DNN based Embeddings for Language Recognition

Alicia Lozano-Diez[1,2], Oldrich Plchot[2], Pavel Matejka[2], Joaquin Gonzalez-Rodriguez[1]

[1]Audias-UAM, Universidad Autonoma de Madrid, Madrid, Spain

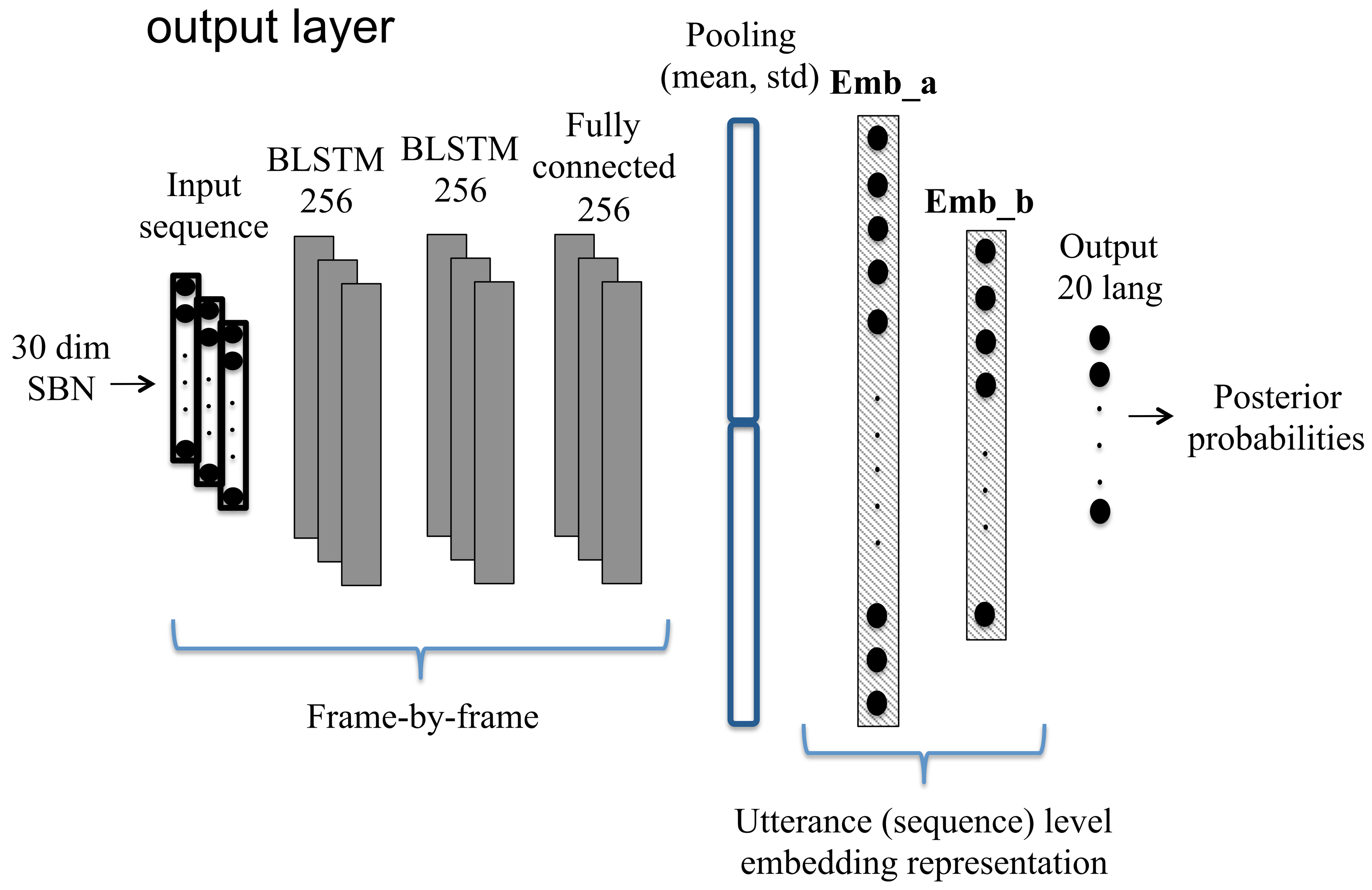
[2] Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czechia

