

주간보고

주제: use offline ASR model as if online: pseudo-stream-asr

학생 최재원

Prof. Jae-Hong Lee

Table of Contents

1. 문제 정의
2. 기존 연구
3. 문제 해결 방안
4. 실험 결과
5. 결론
6. 추후 진행 예정 사항

- 문제
 - 실시간일 것
 - 자동 음성 인식(ASR)이 가능할 것
 - 데모 프로그램을 만들 것
- 제한
 - 학습 시킨 모델을 적용할 수 있을 것 (스트리밍 모델을 사용하지 말 것)
 - 발화의 시점과 종점도 예측할 것
 - 실시간성을 해칠 만큼 느리지 않을 것

- offline model을 teacher로, online model을 student로의 knowledge distillation
 - for offline use – process entire one
 - [ctc](#) | Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with RNN
 - [conformer](#) | Convolution-augmented Transformer for Speech Recognition
 - for online use – process whenever model received
 - [rnn-t](#) | Sequence Transduction with Recurrent Neural Networks
 - pros: (WER) offline < online, narrowing gap between offline and online, imitate using unlabeled data
 - cons: complicated
- chunked attention
 - ??

- KD examples only available for TTS(mainly) and SVS task | @ESPnet
 - SVS(Singing Voice Synthesis, 노래음성합성)
 - 25일까지 결과물을 만들 수 없음

- 모델에 음성 데이터를 넣었으나 제대로 인식되지 않는 문제 (gradio audio 이용)
 - 현상
 - 데이터를 numpy.ndarray로 받으면 엉뚱하게 인식함
 - 음성이 임시로 저장된 경로를 받으면 제대로 인식함
 - 진행한 전처리
 - 학습 시 사용했던 sampling rate만 맞추어 줌
 - 추론 과정에서 dtype of numpy.ndarray가 float32여야 한다고 하여 형변환만 해줌
 - 해결
 - float32 형태로 형변환 후 +-1로 정규화
 - 관찰
 - 음성이 저장된 경로를 받으면 제대로 인식한 이유
 - float32 형태로 불러오며 자동으로 정규화 해주었기 때문

```
pseudo_stream_asr.py X
with_gradio > pseudo_stream_asr.py > ...
23 class PseudoStreamASR:
68     async def append_stream(self, audio, audio_stream, text_stream):
69         if audio is None:
70             return gr.Audio(), text_stream, audio_stream, text_stream
71
72         _sr, y = audio
73         y = y.astype(np.float32, order="C") / 32768.0
74         y = librosa.resample(y, orig_sr=_sr, target_sr=self.sr)
75
76         if audio_stream is None:
77             new_audio = (self.sr, y)
78         else:
79             new_audio = (self.sr, np.concatenate((audio_stream[1], y)))
80
81         self.helper.append_stream(y)
82         text = self.helper.get_status()
83
84         return new_audio, text, new_audio, text
85
```

- 특별한 토큰을 이용하면 정리할 수 있을까?
 - 단순 음성 시작/종료시 <sos/eos> 토큰을 뱉음을 확인
 - 쪼갬 음성을 여러 번 모델에 넣을 예정이라 부적절
 - 구현 과정
 - bpe config에 인덱스-토큰 정보 존재 (asr_train_config로 우회)
 - <instance of Speech2Text>.converter.ids2tokens
 - <blank> 토큰 제거 주석처리
- 0000 ⇔ <blank>
- 0001 ⇔ <unk>
- 4999 ⇔ <sos/eos>

```
asr_inference.py x
Users > lai01 > dev > espnet > espnet2 > bin > asr_inference.py > Speech2Text > from_pretrained
75 class Speech2Text:
587 def _decode_single_sample(self, enc: torch.Tensor) -> ListOfHypothesis:
661
662     # remove sos/eos and get results
663     last_pos = None if self.asr_model.use_transducer_decoder else -1
664     if isinstance(hyp.yseq, list):
665         token_int = hyp.yseq[1:last_pos]
666     else:
667         token_int = hyp.yseq[1:last_pos].tolist()
668
669     # remove blank symbol id, which is assumed to be 0
670     # token_int = list(filter(lambda x: x != 0, token_int))
671
672     # Change integer-ids to tokens
673     token = self.converter.ids2tokens(token_int)
674
675     if self.tokenizer is not None:
676         text = self.tokenizer.tokens2text(token)
677     else:
678         text = None
679     results.append((text, token, token_int, hyp))
680
681     return results
```

