Unsupervised ASR

TA Guan-Ting Lin

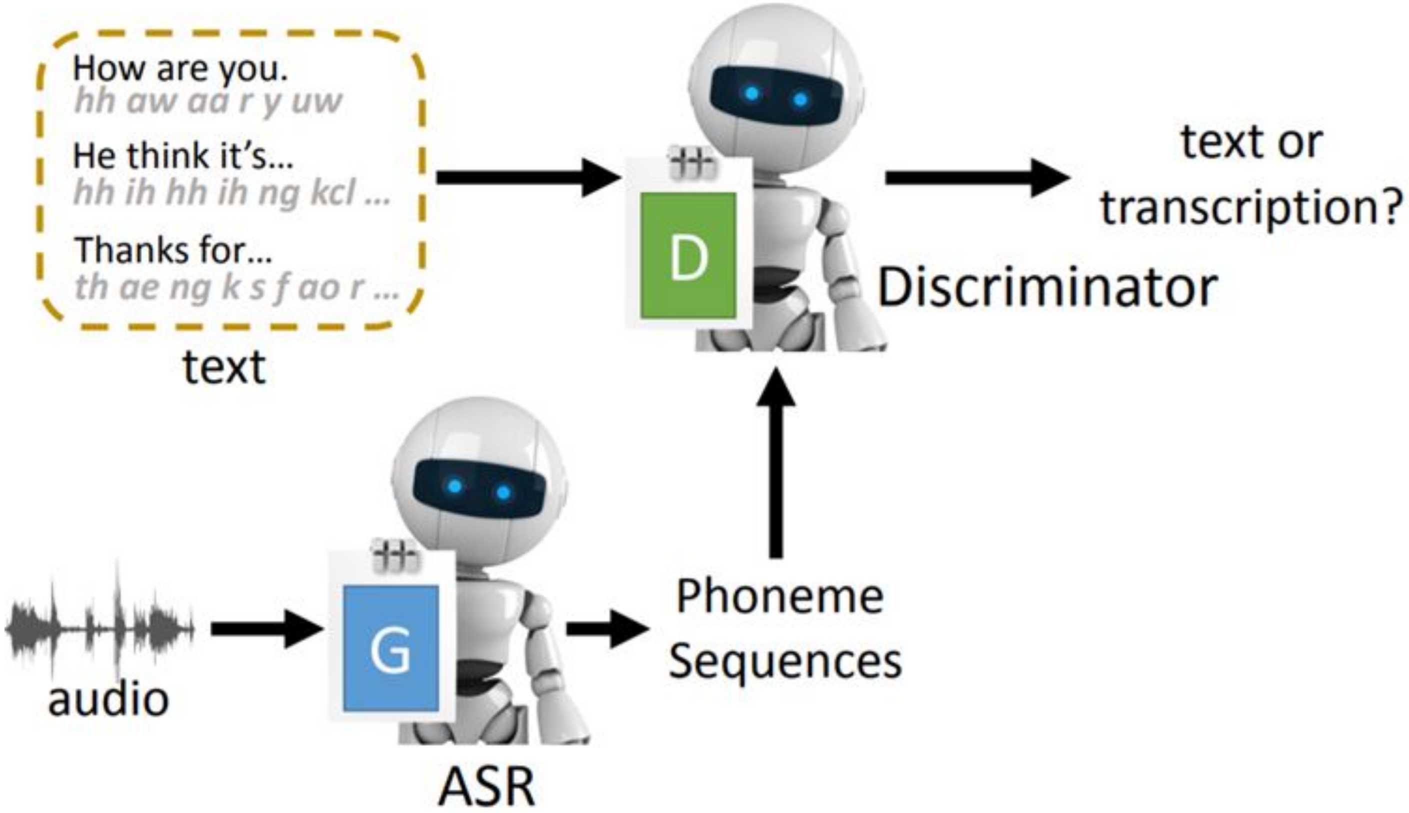
daniel094144@gmail.com

2021/11/21

Outline

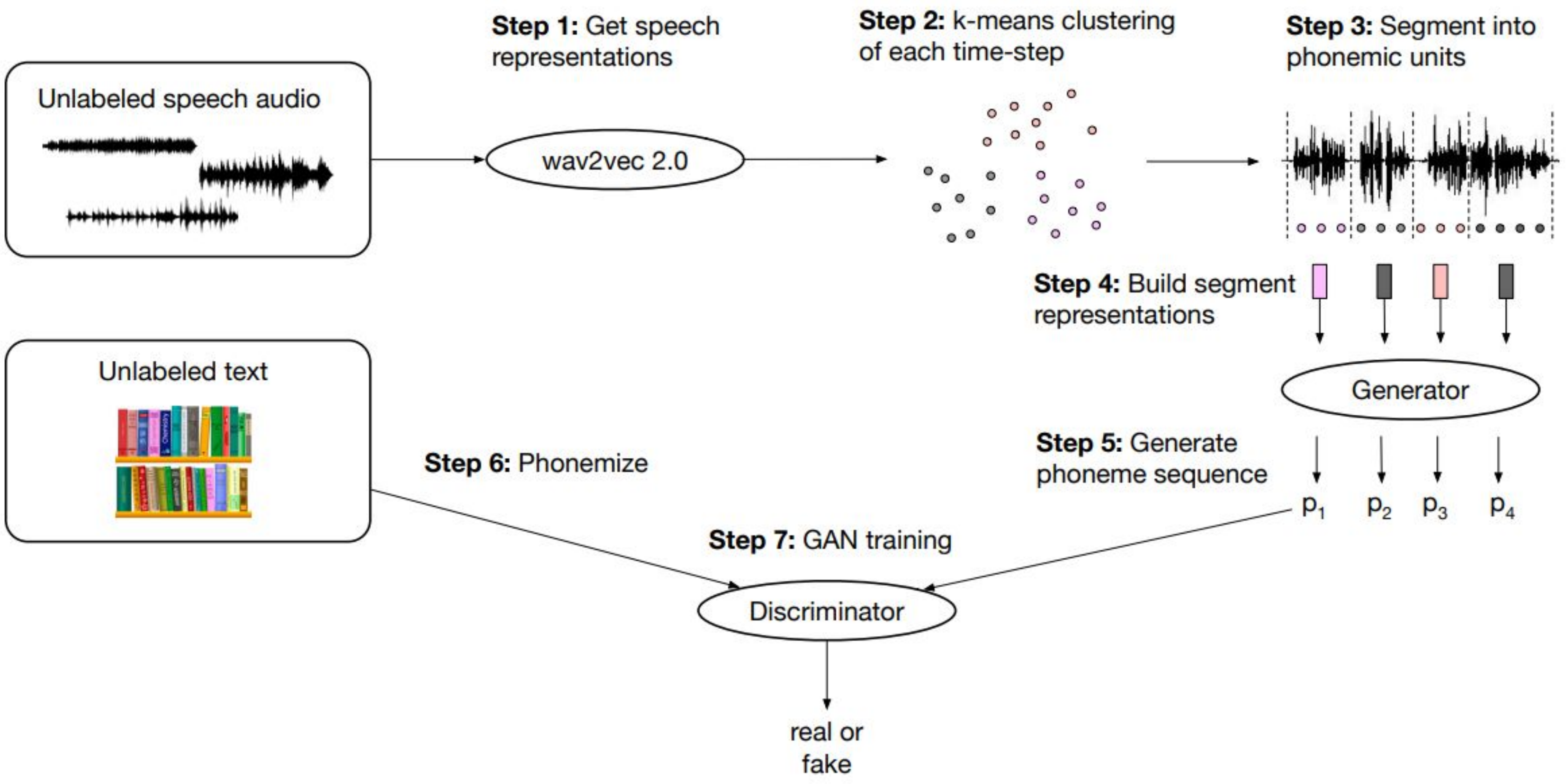
- unsupervised ASR introduction
    - self-supervised representation
    - data pre-processing
    - unsupervised learning (GAN)
    - result
- Homework
    - problem 1
    - problem 2

Unsupervised Speech Recognition



Reference: [DLHLP 2020] Text Style Transfer and Unsupervised Summarization/Translation/Speech Recognition

Overview



Step 1: Get speech representations
Step 2: k-means clustering of each time-step
Step 3: Segment into phonemic units
Unlabeled speech audio
wav2vec 2.0
Step 4: Build segment representations
Unlabeled text
Generator
Step 5: Generate phoneme sequence
Step 6: Phonemize
p1 p2 p3 p4
Step 7: GAN training
Discriminator
real or fake

Wav2vec 2.0

true quantized latent speech representation $\mathbf{q}_t$

$$\mathcal{L}_m = -\log \frac{\exp(sim(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

