Spoken language identification using language bottleneck features

Malo Grisard *†, Petr Motlicek ‡, Wissem Allouchi †, Michael Baeriswyl †,
Alexandros Lazaridis ‡, and Qingran Zhan ‡

* EPFL, Department of Electrical Engineering, Lausanne, Switzerland*
† Artificial Intelligence and Machine Learning Group, Swisscom, Switzerland
‡ Idiap Research Institute, Martigny, Switzerland

Abstract. In this paper, we introduce a novel approach for Language Identification (LID). Two commonly used state-of-the-art methods based on UBM/GMM I-vector technique, combined with a back-end classifier, are first evaluated. The differential factor between these two methods is the deployment of input features to train the UBM/GMM models: conventional MFCCs, or deep Bottleneck Features (BNF) extracted from a neural network. Analogous to successful algorithms developed for speaker recognition tasks, this paper proposes to train the BNF classifier directly on language targets rather than using conventional phone targets (i.e. international phone alphabet). We show that the proposed approach reduces the number of targets by 96% when tested on 4 languages of SpeechDat databases, which leads to 94% reduction in training time (i.e. to train BNF classifier). We achieve in average, relative improvement of approximately 35% in terms of cost average C_{avg}, as well as Language Error Rates (LER), across all test duration conditions.

Keywords: Bottleneck features, Language identification, Language targets, Deep neural network

1 Introduction

Language Identification (LID) is the task of automatically recognizing the language that is being spoken. This task can be carried out using different levels of language representations, whether it can be phones, words, or universal speech features. In the early 1990s, phonotactic models had been proposed, exploiting phone-based acoustic likelihood ratios [8] to identify the language. In the recent years, many acoustic approaches have been proposed, or borrowed especially from Speaker Recognition (SR), often based on Universal Background Model (UBM)/Gaussian Mixture Model (GMM) framework [12]. One of the best performing LID, established today as one of a baseline systems, is based on UBM/GMM I-vector approach [4, 1, 9]. Nonetheless, the growth in available computational power has shifted the focus, in language identification domain as well, towards neural networks based approaches.

In automatic speech recognition, neural networks have become a widely used technique rapidly expanding to other fields of speech processing. Proposed bottleneck features (i.e. BNF vectors) [7] extracted from a narrow layer of neural network have shown

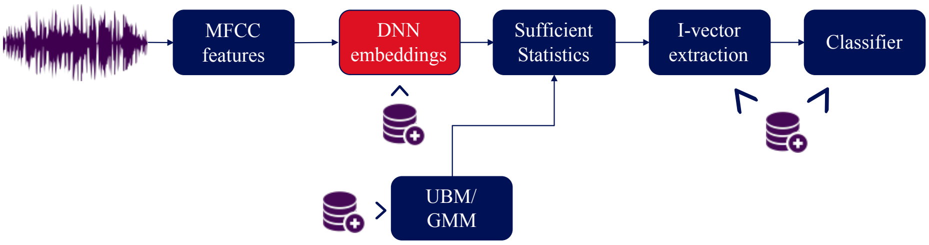

Fig. 1. UBM/GMM$_L$-IV-LR system for language identification. UBM, I-vector extractor and back-end classifier (e.g. logistic regression) are data-driven blocks.

to convey information about phonetic content in a non-linearly compressed form, which can be directly used as features for GMM modelling. In LID, this approach has shown a remarkable improvement over UBM/GMM I-vector, proposing a linear bottleneck (i.e. phone embedding) layer produced by Deep Neural Networks (DNNs) [10, 3, 11]. DNNs were trained on phone targets using a small hidden layer representing a phone-embedding (i.e. BNFs). In [11], authors have shown that BNFs when employed as an input to a UBM/GMM i-vector based system, significantly outperform conventional MFCCs on NIST LRE 2007 task [15]. Long Short Term Memory (LSTM) networks were also proposed to take better advantage of the temporal information incorporated in a speech segment [5]. In SR, the aforementioned approach was later adapted by inserting a temporal pooling layer into the network to handle variable-length segments [14]. Inspired by previous work in the LID and SR domains, this paper proposes to extract embedding vectors from a DNN directly trained on language-targets. More specifically we investigate building an embeddings space which can incorporate more information for each language. We hypothesize that extracting BNFs by using phone-embeddings is a sub-optimal approach in language identification. Instead, we presume that the extracted language-embeddings will be more representative for LID task. In practice, when building a DNN to extract bottleneck features, phone targets are replaced by language targets. Additionally. replacing the phone-targets at the output of DNN by language-targets brings a significant reduction in training time.