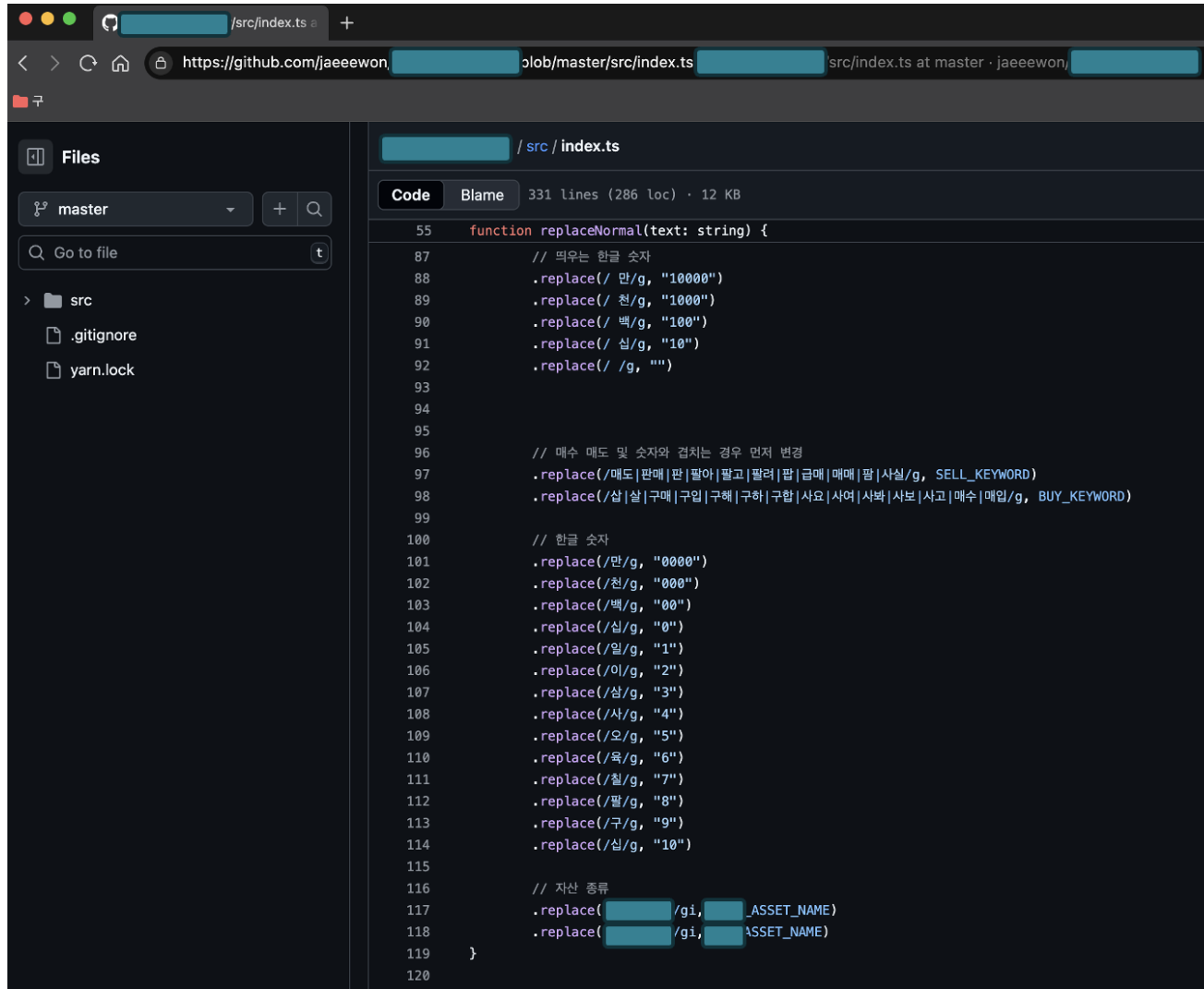
```
espnet2asr / assignment / 250825 / with_gradio / stream_helper.py
Code Blame 103 lines (84 loc) · 3.28 KB
23
24 class StreamHelper:
25 def __init__(self, s_context: int, t_poll: float, model: Speech2Text):
26     self.s_context = s_context
27     self.t_poll = t_poll
28     self.model = model
29     self.store: List[np.ndarray] = []
30     self.golden = -1
31     self.inference_task = None
32     self.status: List[Status] = []
33     self.joiner = RealtimeJoiner()
34
35     # with open(model_config['asr_train_config'], mode='r') as f:
36     #     self.token_list: List[str] = yaml.safe_load(f)['token_list']
37     # self.model.converter does it.
38
```

```
espnet2asr / assignment / 250825 / with_gradio / stream_helper.py
Code Blame 103 lines (84 loc) · 3.28 KB
55 def get_average_decibels(self, audio_data: np.ndarray):
56     average_decibels = 10 * np.log10(mean_square + epsilon)
57
58     return average_decibels
59
60 def exec_inference(self):
61     audio, window = self.get_stream()
62     db = self.get_average_decibels(audio)
63     start_time = time.perf_counter()
64     out = self.model(audio)
65     text = out[0][0]
66
67     # out[i] = (text, token, token_int, hyp)
68     # print("=" * 10)
69     # for i in range(len(out)):
70     #     hyp = out[i][3]
71     #     print(f"[{i}] total log probability: {hyp.score:.2f}")
72     #     print(f"[{i}] normalized log probability: {hyp.score / len(hyp.yseq):.2f}")
73     #     print(
74     #         f"[{i}] hypo: "
75     #         + "".join(self.model.converter.ids2tokens(hyp.yseq))
76     #         + "\n"
77     #     )
78     # print("=" * 10)
79     # text = "".join(self.model.converter.ids2tokens(out[0][3].yseq))
80
```

