JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

CENTER FOR LANGUAGE
AND SPEECH PROCESSING

ESPnet: End-to-end speech processing toolkit

Shinji Watanabe
Center for Language and Speech Processing
Johns Hopkins University

Joint work with Takaaki Hori , Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, Tsubasa Ochiai

Lab instruction

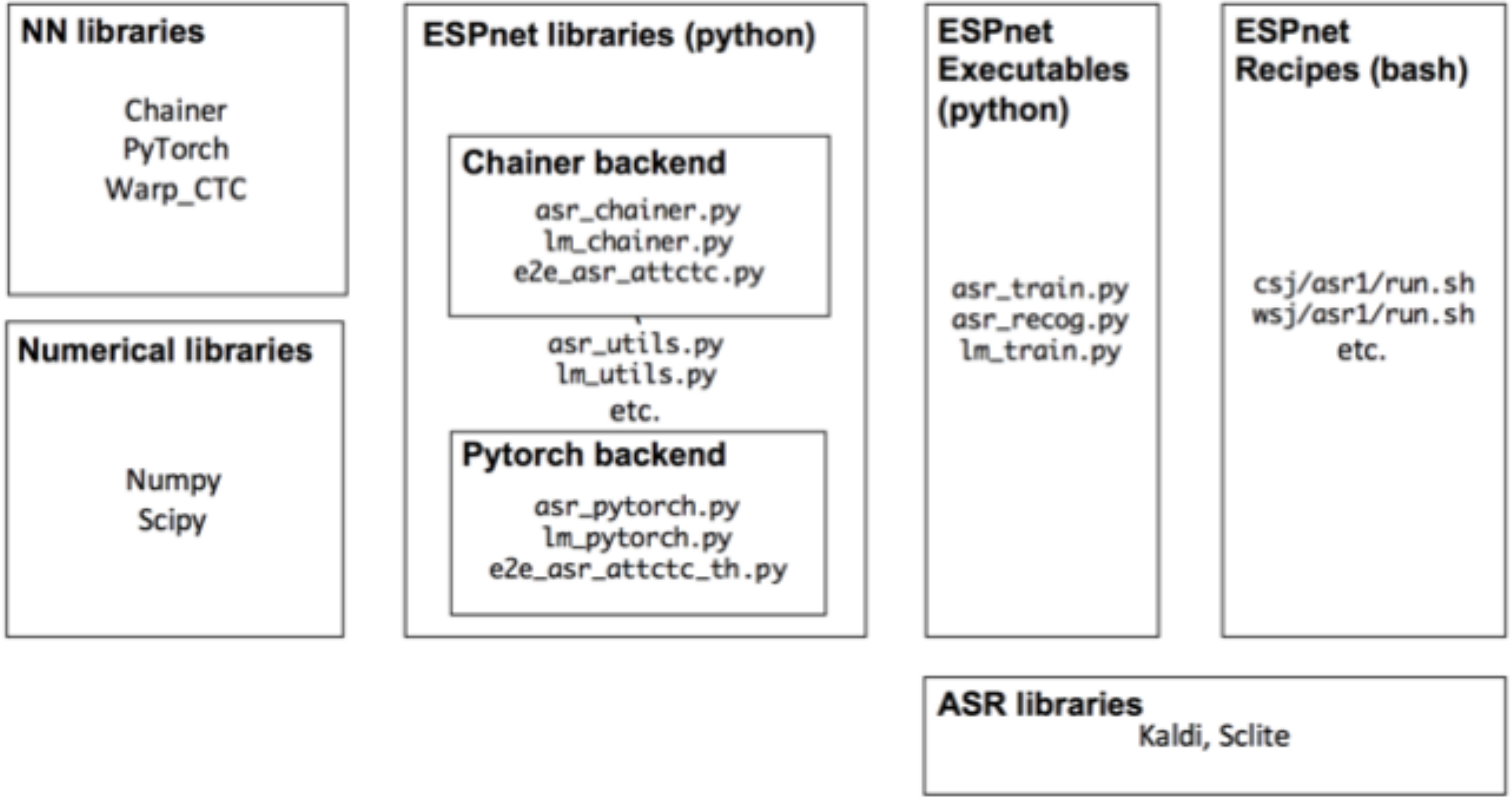
- https://hackmd.io/s/rJ6TDZPeQ

ESPnet

https://github.com/espnet/espnet

- Open source (Apache2.0) end-to-end ASR toolkit
  - Developed for the 2018 JSALT workshop “Multilingual End-to-end ASR for Incomplete Data”
- Actively developed by researchers all over the world (JHU, MERL, Nagoya Univ., NTT, Paderborn Univ., PFN, …)
- Chainer or Pytorch backend
- Follows the Kaldi style
  - Data processing
  - Feature extraction/format
  - Recipes to provide a complete setup for speech recognition and other speech processing experiments

Software architecture



NN libraries

Chainer
PyTorch
Warp_CTC

Numerical libraries

Numpy
Scipy

ESPnet libraries (python)

Chainer backend

asr_chainer.py
lm_chainer.py
e2e_asr_attctc.py

asr_utils.py
lm_utils.py
etc.

Pytorch backend

asr_pytorch.py
lm_pytorch.py
e2e_asr_attctc_th.py

ESPnet Executables (python)

asr_train.py
asr_recog.py
lm_train.py

ESPnet Recipes (bash)

csj/asr1/run.sh
wsj/asr1/run.sh
etc.

ASR libraries

Kaldi, Sclite

Functionalities

- Kaldi style data preprocessing
    1) fairly compare the performance obtained by Kaldi hybrid systems
    2) make use of data preprocessing developed in the Kaldi recipe
- Attention-based encoder-decoder
    - Subsampled BLSTM and/or VGG-like encoder
    - location-based attention (+10 attentions)
- CTC
    - WarpCTC, label-synchronous decoding
- Hybrid CTC/attention
    - Multitask learning, joint decoding
- Use of language models
    - Combination of RNNLM and label-synchronous hybrid CTC/attention decoding

Backends

- Use Chainer and PyTorch

	Chainer	**PyTorch**
Performance	◎	○
Speed	○	◎
Multi-GPU	supported	supported
VGG-like encoder	supported	no support
RNNLM integration	supported	supported
#Attention types	3 (no attention, dot, location)	12 including variants of multihead

Lines of code, etc.

