Paper

Target: Investigation on how to improve the performance of LID system under low-resource condition.

Approach

- **Data augmentation**
  - 2-fold: superimposed
  - 5-fold: combined
  - **Conclusion:** 2-fold may have been corrupted by noise due to raw data. And 5-fold is the better choice for LID.
- **Language-aware training**
  - Single-language BNFs
    - EN bnfs: only en bnfs
    - CN bnfs: only cn bnfs
    - **Conclusion:** It solves greatly the low-resource problem.
    - Which layer is best?
  - Multi-language BNFs
    - Feature level fusion
      - append(enbnf,cnbnf)
      - no append directly

        iVector: Two inputs are PCA respectively, then Append

        xVector: 2 input, and xVector shared
    - Score level fusion
    - **Conclusion:** Now, we know that score level fusion is better than single-languag BNFs

Data

- **Training data (10 languages)**
  - train_25h
  - train_50h
  - train_75h
  - train_106h
- **Test data**
  - in-domain data (10 languages)
  - out-of-domain (6 languages)

Experiment

1. Baseline

Confirm the problem of low-resource for LID task

Incremental learning:

train_25h ->train_50h ->train_75h ->train_106h

Result:

system	in-domain				out-of-domain			
	25h	50h	75h	106h	25h	50h	75h	106h
iVec_mfcc_lr	71.43	84.00	88.05	90.62	31.94	37.28	40.15	37.51
xVec_fbank_lr	61.70	76.76	82.87	86.34	31.05	35.56	37.76	35.48

Table 1 Comparing the accuracy of different durations of training sets in in-domain and out-of-domain. All systems conform to the fixed training condition.

Conclusion:

In Table 1, we find that the smaller the amount of training data, the lower the accuracy in the in-domain. And the accuracy of out-of-domain is much lower than in-domain on same training data. Overall, in xVec_fbank_lr, the accuracy of in-domain in train_106h is 64.04% which is better than out-of-domain in train_25h. The experimental results above demonstrate the influence of low-resource on the accuracy of language recognition.

2. Data augmentation

We use augmentation to increase the amount and diversity of the language system training data.

25h+ * 训练数据 -> 50h+ * 训练数据 -> 75h+ * 训练数据 -> 106h+ * 训练数据

We use two ways of data augmentation.

One is superimposed, which consists of 2-fold augmentation that combines the original “clear”training data with 1 mixed noise of multiple noises.

The other is combined, which consists of 5-fold augmentation that combines the original“clean”training data with 4 copies of augmented data.

Result:

system	in-domain				Out-of-domain			
	25h	50h	75h	106h	25h	50h	75h	106h
iVec_mfcc_2f_lr	69.89	76.46	87.83	89.25	25.23	31.27	44.94	46.74
iVec_mfcc_5f_lr	**72.52**	**86.54**	**90.09**	**91.83**	**43.31**	**44.61**	**44.86**	**45.36**

xVec_fbank_2f_lr	60.51	75.98	80.12	83.31	25.12	28.89	35.29	37.35
xVec_fbank_5f_lr	**62.05**	**76.89**	**83.07**	**89.89**	**33.57**	**36.43**	**37.97**	**43.73**

Table 2 Comparing the accuracy of different data augmentation on iVector/xVector (with Table 1)

2f: 2-fold superimposed augmentation

5f: 5-fold combined augmentation

Conclusion:

In Table 2, we observe that augmentation using 2-fold significantly degrades in in-domain, which may have been corrupted by noise due to raw data. And comparing with the 5-fold, removing augmentation degrades performance significantly. Due to 5-fold augmentation increasing the limited amount of training data, the system is more robust against degraded audio. 5-fold augmentation is good for LID low-resource task, whether it is on iVector system or xVector system.

3. Language-aware training

With the help of ASR system to feed phonetic information.

3.1 Single-language

We use two ways of ASR model. One is English ASR model, 1300h of training data is used. The other is Chinese ASR model, 3000h of training data is used.

It is worth noting that we reduced the BNFs from 256-dim to 40-dim by PCA in the iVector system.

Result1:

system	in-domain				out-of-domain			
	25h	50h	75h	106h	25h	50h	75h	106h
iVec_enbnf_lr	93.62	95.38	97.38	97.61	**71.24**	**73.90**	**76.72**	**77.23**
xVec_enbnf_lr	93.72	97.66	98.31	98.41	59.13	60.78	61.02	64.22
iVec_cnbnf_lr	**96.29**	**97.12**	**98.27**	**98.41**	67.22	69.30	70.80	71.70
xVec_cnbnf_lr	**96.81**	**98.53**	**98.65**	**98.91**	**64.51**	**64.53**	**66.40**	**68.99**

Table 3 Comparing Single-language BNFs with original. (with Table 1)

enbnf: bnfs extracted from EN ASR model

cnbnf: bnfs extracted from CN ASR model

Conclusion:

In Table 3, we observed that single-language BNFs is beneficial to both iVector system and xVector system. And Single-language BNFs solves greatly the low-resource problem.

We also compare the effects of same extraction layers of different language models to the LID. It shows that the performance of Chinese ASR model is better than the English ASR model. Because the Chinese ASR model has more training data and the accuracy of phone recognizer is higher. The accuracy of phone recognizer is critical for LID task.

Result2:

System	layers	en		cn	
		in-domain	out-domain	in-domain	out-domain
iVector	output-xent.linear	93.62	**71.24**	**96.29**	**67.22**
	output.linear	94.53	67.05	96.2	64.67
	prefinal-l	**93.84**	71.12	96.61	65.89
	tdnn8l	88.53	63.38	90.1	66.51
xVector	output-xent.linear	93.72	**59.13**	**96.81**	**64.51**
	output.linear	94.04	56.5	96.51	53.87
	prefinal-l	**94.35**	55.09	96.53	57.85
	tdnn8l	88.96	45.25	89.83	40.6

Table 4 Comparing different layers when training data is 25h, under different ASR models

In Table 4, we also compare the effects of different extraction layers of different language models to the LID. It shows that BNFs extracted from output-xent.linear is best regardless of the ASR model in out-of-domain. And it also shows that the performance of Chinese ASR model is better than the English ASR model. The accuracy of phone recognizer is critical for LID task.

3.2 Multi-language BN

It might be possible that some language ASR systems attribute to one aspect of the language space.

Then why we do not combine the BN features from different ASR decoders?

system	in-domain				out-of-domain			
	25h	50h	75h	106h	25h	50h	75h	106h
iV-Feats-append-f	**96.46**	**98.83**	**98.94**	**99.08**	64.96	71.12	73.24	75.27
iV-Feats-f	**96.41**	**98.12**	**98.71**	**98.83**	63.05	70.05	72.13	73.66
iV-Score-f	**97.13**	**98.64**	**98.92**	**99.03**	**72.38**	**75.01**	**77.35**	**78.92**
xV-Feats-append-f	**96.95**	**98.67**	**99.01**	**99.56**	55.96	57.94	63.21	65.54
xV-Feats-f	93.88	97.58	98.30	98.50	58.68	61.60	63.63	64.19
xV-Score-f	**97.62**	**98.98**	**98.99**	**99.06**	**65.02**	**65.22**	**68.95**	**70.14**

Table 5 Comparing different layers in train_25h, under different ASR models (with Table 3)

Conclusion:

In Table 5, we find that Multi-language BNFs much better than single-language BNFs, due to the advantage and complementarity of universal speech attributes to language-dependent phonemes. We conducted experiments at the feature level and the score level. We fine that…. This approach is also beneficial to both iVector system and xVector system.