- Pseudo Stream ASR; 마치 실시간인 것처럼
 - context window: 모델에 입력할 최대 context 개수
 - every: 한 context가 가질 음성의 길이(초)
 - context가 겹쳐 발생하는 overlapped token들은 rule-based로 정리
 - execution time | (context_from, to) | avg_db | recognized > total
 - 문장이 끝난 것으로 인식하면 대괄호로 나타냄

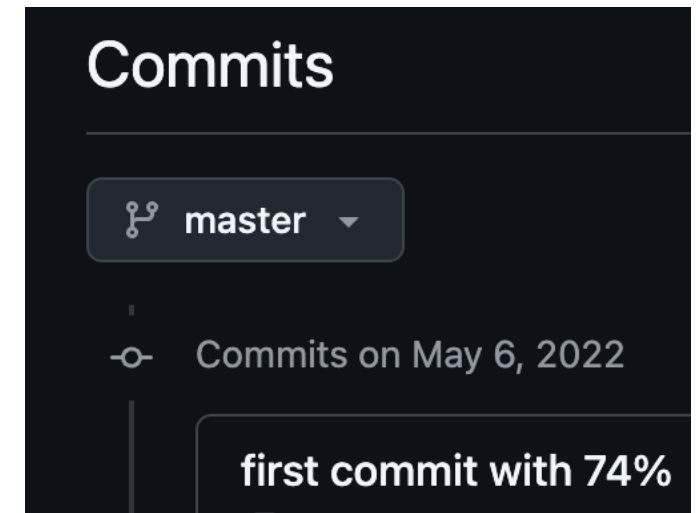
Textbox

```
0.77s | (9, 14) | -69.98 | > [HELLO]
0.69s | (11, 16) | -69.45 | > [HELLO]
0.75s | (13, 18) | -35.54 | BUT > [HELLO] BUT
0.82s | (15, 20) | -31.54 | WHERE ARE YOU NOW > [HELLO] [BUT]
1.33s | (17, 22) | -26.27 | WHERE ARE YOU NOW I'M WAITING FOR > [HELLO] [BUT] WHERE ARE YOU NOW I'M WAITING FOR
1.12s | (20, 25) | -27.23 | I'M WAITING FOR YOU > [HELLO] [BUT] WHERE ARE YOU NOW I'M WAITING FOR YOU
0.85s | (23, 28) | -37.11 | YOU > [HELLO] [BUT] [WHERE ARE YOU NOW I'M WAITING FOR YOU]
1.03s | (25, 30) | -63.11 | EH > [HELLO] [BUT] [WHERE ARE YOU NOW I'M WAITING FOR YOU]
0.87s | (28, 33) | -62.21 | OH > [HELLO] [BUT] [WHERE ARE YOU NOW I'M WAITING FOR YOU]
0.62s | (30, 35) | -61.16 | > [HELLO] [BUT] [WHERE ARE YOU NOW I'M WAITING FOR YOU]
```



