Speech Coding Enhancement for Opus: Progress report

Presenter: Jan Buethe (AWS)

jbuethe@amazon.com

IETF 120

draft-buethe-opus-speech-coding-enhancement

Overview

I. Requirements for speech coding enhancement (test data)

II. Some examples regarding IETF119 questions about real-world / out-of-domain signals (farfield, music, double talk)

III. Next steps

Attachment sent to mlcodec@ietf.org on 07/26/2024

(opus_speech_coding_enhancement_ietf120_attachment.zip)

Part I: Requirements

- Aim of draft-buethe-opus-speech-coding-enhancement is to specify requirements for speech coding enhancement methods
- Opus 1.5 released and includes two optional speech enhancement methods (LACE and NoLACE) => have integrated methods for testing requirements
- So far: Evaluation of quality metrics at IETF 118 => modified opus compare (MOC) distance simple to use and correlates reasonably well with quality.
- Next question: What data to test on?

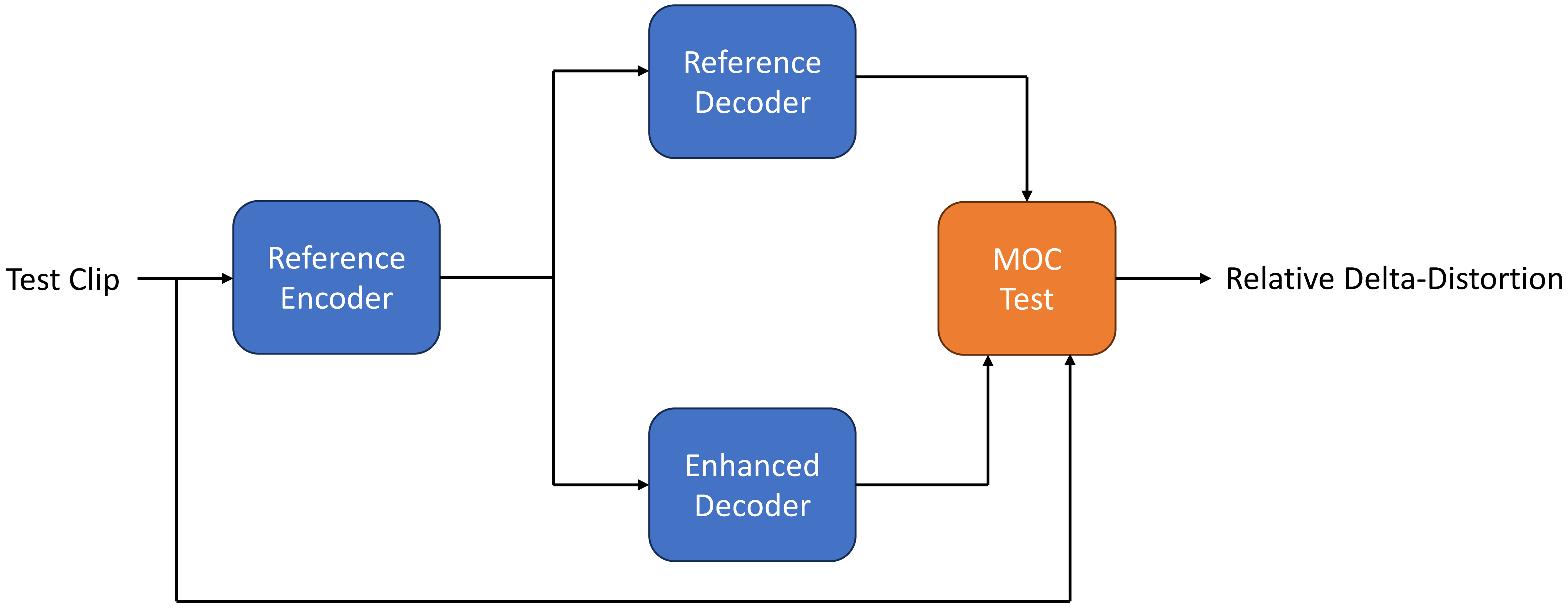
Test data

- Conventional approach: clean-speech dataset with controlled degradations (add noise, simulate reverb, etc.)
- Drawbacks:
  - Not much (free) multi-lingual high-quality data available
  - Not clear how well simulations reflect real use cases
- Alternative: Mozilla Common Voice dataset
  - User generated (quality ranges from clean to noisy to reverberant)
  - Recording scenarios are very relevant for communication over the internet
  - Data covering 100+ languages gathered from all over the world
  - Addresses concerns regarding real-world data

A multi-lingual test on Common Voice 18

- Testing done per language:
  - 10 samples per language
  - 5 female, 5 male sampled uniformly from different sources (client id)
- Languages that do not permit such sampling (due to size) are excluded => leaves 81 languages or dialects
- Items are converted from 32 kHz to 16 kHz and normalized to -6 dBFS
- Script for item selection can be found on opus main branch (https://gitlab.xiph.org/xiph/opus/-/blob/main/dnn/torch/osce/stndrd/evaluation/commonvoice_clip_selection.py)