Abstract

- **Bottleneck** features: **frame-wise** representation
  - From DNN trained for speech recognition (ASR)
  - Variable-length sequence of features → fixed-length **i-vector**
- **Embeddings**: **utterance-level** representation (**fixed-length**)
  - From DNN trained for language recognition (LID, target task)
  - Similar approach previously applied to speaker recognition

Proposed DNN-Embeddings

- **DNN** trained to **discriminate** between **languages**
- **Also** provides **fixed-length embeddings** that summarize the **whole utterance** with useful information about the language
- Proposed architecture:
  - **Frame level**: input + BLSTM + fully connected
  - **Pooling** layer (mean and std over time)
  - **Utterance level**: fully connected (embeddings) + softmax output layer



- **Input**: 30-dimensional stacked bottleneck features (SBN)
- Trained to optimize multi-class cross-entropy loss function
- 3s sequences for training, no length constraints for embedding extraction

Experimental Framework

NIST LRE 2015

- **20 languages** clustered into **six groups**
- **Fixed** condition (SBN extractor trained on Fisher English)
- 60% for training, 40% for development (cuts of 3-30s)
- ~248h for training (limited up to 15h per language for embedding system, ~144h)
- ~146h for validation
- 164334 test segments of different durations

Reference i-vector

- Same **input** features as for embedding system:
  - 30-dimensional **SBN** features
- Diagonal-covariance **UBM** with **2048** components
- **600**-dimensional **i-vectors**

LID Backend

- **Gaussian Linear Classifier** (GLC):
  - On top of i-vectors or embeddings
  - Outputs the vector of 20 class-conditional log-likelihoods for each segment
  - Model of each language: Gaussian distribution
    Mean: over i-vectors of given language
    Covariance matrix: shared across all models
- **Calibration** and score level **fusion**:
  - Multi-class logistic regression trained on top of the development scores

Experiments and Results

Original DNN-embedding

Different size of DNN-embeddings (concatenation emb_a + emb_b):
- Halving size of the embedding layers:
  - DNN_1: 512 + 300 = 812
  - DNN_2: 256 + 150 = 406
  - DNN_3: 128 + 75 = 203
- Comparison of performance:
  - Embeddings vs. posterior probabilities (directly from the DNN)

	C_{avg} x 100	
System	**Embedding (GLC)**	**Posteriors**
DNN_1	20.04	20.37
DNN_2	**19.19**	19.68
DNN_3	19.30	19.76

PCA post-processing

Study of **PCA post-processing of embeddings**, motivated by:
- Better results with smaller embeddings (DNN_2 and DNN_3 vs. DNN_1)
- First architecture (DNN_1) inspired by success on speaker recognition with larger number of classes (thousands of speaker vs. 20 languages)

	C_{avg} x 100		
System	**None**	**PCA 100**	**PCA 25**
DNN_1	20.04	18.67	19.98
DNN_2	19.19	18.11	**17.44**
DNN_3	19.30	18.70	18.13

Fusion with i-vector

Score-level fusion of **embedding** and reference **i-vector** systems:
- Reference i-vector (GLC)
- Best DNN-embeddings: DNN_2 (concatenated emb_a + emb_b) + PCA 25
- Best DNN posterior probabilities: DNN_2

System	C_{avg} **x 100**
(1) Ref. i-vector	16.93
(2) Best embeddings	17.44
(3) Best DNN posteriors	19.68
Fusion (1) + (2)	**15.69**
Fusion (1) + (3)	16.41

Conclusions

- Proposed **DNN-embedding** system for **LID**
- Embedding: **fixed-length utterance-level** representation, provided by a DNN trained for the target task (LID)
- **Novel** approach for LID (in line with research in speaker ID)
- Results comparable with state-of-the-art i-vector system
  - Up to 7.3% relative improvement with simple fusion
- Better results with embeddings than posteriors from the DNN, possibility for **more general DNNs** usable across LID tasks

Selected References

[1] Snyder et al.: Deep neural network embeddings for text-independent speaker verification. Proceedings of Interspeech 2017.
[2] Dehak et al.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech & Language Processing 19(4), 788{798 (2011).
[3] Frantisek et al.: Investigation into bottleneck features for meeting speech Recognition. Proceedings of Interspeech 2009.
[4] "The 2015 NIST Language Recognition Evaluation Plan (LRE15), "http://www.nist.gov/itl/iad/mig/upload/LRE15 EvalPlan v23.pdf.
[5] Cumani et al.: Exploiting ivector posterior covariances for short-duration language recognition. Proceedings of Interspeech 2015.