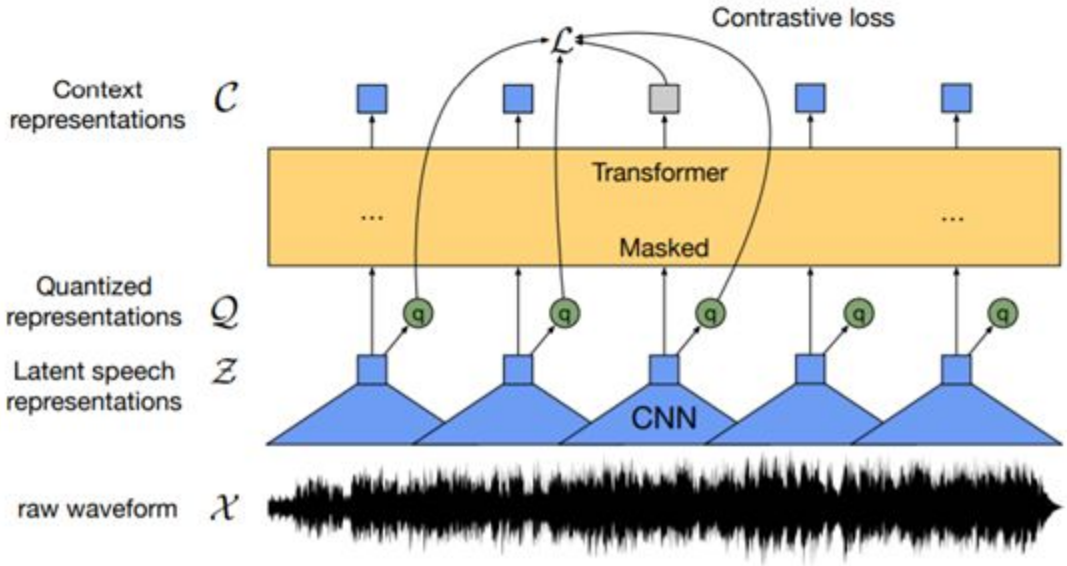
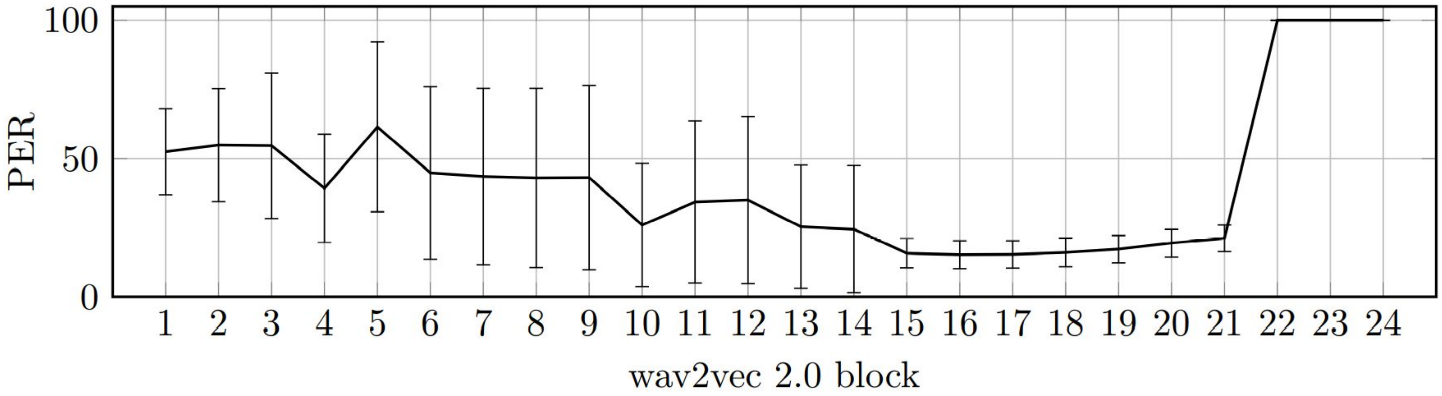


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

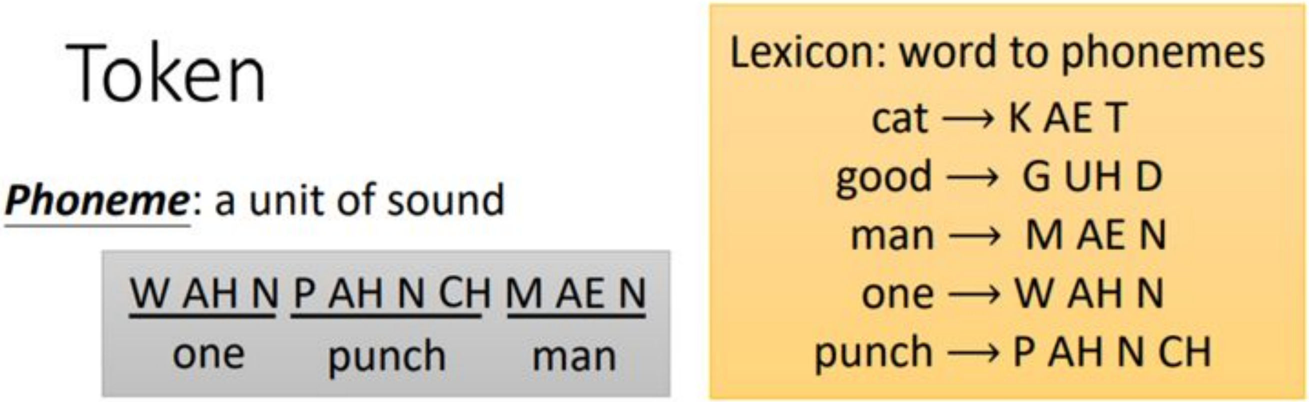
Speech pre-processing

- Removing silences: voice activity dectection (VAD)
- Speech Audio Representations: from w2v2-large 15th layer
- Identifying Speech Audio Segments: K-means clustering
- Segment Representations: PCA > mean-pooling