The remaining of this paper is organized as follows: Section 2 presents related work in language identification, together with the baseline LID systems considered in this paper. Section 3 presents the proposed work, while Section 4 describes our experimental setup. The results are discussed in Section 5 and conclusions are given in Section 6.

2 Related work

The traditional architecture applied in the LID task is shown in Figure 1. First, statistics from the input speech features are extracted at a frame-level. A DNN is then trained to extract phoneme-embeddings [10], [3], [11]. The embedding vectors estimated at the frame-level are then projected on an acoustic space to train a UBM/GMM model. An I-vector extractor is then trained to extract the relevant information from speech

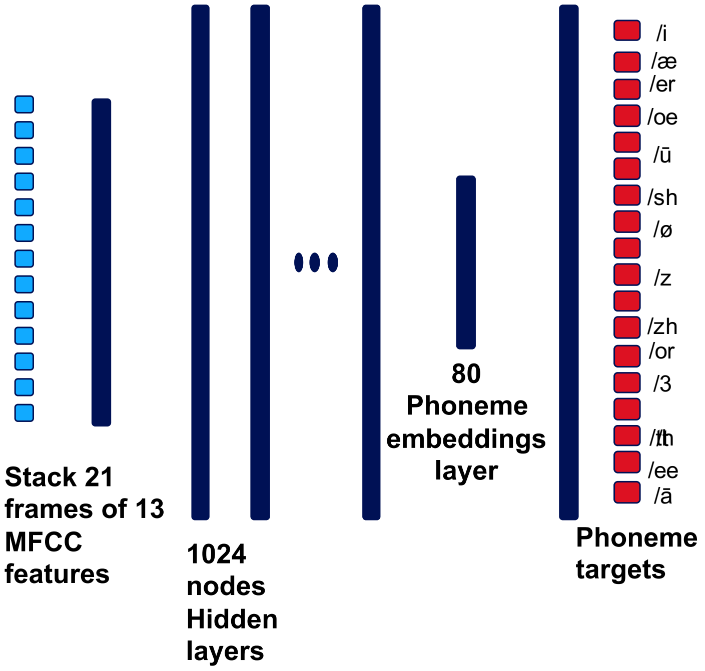

Fig. 2. DNN architecture to extract embedding (BNF) vectors. The neural net is trained on phone targets, thus extracting low dimensional phone embeddings.

(i.e. projecting a variable-length speech to a fixed low dimensional vector). The low dimensional I-vectors are finally fed to a back-end classifier to detect the language.

2.1 BNF extraction

Bottleneck features (BNF) trained in the following discriminative framework are used to represent the acoustic space of speech. A DNN model is used, employing seven hidden layers, trained using conventional MFCC features. The DNN is trained to discriminate the phone targets as this is the case in acoustic modelling in Automatic Speech Recognition (ASR) tasks. The weights of the compressed hidden layer of the DNN are considered as phone embedding representations, which are estimated at a frame-level and further applied to train a UBM/GMM model. The architecture of the DNN is shown in Figure 2.

2.2 UBM/GMM

A UBM is built using the input features to represent the acoustic space of the speech [12]. The UBM is represented by a large number of Gaussian mixtures. It is expected to cover the acoustic space across all languages, if sufficient amount of training data is used.