The screenshot shows a GitHub web interface for a repository. The left sidebar displays the file structure with a 'src' directory containing '.gitignore' and 'yarn.lock'. The main area shows the 'src/index.ts' file with 331 lines of code. The code is a TypeScript function 'replaceNormal' that takes a string 'text' and returns a modified string. It uses multiple '.replace()' calls to substitute Korean units (e.g., '만', '천', '백', '십') and asset types (e.g., '매수', '판매', '팔아', '팔고', '팔려', '팔파', '급매', '매매', '팜', '사실') with numerical values or specific keywords. The code is currently on the 'master' branch.

```
55 function replaceNormal(text: string) {
87     // 띄우는 한글 숫자
88     .replace(/ 만/g, "10000")
89     .replace(/ 천/g, "1000")
90     .replace(/ 백/g, "100")
91     .replace(/ 십/g, "10")
92     .replace(/ /g, "")
93
94
95
96     // 매수 매도 및 숫자와 겹치는 경우 먼저 변경
97     .replace(/매도|판매|판|팔아|팔고|팔려|팔파|급매|매매|팜|사실/g, SELL_KEYWORD)
98     .replace(/삼|살|구매|구입|구해|구하|구함|사요|사여|사박|사보|사고|매수|매입/g, BUY_KEYWORD)
99
100    // 한글 숫자
101    .replace(/만/g, "0000")
102    .replace(/천/g, "000")
103    .replace(/백/g, "00")
104    .replace(/십/g, "0")
105    .replace(/일/g, "1")
106    .replace(/이/g, "2")
107    .replace(/삼/g, "3")
108    .replace(/사/g, "4")
109    .replace(/오/g, "5")
110    .replace(/육/g, "6")
111    .replace(/칠/g, "7")
112    .replace(/팔/g, "8")
113    .replace(/구/g, "9")
114    .replace(/십/g, "10")
115
116    // 자산 종류
117    .replace(/[redacted]/g, [redacted].ASSET_NAME)
118    .replace(/[redacted]/g, [redacted].ASSET_NAME)
119 }
120
```



- Pseudo Stream ASR; 마치 실시간인 것처럼
 - 모델은 한 시점에 두 번 이상 추론을 수행하지 않음 (stall)
 - 음성 데이터를 List에 추가할 때 추론이 진행되고 있지 않다면 새로운 스레드에서 추론 시작
 - multi-thread는 추론 과정에서의 blocking을 예방하고자 사용함
 - List는 thread-safe하지 않아 접근시 threading.Lock을 사용함
 - queue는 thread-safe하지만, 임의로 접근하는 데 한계가 있어 사용하지 않음
 - 추론을 위한 데이터를 불러오는 메소드를 따로 둬م
 - 추론 결과가 최종 정답으로 확정된다면, 해당 context를 golden으로 두고자 했으나, 해당 기능 미구현
 - 끝에서 최대 s_context(context size)까지의 데이터를 concat하고, context window와 함께 반환함
 - 추론 결과를 잘 처리하여 최종 결과를 도출함
 - 데시벨, 토큰 개수로 1차 필터링 (quiet++)
 - 이전 결과로 시작하면 덮어쓰기
 - 이전 결과와 중복 찾아 결합하기 (difflib)
 - 전혀 다르거나 quiet > threshold면 문장 끝

```
1 def get_stream(self):
2     with self.lock:
3         len_str = len(self.audio)
4         start_index = max(self.golden + 1, len_str - self.s_context)
5         return np.concatenate(self.audio[start_index:]), (start_index, len_str - 1)
```

- 데모1 | Gradio를 통해 음성 획득
 - 데모2 | Sounddevice로 음성 획득
-
- 직접 사용해봅시다.

- 거의 실시간이라고 봐도 될 정도의 음성 인식기 데모를 제작함
 - 현재 설정은 모델에 0.5초 짜리 context를 최대 6개 입력하고 잘 다듬어 결과를 냄
 - 모델에 입력할 시간은 유지하며 context의 시간을 짧게 했을 때 성능이 선형적으로 증가함을 확인
 - 0.5s -> 0.25s
 - context 6 -> 12
 - 그러나, rule-based로 처리한 부분에서 문장을 두 번 인식하는 문제가 있음
 - 시간을 들여 해결할 수 있는 부분임

- 현재의 rule-based는 상당히 취약한 부분이 많음
 - Speech2Text의 nbest=1를 수정하면 더 많은 hypothesis를 제공함을 확인함
 - 여러 개의 후보를 두고 가장 잘 연결되는 후보를 gold로 결정하는 부분이 추가되면 좋을 것 같음
- KD를 통해 stream 가능하도록 시도해보고 싶음
 - 충분한 시간이 필요하며, 당장 진행하기에는 어려움이 많음
- librispeech pretrained model을 성공적으로 학습하지 못했는데, 관련 hyper-parameters를 공부하고 잘 학습시켜 본 데모에 적용해보고 싶음
 - ex. 어떤 모델은 어떤 optimizer가 좋은 성능을 내고, 그 때 lr이나 weight_decay 등은 얼마나 조절?
 - ex. gradient accumulation 공부로 VRAM이 부족하면 batch_bins를 줄이고, 그 비율만큼 accum_grads 올리면 거진 비슷한 걸 알게 되는 등

주간보고

학생 최재원
Language & AI융합전공

2025. 08. 25. 10 AM.
한국외대 교수회관 401호