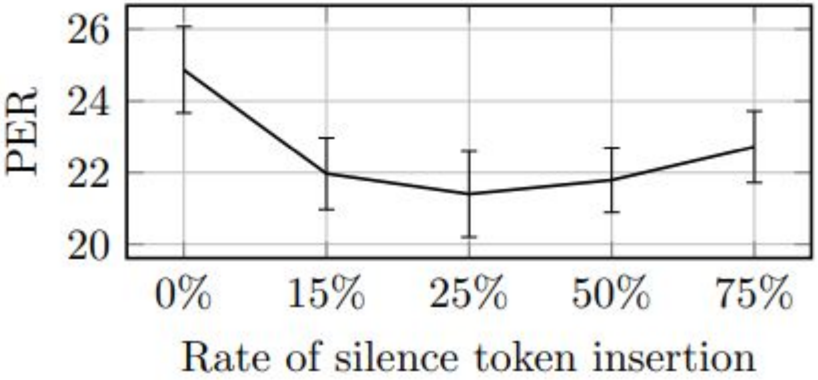
Text pre-processing

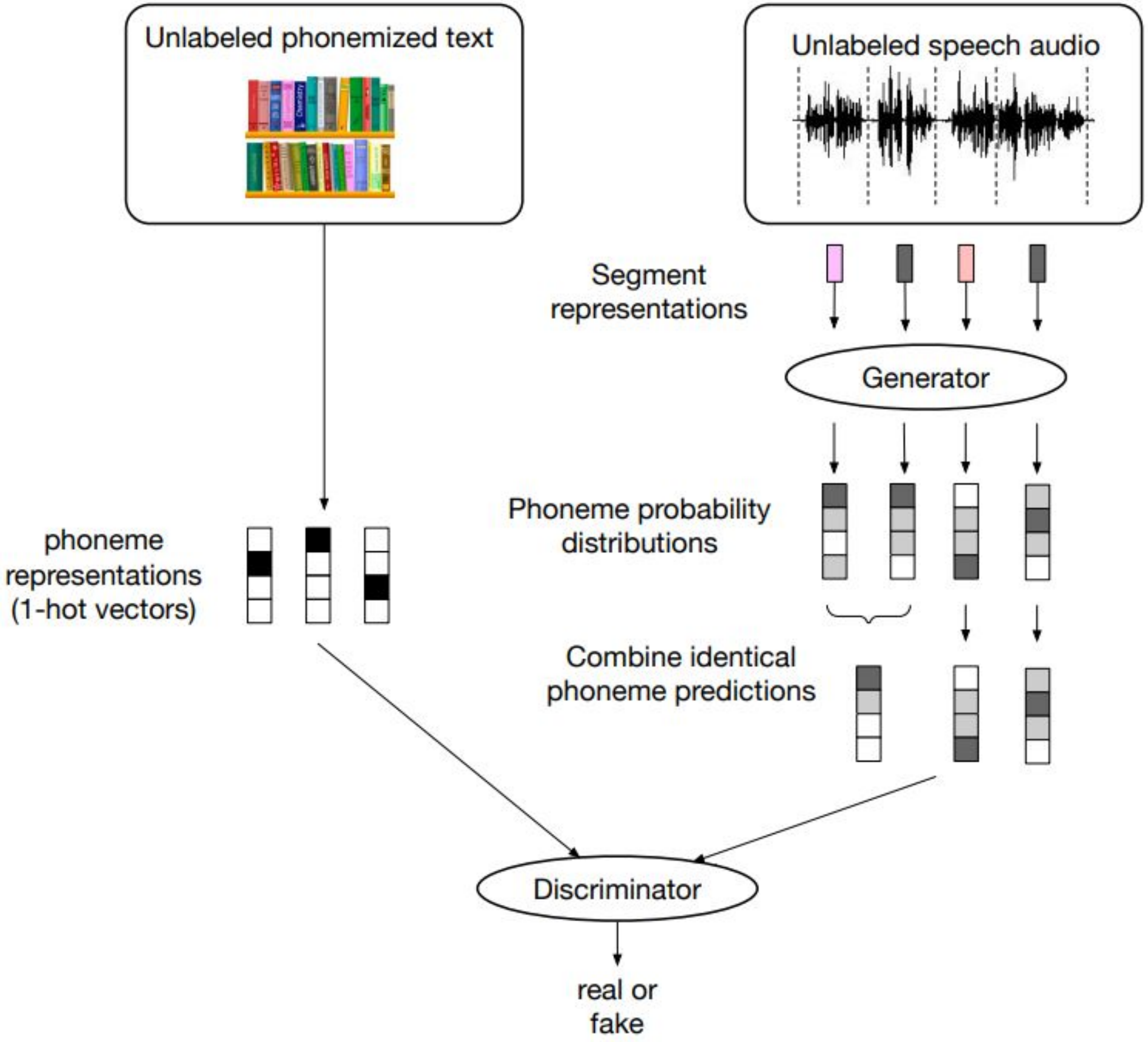
- Phonemization: grapheme to phoneme conversion



Text pre-processing

- Silence token insertion (0.25): since silence removal procedure is not perfect





Unlabeled phonemized text
Unlabeled speech audio
Segment representations
Generator
phoneme representations (1-hot vectors)
Phoneme probability distributions
Combine identical phoneme predictions
Discriminator
real or fake

Unsupervised learning (GAN)

Original GAN with gradient penalty to stablize training

$$\mathcal{L}_{gp} = \mathop{\mathbb{E}}_{\tilde{P} \sim \tilde{\mathcal{P}}} \left[\left(\|\nabla \mathcal{C}(\tilde{P})\| - 1 \right)^2 \right]$$

segment smoothness penalty

$$\mathcal{L}_{sp} = \sum_{(p_t, p_{t+1}) \in \mathcal{G}(S)} \|p_t - p_{t+1}\|^2$$

phoneme diversity

$$\mathcal{L}_{pd} = \frac{1}{|B|} \sum_{S \in B} -H_{\mathcal{G}}(\mathcal{G}(S))$$