Test Setup


Test Clip
Reference Encoder
Reference Decoder
Enhanced Decoder
MOC Test
Relative Delta-Distortion

Pass Criteria

- For reference output x_{ref} and test output x_{test} corresponding to input signal x_{orig} relative delta-distortion is defined as

$$d(x_{ref}, x_{test}) = \frac{\sqrt{MOC(x_{orig}, x_{ref})} - \sqrt{MOC(x_{orig}, x_{test})}}{\sqrt{MOC(x_{orig}, x_{ref})}}$$

- For a test set M, pass criteria are
  - Limited worst case relative delta-distortion: $\min_{M} d(x_{ref}, x_{test}) > \theta_{min}$
  - Limited average relative delta-distortion: $\mathrm{avg}_M\, d(x_{ref}, x_{test}) > \theta_{avg}$
- Test script on opus main branch (https://gitlab.xiph.org/xiph/opus/-/blob/main/dnn/torch/osce/stndrd/evaluation/run_osce_test.py)

Test Conditions

- Reference encoder: opus 1.5.2
- Reference decoder: opus 1.5.2 with -dec_complexity 0
- LACE: opus decoder with -dec_complexity 6
- NoLACE: opus 1.5.2 with -dec_complexity 7
- BadLACE, BadNoLACE: same as LACE, NoLACE but with weight files from almost untrained models (same as IETF118)

Examples of BadLACE and BadNoLACE output in attachment (part1/badmodel_examples).

Results

- Thresholds: $\theta_{min} = -0.1$, $\theta_{avg} = -0.025$
- All good methods pass, all bad methods fail (at least partially)
- Worst case NoLACE samples included in attachment, no perceptual degradation found (part1/nolace_worst_case_items)

Tests passed

Condition \ Bitrate	6 kb/s	9 kb/s	12 kb/s	15 kb/s	24 kb/s
LACE	81\81	81\81	81\81	81\81	81\81
NoLACE	81\81	81\81	81\81	81\81	81\81
BadLACE	42\81	0\81	0\81	0\81	0\81
BadNoLACE	0\81	0\81	0\81	0\81	0\81

Part II: Out-of-domain signals

- At IETF 119 the question about out-of-(training)-domain signals was raised.
- Specifically, farfield recordings, music and double talk were mentioned
- Prepared three samples as a case study
  1. Violin sample
  2. Double-talk item
  3. Farfield recording
- Reporting MOC scores, subjective impression, and full set of items in attachment (part2/examples)

Music

- Item: short violin clip within the frequency range of human speech
- Observations:
  - Item is classified as speech but output degraded due to pitch instabilities
  - Listening impression: NoLACE better than LACE better than Opus (but quality with NoLACE still poor)
  - Indicates that LACE and NoLACE are robust to pitch prediction errors

MOC

Bitrate \ condition	Opus	LACE	NoLACE
9 kb/s	0.769	0.659	0.663
12 kb/s	0.576	0.527	0.529

Farfield

- Item: Clean speech item from EBU SQAM CD* processed with real room impulse response
- Observations:
  - Listening impression: NoLACE better than LACE better than Opus

MOC

Bitrate \ condition	Opus	LACE	NoLACE
9 kb/s	0.767	0.619	0.638
12 kb/s	0.556	0.538	0.542

* https://tech.ebu.ch/publications/sqamcd

Double-talk

- Item: Downmix of stereo arrangement of two speech items from EBU SQAM with a bit of stereo reverb (simulates two talkers in a room)
- Observations:
  - Listening impression: NoLACE better than LACE better than Opus (though differences are small for 12 kb/s)

MOC

Bitrate \ condition	Opus	LACE	NoLACE
9 kb/s	0.742	0.668	0.680
12 kb/s	0.587	0.577	0.578

Conclusion

- Test on Common Voice successfully distinguished good from bad enhancement methods (though the number of test methods is currently limited).
- In current setting, thresholds must allow a small MOC degradation. Listening impression is that enhanced signals still sound better (feedback welcome).
- Critical test items (music, farfield, double talk) also improved by Opus 1.5 enhancement methods (please also hear for yourself!).
- Modified Opus Compare (moderately) successfull in confirming quality improvement of enhancement methods.

Questions and next steps

- Add small clean-speech, farfield and music test sets to Common Voice sample and validate tests
- Collect feedback
  - Is test coverage good enough?
  - Is the test method acceptable?
- Define test points
  - Is having a reference encoder/decoder acceptable? => 120 MB for raw audio
  - Do we need per-test-point data? => ~ (num test points + 1) * 120 MB for raw audio
  - Where can we host data?

Other updates

- First attempt at using side-info for enhancement
  - Idea: feed extra info from clean signal to NoLACE
  - Improved quality but not enough to justify bitrate overhead

Thank you!