- Kaldi

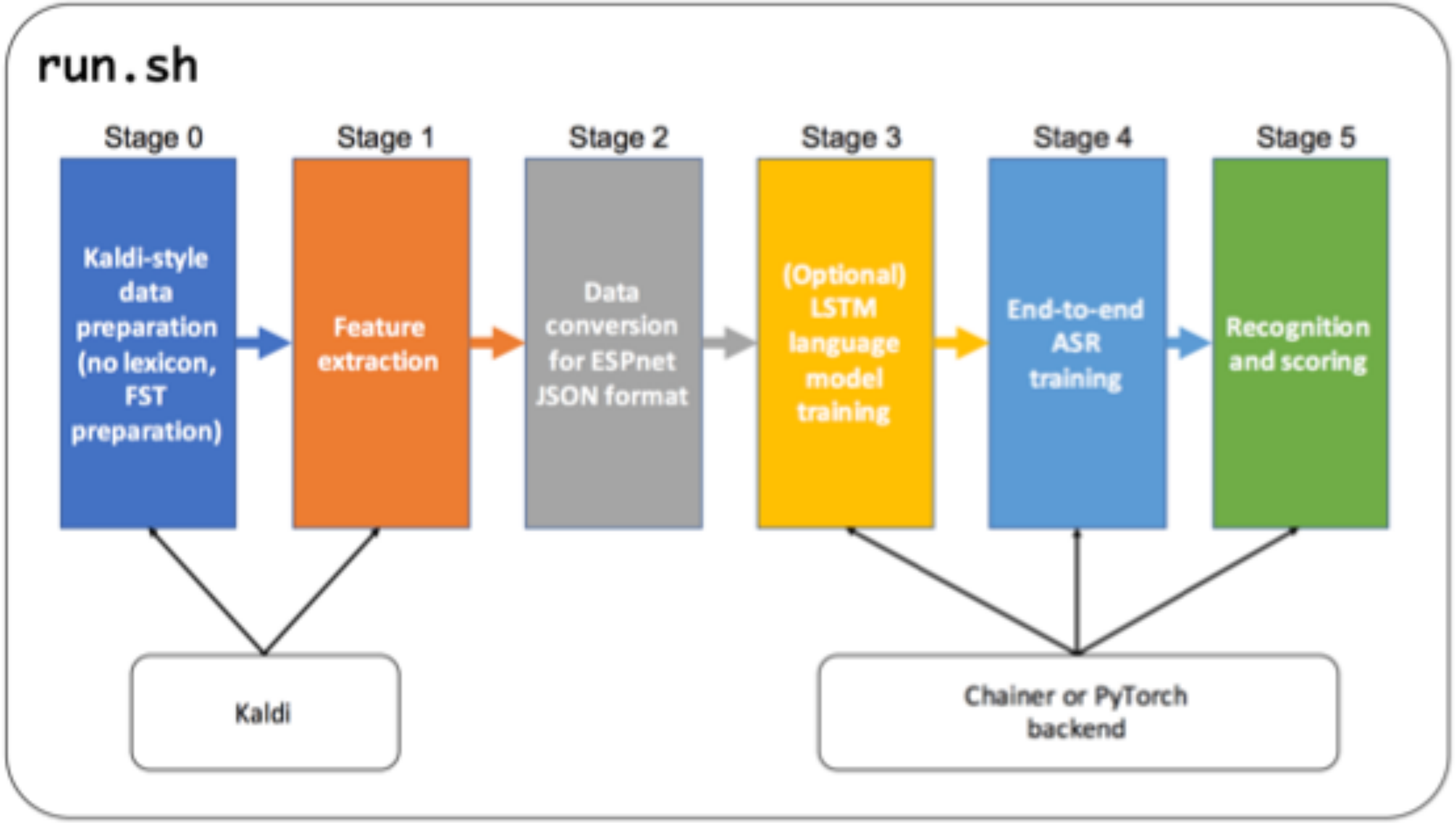
```
$ cat kaldi/src/*/*.{cc,cu,h} | wc –l
~330k
```

- ESPnet

```
$ cat espnet/src/*/*.{sh,py} | wc –l
~6.9k
```

  - Chainer/Pytorch as a main deep learning engine
  - Use Kaldi feature extraction, and python-based reader/writer

Basic flow of recipes



run.sh
Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Kaldi-style data preparation (no lexicon, FST preparation)
Feature extraction
Data conversion for ESPnet JSON format
(Optional) LSTM language model training
End-to-end ASR training
Recognition and scoring
Kaldi
Chainer or PyTorch backend

- Very simple flow
  - No Gaussian construction