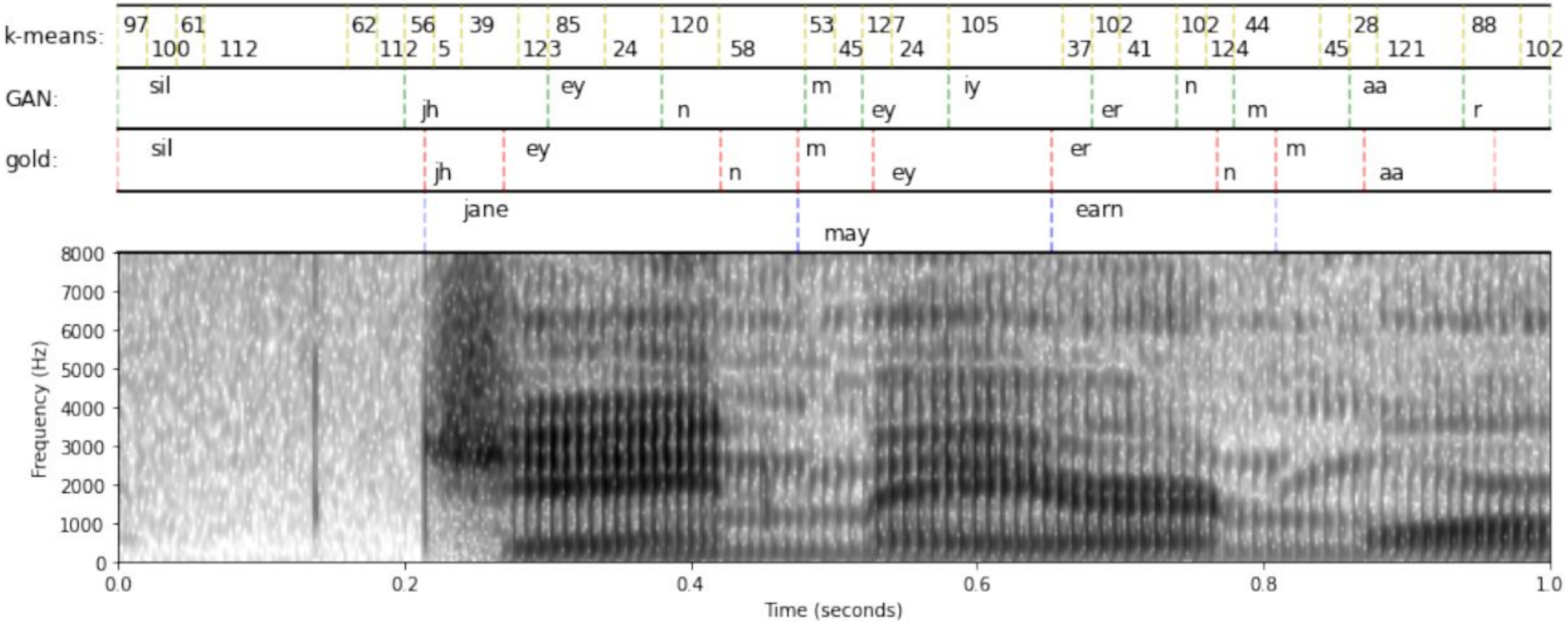
$$\min_{\mathcal{G}} \max_{\mathcal{C}} \mathop{\mathbb{E}}_{P^r \sim \mathcal{P}^r} [\log \mathcal{C}(\underbrace{P^r}_{\text{real phn seq}})] + \mathop{\mathbb{E}}_{S \sim \mathcal{S}} [\log (1 - \mathcal{C}(\underbrace{\mathcal{G}(S)}_{\text{generated phn seq}}))] - \lambda \mathcal{L}_{gp} + \gamma \mathcal{L}_{sp} + \eta \mathcal{L}_{pd}$$

Self-training

- HMM
- finetuning by pseudo-label

Model	LM	core-dev	core-test	all-test
wav2vec-U	4-gram	17.0	17.8	16.6
+ HMM	4-gram	13.7	14.6	13.5
+ HMM + HMM	4-gram	13.3	14.1	13.4
+ HMM resegment + GAN	4-gram	13.6	14.4	13.8
+ fine-tune	4-gram	12.0	12.7	12.1
+ fine-tune	-	12.1	12.8	12.0
+ fine-tune + fine-tune	-	12.0	12.7	12.0
+ HMM + fine-tune	-	11.3	11.9	11.3
+ HMM + fine-tune	4-gram	11.3	12.0	11.3

Result



k-means:
97 100 61 112 62 112 56 5 39 123 85 24 120 58 53 45 127 24 105 37 102 41 102 124 44 45 28 121 88 102
GAN:
sil jh ey n m ey iy er n m aa r
gold:
sil jh ey n m ey er n m aa
jane may earn
Frequency (Hz)
Time (seconds)

