Обзор методов

Подходы с предобученными моделями

Один из самых распространенных подходов, это использование фрейворка от Google с готовой моделью WebRTC VAD (использовалась GMM). Из доступных настроек: три режима чувствительности к речи, которые можно использовать в зависимости от уровня шума в данных. Из главных преимуществ, что он открытый для использования и бесплатный.
https://github.com/wiseman/py-webrtcvad

Есть и другие реализации:

Silero: https://github.com/snakers4/silero-vad. В своей статье они пишут, что смогли побороть метрики качества WebRTC.
Kaldi: https://www.idiap.ch/software/bob/docs/bob/bob.kaldi/master/guide.html
NeMo: использование архитектуры MatchBox
https://ngc.nvidia.com/catalog/models/nvidia:vad_matchboxnet_3x1x1

Нейросетевые подходы

За последние несколько лет вышло достаточно много статей, посвященных нейросетевым подходам к решению задачи VAD

RNN-LSTM

Достаточно много экспериментов было проведено с рекуррентными сетями, в которых использовались LSTM слои. Такой подход обошел по метрикам качества классические методы решения, особенно если в примерах встречался шум. Рекуррентные сети способны захватывать временных закономерностей в аудио данных.

https://arxiv.org/pdf/2003.12266.pdf

CNN

Также использовались сверточные сети, и в некоторых экспериментах были указаны результаты, которые были лучше, чем с использованием LSTM архитектур. Например, FAR равный 5.67% при использовании CNN по сравнению с LSTM при FRR of 1%. Но главный минус сверточных архитектур в том, что они очень глубокие, в результате чего, их невозможно использовать в риал-тайм ASR системах.

https://github.com/SIP-Lab/CNN-VAD
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8278160

CNN-BiLSTM

Следующим шагом в развитии нейросетевых архитектур стало использование комбинации сверточных и рекуррентных сетей в модели. Использование BiLSTM слоя позволило облегчить модели, но и не потерять в качестве. В итоге, такая модель смогла побить метрики бейзлайновой модели с использованием глубокой сети ResNet, и при этом сохраняя небольшие размеры самой сети, что позволяет ее использование на мобильных устройствах.

https://github.com/NickWilkinson37/voxseg CNN-BiLSTM
https://arxiv.org/pdf/2103.03529.pdf

LSTM + Attention

Классическую модель с LSTM слоем можно улучшить, добавив в нее аттеншн. Это позволит не особо увеличивая размеры модели, сделать ее более устойчивой к шуму

https://indico2.conference4me.psnc.pl/event/35/contributions/3072/attachments/717/755/Thu-1-4-2.pdf

U-NET

Если говорить о глубоких архитектурах, то в последнее время часто стали заимствоваться архитектуры из тех, что успели показать свои результаты, например, в Computer Vision. Но реализовать такие архитектуры в риал тайме сложнее.

https://www.mdpi.com/2076-3417/10/9/3230/htm
file:///tmp/Multi-Task_Learning_U-Net_for_Single-Channel_Speec.pdf
https://arxiv.org/pdf/2002.06033.pdf

Использование фичей

В некоторых работах по применению VAD в качестве фичей использовались, например, фичи в результате преобразования фурье (STFT) и последующей 2D сверточного слоя. Но в таком случае придется работать в фичесетом в трехмерном пространстве, что достаточно дорого по вычислениям

Чтобы побороть эту проблему было решено использовать коэффициенты MFCC и их дельты на каждом фрейме длительностью 10мс. Такие характеристики не требовательны к ресурсам, их достаточно просто вычислить и хранить.

Использование loss

Задача VAD является задачей бинарной классификации, поэтому можно спокойно использовать cross-entropy loss (CE)

В ряде статей упоминалось использование Focal Loss, но значимого прироста в качестве по сравнение с CE практически нигде нет.
https://raw.githubusercontent.com/nicklashansen/voice-activity-detection/master/Paper.pdf

FL может зайти если присутствует жесткие дисбаланс классов

https://indico2.conference4me.psnc.pl/event/35/contributions/3072/attachments/717/755/Thu-1-4-2.pdf

Репозитории

VAD

https://github.com/nicklashansen/voice-activity-detection
https://github.com/filippogiruzzi/voice_activity_detection
https://github.com/RicherMans/GPV/blob/master/models.py
https://github.com/jtkim-kaist/VAD

Loss

https://github.com/BloodAxe/pytorch-toolbelt/tree/develop/pytorch_toolbelt/losses
https://github.com/CoinCheung/pytorch-loss/blob/master/pytorch_loss/focal_loss.py