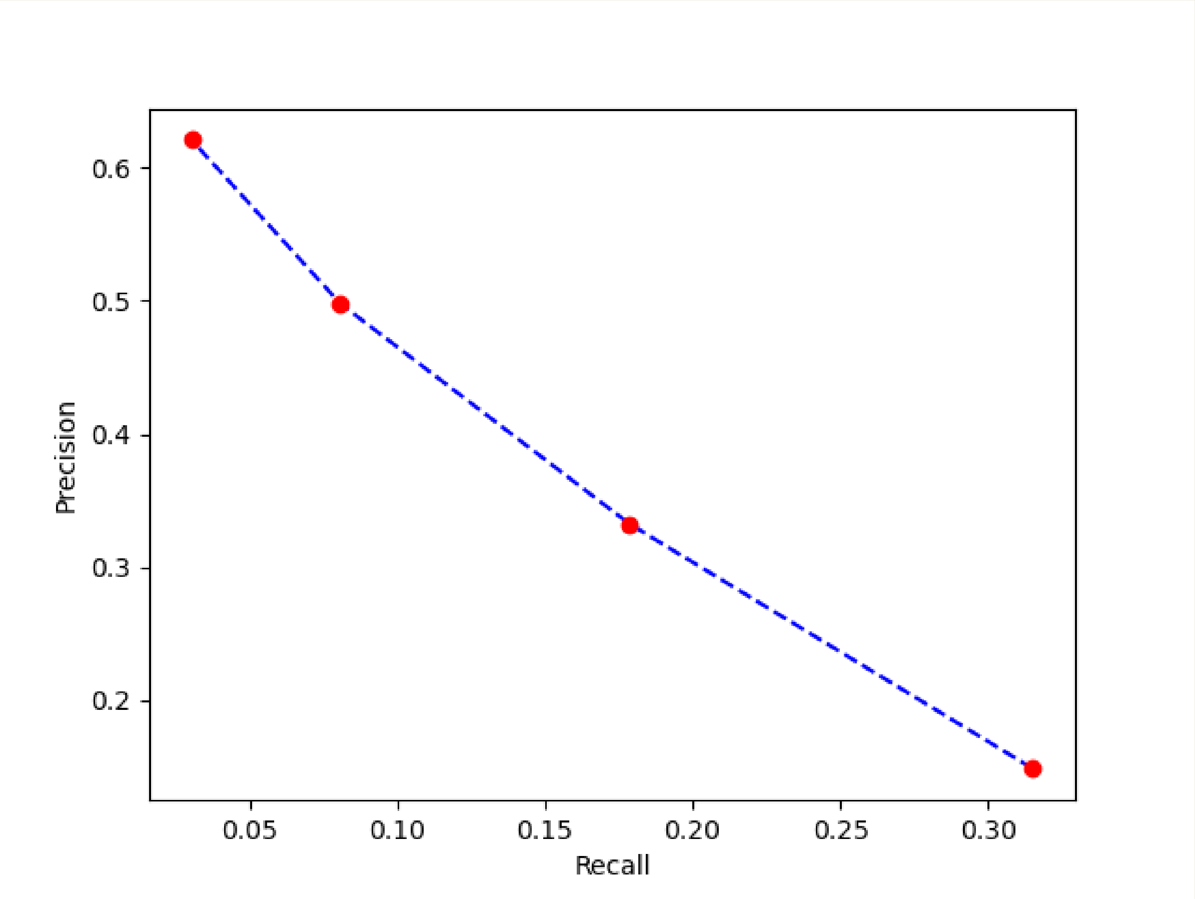
2 Evaluation on ICSI

Parameters:
threshold:
- minimum probability to classify frame as laughter
minimum length:
- minimum length of laughter segment to be identified

Meetings Evaluated: 75
thresholds tried: 4
Experiments with outcome: 1475

threshold	precision	recall
0.2	14.88 %	31.53%
0.4	33.26%	17.85%
0.6	49.81%	8.05%
0.8	63.08%	3.01%

- 29 of 1475 have a precision over 80%



3 Improved Evaluation

- Discarding laughter next to speech segments
  - Significantly improved precision
  - minor improvement in recall

	original method		new method	
threshold	precision	recall	precision	recall
0.2	14.88 %	31.53%	20.44%	37.40%
0.4	33.26%	17.85%	53.84%	20.68%
0.6	49.81%	8.05%	79.68%	8.92%
0.8	63.08%	3.01%	90.44%	3.22%

3 Practical Example Findings

- average meeting length: 56min
- average laughter length during meeting: 2:06 min

	new method		Laughter in [min:sec]		
thresh old	precision	recall	predicted	actual laughter	noise
0.2	20.44%	37.40%	5:03 min	1:08 min	3:55 min
0.4	53.84%	20.68%	1:00 min	0:33 min	0:27 min
0.6	79.68%	8.92%	0:14 min	0:11 min	0:03 min
0.8	90.44%	3.22%	0:04 min	0:04 min	0:00 min

4 Retrain + Real-Time

- Calculated RTF of current system on different machines
  - *Reference value for future models*

Next steps

- *Retrain model on ICSI data*
- *Investigate more efficient ML models*
  - *e.g. MobileNet*

Audio Duration	Iterations run	Average RTF
CPU - i5-6500 CPU @ 3.20GHz		
3s	20	1.31
30s	20	1.41
120s	10	1.49
CPU AT		
3s	20	0.63
30s	20	0.84
120s	10	0.81
GPU AT		
3s	20	0.14
30s	20	0.10
120s	20	0.10
300s	10	0.10

Next steps

- Retrain model
- Investigate more efficient ML models
- Practicality and possible alternatives

Once a real-time model with good performance is found and implemented

- investigate options for integration into existing systems
  - Possibly as proof of concept prototype

- Fill in this form: https://forms.office.com/r/K8NrMuh209

Possible systems

- Midspace (prev. Clowdr)
  - *https://midspace.app*
  - Open source virtual conference platform

- Jitsi Meet
  - Open source meeting platform similar to zoom/teams

Real-time/Latency - Factors

- Frame- and Window-size
  - if window=1s we need to wait 0.5s before we start prediction
- Model complexity (includes preprocessing)
  - the more complex the function to calculate the probability the higher the latency
- Computational power of device
  - Feature should be used by end-user -> cannot require GPU
- Programming language
- "Minimum-laughter-length"-parameter of the model