Model	Unlabeled data	LM	dev		test	
			clean	other	clean	other
960h - Supervised learning						
DeepSpeech 2 (Amodei et al., 2016)	-	5-gram	-	-	5.33	13.25
Fully Conv (Zeghidour et al., 2018)	-	ConvLM	3.08	9.94	3.26	10.47
TDNN+Kaldi (Xu et al., 2018)	-	4-gram	2.71	7.37	3.12	7.63
SpecAugment (Park et al., 2019)	-	-	-	-	2.8	6.8
SpecAugment (Park et al., 2019)	-	RNN	-	-	2.5	5.8
ContextNet (Han et al., 2020)	-	LSTM	1.9	3.9	1.9	4.1
Conformer (Gulati et al., 2020)	-	LSTM	2.1	4.3	1.9	3.9
960h - Self and semi-supervised learning						
Transf. + PL (Synnaeve et al., 2020)	LL-60k	CLM+Transf.	2.00	3.65	2.09	4.11
IPL (Xu et al., 2020b)	LL-60k	4-gram+Transf.	1.85	3.26	2.10	4.01
NST (Park et al., 2020)	LL-60k	LSTM	1.6	3.4	1.7	3.4
wav2vec 2.0 (Baevski et al., 2020c)	LL-60k	Transf.	1.6	3.0	1.8	3.3
wav2vec 2.0 + NST (Zhang et al., 2020b)	LL-60k	LSTM	1.3	2.6	1.4	2.6
Unsupervised learning						
wav2vec-U Large	LL-60k	4-gram	13.3	15.1	13.8	18.0
wav2vec-U Large + ST	LL-60k	4-gram	3.4	6.0	3.8	6.5
	LL-60k	Transf.	3.2	5.5	3.4	5.9

Homework

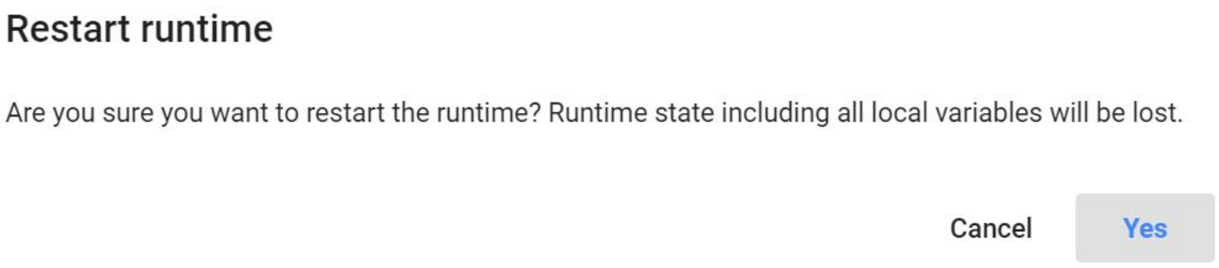
Colab link:
https://colab.research.google.com/drive/15lFjIFxwtYVuF-SGVIRPaXrT-PkZuyg-?usp=sharing

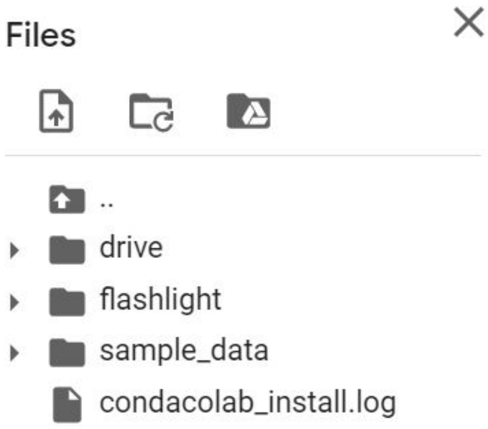
You only need to do:

1. ~~Pre-processing speech and text data~~
2. **unsupervised learning (GAN)**
3. ~~self-training~~

Note

- w2vu requires lots of environment dependencies, you need to install those packages for about 20 mins.
- when installation, sometimes the ‘Files’ section cannot display as normal, you just need to ‘refresh’ the webpage and the ‘Files’ section would be back.
- after installing **conda** and **flashlight,** you have to restart the runtime.