  - No FST
  - No alignments
  - No lattice outpts
- Easily ported from existing Kaldi recipes

Supported recipes (15 recipes)

- ami
- an4
- babel
- chime4
- chime5
- csj
- fisher_swbd
- hkust
- hub4_Spanish
- librispeech
- li10
- swbd
- tedlium
- voxforge
- wsj

Supported languages (25 languages)

Major English tasks (WSJ, Fisher+Switchboard, Librispeech)

Japanese (Corpus of Spontaneous Japanese)

Mandarin (HKUST CTS)

Babel 15 languages

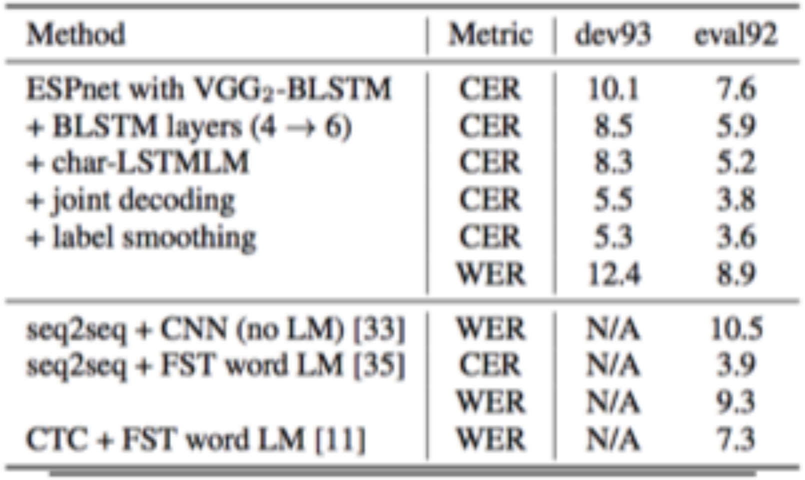
- Cantonese , Assamese, Bengali, Pashto, Turkish, Georgian, Tagalog , Vietnamese , Haitian, Swahili, Lao, Tamil, Zulu, Kurmanji Kurdish, Tok Pisin