Problem 1: Out-of-domain text

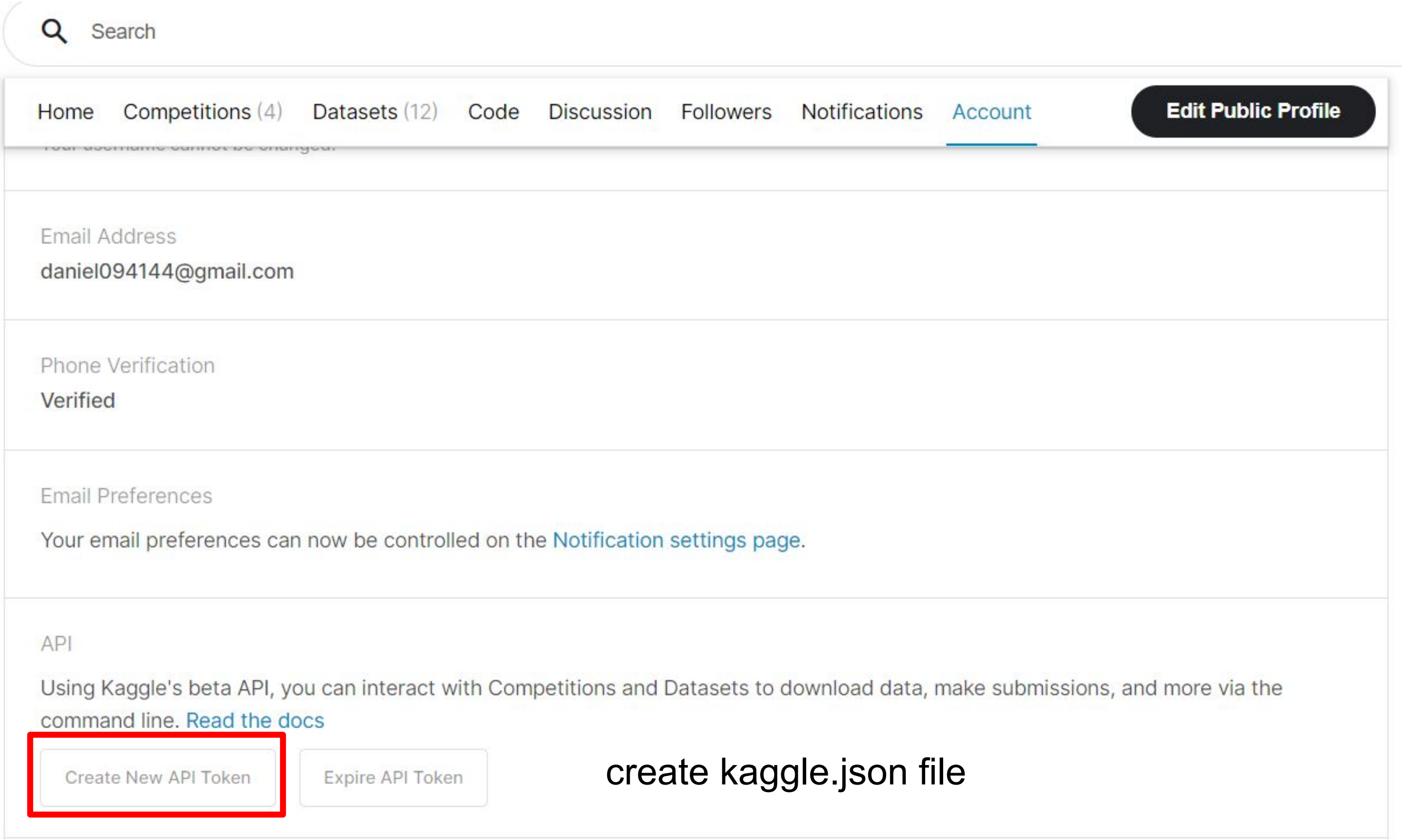
- **Training robustness of w2vu:** In w2vu, the speech and text data is from the same domain (audiobook), what if the speech and text have domain mismatched problem?

- Speech:
  - Librispeech 9.6 hours subset
- Text:
  - Librispeech LM
  - wiki
  - Image caption

Using different content of text can still learn how to map speech and phoneme?

Download prepared text data

by kaggle



create kaggle.json file

```python
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
  print('User uploaded file "{name}" with length {length} bytes'.format(
      name=fn, length=len(uploaded[fn])))

# Then move kaggle.json into the folder where the API expects to find it.
!mkdir -p ~/.kaggle/ && mv kaggle.json ~/.kaggle/ && chmod 600 ~/.kaggle/kaggle.json
```

Choose Files No file chosen Cancel upload

download kaggle dataset by:

kaggle datasets download <user/dataset_name>

Training

GAN training

- Finally, we resolve the environment settings and download the pre-processed data.
- Before training, you can modify the config file in `config/gan/w2vu` to test different hyperparameters and model architectures.

```
%cd /content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/
!export PREFIX=w2v_unsup_gan_xp

!PREFIX=$PREFIX fairseq-hydra-train \
    -m --config-dir ${FAIRSEQ_ROOT}/examples/wav2vec/unsupervised/config/gan \
    --config-name w2vu \
    dataset.num_workers=0 \
    task.data=${FAIRSEQ_ROOT}/examples/wav2vec/unsupervised/Libri_small \
    task.text_data=${FAIRSEQ_ROOT}/examples/wav2vec/unsupervised/librilm/phones \
    task.kenlm_path=${FAIRSEQ_ROOT}/examples/wav2vec/unsupervised/librilm/phones/lm.phones.filtered.04.bin \
    common.user_dir=${PWD} \
    model.code_penalty=4 model.gradient_penalty=2.0 \
    model.smoothness_weight=0.5 \
    common.seed=0 \
    distributed_training.distributed_world_size=1 \
    dataset.batch_size=32
```

speech data

text data

text data langauge model

other hyperparameters

REF: N AO R W AA Z DH IH S IH G Z AE K T L IY DH AH SH EY P DH
HYP: T HH IY HH AH N ER N S AY B IH Z D AE D B EH R AH S ER S

REF: N AO R W AA Z DH IH S IH G Z AE K T L IY DH AH SH EY P DH
HYP: AY N AY B Z EY S IH K S AE K IH DH AH TH F AO R P DH AH DH