VoxForge 7 languages

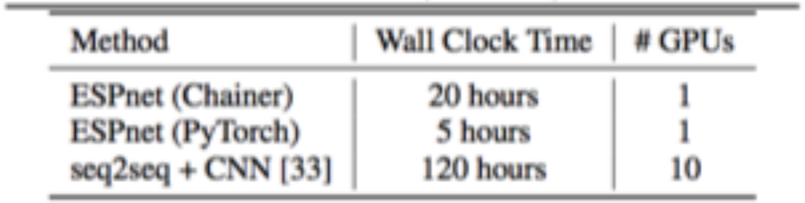
- German, Spanish, French, Italian, Portuguese, Russian, Dutch

Performance

- WSJ

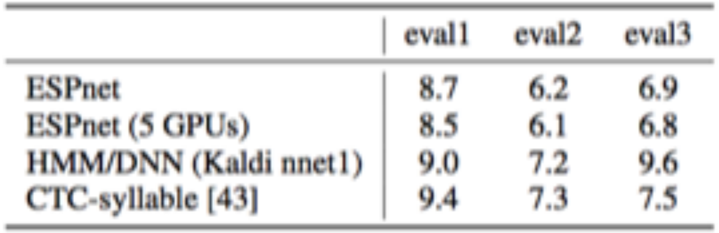


Method	Metric	dev93	eval92
ESPnet with VGG$_2$-BLSTM	CER	10.1	7.6
+ BLSTM layers (4 → 6)	CER	8.5	5.9
+ char-LSTMLM	CER	8.3	5.2
+ joint decoding	CER	5.5	3.8
+ label smoothing	CER	5.3	3.6
	WER	12.4	8.9
seq2seq + CNN (no LM) [33]	WER	N/A	10.5
seq2seq + FST word LM [35]	CER	N/A	3.9
	WER	N/A	9.3
CTC + FST word LM [11]	WER	N/A	7.3



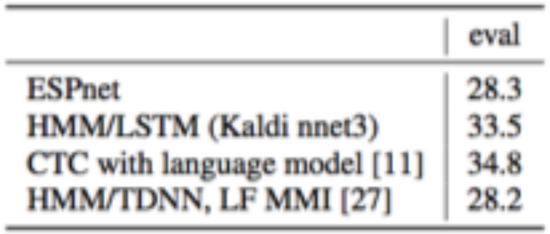
Method	Wall Clock Time	# GPUs
ESPnet (Chainer)	20 hours	1
ESPnet (PyTorch)	5 hours	1
seq2seq + CNN [33]	120 hours	10

- CSJ



	eval1	eval2	eval3
ESPnet	8.7	6.2	6.9
ESPnet (5 GPUs)	8.5	6.1	6.8
HMM/DNN (Kaldi nnet1)	9.0	7.2	9.6
CTC-syllable [43]	9.4	7.3	7.5

- HKUST



	eval
ESPnet	28.3
HMM/LSTM (Kaldi nnet3)	33.5
CTC with language model [11]	34.8
HMM/TDNN, LF MMI [27]	28.2

Summary

- Easy to develop, easy to perform experiments
  - Thanks to the simplification of end-to-end ASR and recent developments of deep learning toolkits
- Multilingual functions
  - Make use of end-to-end ASR benefits
- Good performance
  - Moderate ASR performance on English benchmarks
  - State-of-the-art ASR performance on ideogram languages (Japanese and Mandarin)
  - Fast training
- Future development plans
  - Stabilities, faster training/recognition, performance improvement
  - Multi-*** (multilingual, multispeaker, multichannel, multimodal, etc.)

Lab instruction

- https://hackmd.io/s/rJ6TDZPeQ