REF: N AO R W AA Z DH IH S IH G Z AE K T L IY DH AH SH EY P DH
HYP: DH AH AO R W AA Z IH S IH G Z AE K L IY DH AH SH EY P DH

training
step

Evaluation

Evaluation

```
%cd /content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/
!export TASK_DATA=${PWD}/Libri_small
!export exp_name=libri_libriLM
!export HYDRA_FULL_ERROR=1
# copy text data into TASK_DATA
!cp ${FAIRSEQ_ROOT}/examples/wav2vec/unsupervised/librilm/phones/dict.phn.txt /content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/Libri_small

!python w2vu_generate.py --config-dir config/generate --config-name viterbi \
    fairseq.common.user_dir=/content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/ \
    fairseq.task.data=/content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/Libri_small \
    fairseq.common_eval.path=/content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/multirun/2021-11-16/15-11-18/0/checkpoint_686_61000.pt \
    fairseq.dataset.gen_subset=valid results_path=$/content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/results \
```

specify speech data dir and your checkpoint path

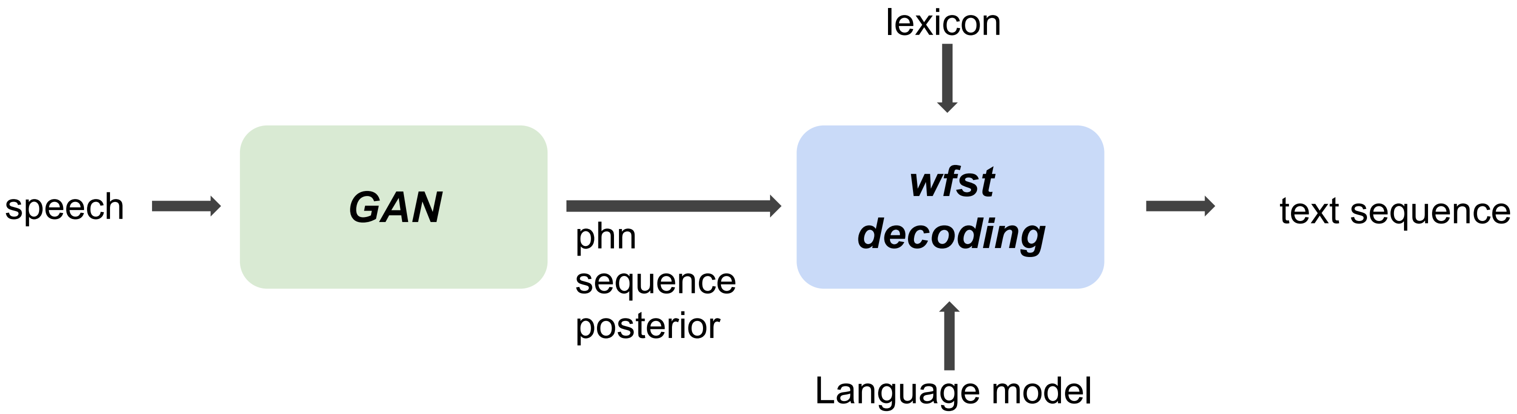
```
/content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised
2021-11-18 11:06:42 | INFO | fairseq.tasks.text_to_speech | Please install tensorboardX: pip install tensorboardX
[2021-11-18 11:07:48,397][__main__][INFO] - {'_name': None, 'fairseq': {'_name': None, 'common': {'_name': None, 'no_progress_bar': False, 'log_interval':
[2021-11-18 11:07:51,208][__main__][INFO] - | loading model(s) from /content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/multirun/2021-11-16/15-11
[2021-11-18 11:08:11,024][unsupervised.data.extracted_features_dataset][INFO] - loaded 2703, skipped 0 samples
[2021-11-18 11:08:11,025][unsupervised.tasks.unpaired_audio_text][INFO] - split valid has unpaired text? False
[2021-11-18 11:08:11,026][__main__][INFO] - | /content/drive/MyDrive/fairseq/examples/wav2vec/unsupervised/Libri_small valid 2703 examples
/content/drive/MyDrive/fairseq/examples/speech_recognition/w2l_decoder.py:43: UserWarning: flashlight python bindings are required to use this functionali
  "flashlight python bindings are required to use this functionality. Please install from https://github.com/facebookresearch/flashlight/tree/master/bindi
[2021-11-18 11:08:51,263][__main__][INFO] - WER: 23.705126403397824
[2021-11-18 11:08:51,264][__main__][INFO] - | Processed 2703 sentences (181568 tokens) in 39.2s (68.96 sentences/s, 4632.33 tokens/s)
[2021-11-18 11:08:51,265][__main__][INFO] - | Generate valid with beam=5, lm_weight=2.0, word_score=1.0, sil_weight=0.0, blank_weight=0.0, WER: 23.7051264
```

Problem 2: Modify GAN

- Use the **libri9.6 (speech) + libriLM (text)** pair for below experiment.
- In *fairseq/examples/wav2vec/unsupervised/config/gan/w2vu.yaml*
- you can play with:
  - learning rate
  - size of kernel, dimension, depth
  - loss weight
  - you can even modify GAN source code in (WGAN?) *fairseq/examples/wav2vec/unsupervised/models/wav2vec_u.py*

Beyond unsupervised phoneme recogition?

Directly output char output instead of phoneme



Q & A

有任何問題可以直接在Fackbook社團投影片下方留言討論

Further reading list

1. **“Unsupervised speech recognition”**, Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and MichaelAuli
2. **“Completely unsupervised phoneme recognition by adversari-ally learning mapping relationships from audio embeddings”**, Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee
3. **“Completely unsupervised speech recogni-tion by a generative adversarial network harmonized with iter-atively refined hidden markov models”**, Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-Yi Lee,and Lin shan Lee
4. **“Unsupervised automatic speech recognition: A review”**, Hanan Aldarmaki, Asad Ullah, and Nazar Zak