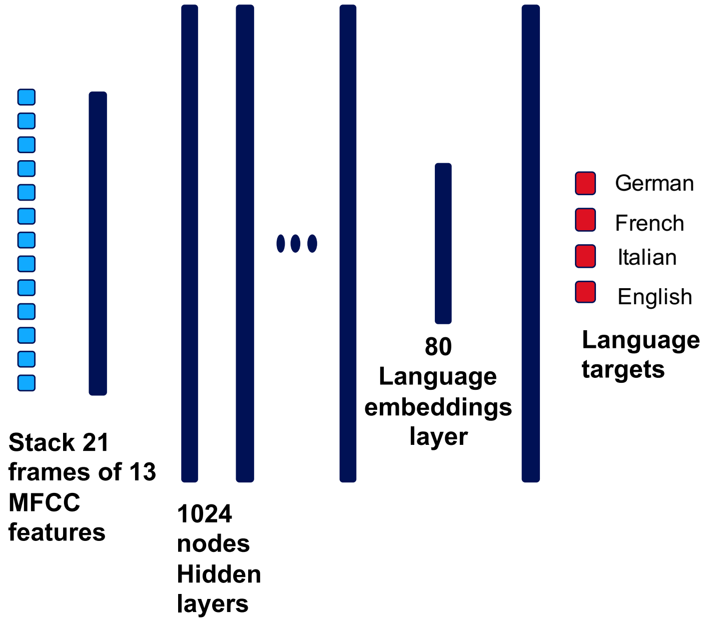

Fig. 3. Proposed DNN architecture used to extract embedding (BNF) vectors. Here, the network is trained on language targets to output a language embedding allowing the network to discriminate among languages directly.

2.3 I-vector extractor

The I-vector or total variability space approach is a technique borrowed from the speaker recognition field [1]. It consists of a mapping of a sequence of frames of speech into a low-dimensional vector space, i.e., the total variability space. The motivation for the use of I-vectors in speech is to convey speech information of variable length into a fixed-length feature vector. Unlike joint factor analysis [6], the I-vector approach models all important variability (language, speaker, channel, etc.) in the same fixed dimensional space.

2.4 Language identification

Most of the related work in LID has focused on extracting features representative of the language fed to a back-end LID classifier to predict the spoken language. Since LID is a closed-set classification task (i.e. with limited number of classes), there is no need to apply Probabilistic Linear Discriminant Analysis (PLDA) [9], often implemented in SR for its inter-class distribution modeling. In [9], Martinez et al. experimented with Support Vector Machines (SVMs) and Logistic Regression (LR) as back-end language classifiers. The results of their experiments led to an optimal performance using LR. LR is deployed in this work as well.

We will consider two different baseline front-ends in this work:

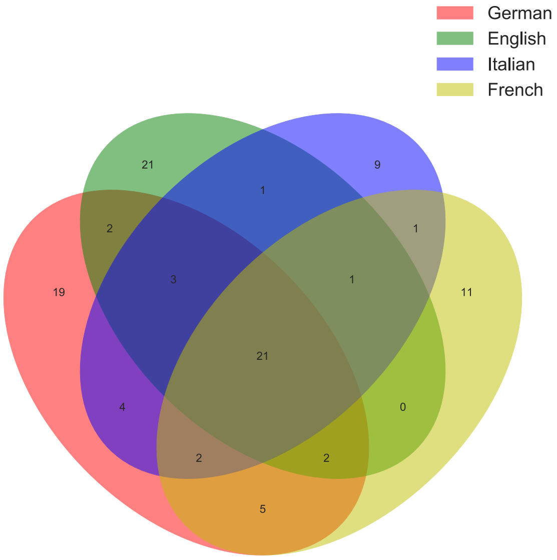

Fig. 4. Languages' manifolds in the phone space.

Language	**hours**	**utterances**
German	30.11	108'422
French	25.45	91'612
Italian	17.25	62'127
English	22.64	81'534
Total	93	343'695

Table 1. Language dataset sizes.

- **UBM/GMM-IV-LR** - UBM/GMM I-vector model developed using MFCC features, followed by the logistic regression back-end.
- **UBM/GMM$_P$-IV-LR** - hybrid UBM/GMM I-vector model considering embedding (BNF) vectors extracted using phone-based DNN, followed by the logistic regression back-end.

3 Proposed DNN embeddings

In this paper, we hypothesize that DNN embeddings can be more relevant for language discrimination if they are trained on language targets rather than trained on phone targets. This paper therefore proposes an architecture to train the LID front-end while reducing the number of target classes (i.e. equal to the number of considered languages). We denote this technique as **UBM/GMM$_L$-IV-LR**, replacing phone targets of the DNN by language targets, as shown in Figure 3. Following the second baseline principle of BNFs, our method proposes a critical reduction in the number of training targets. Keep-

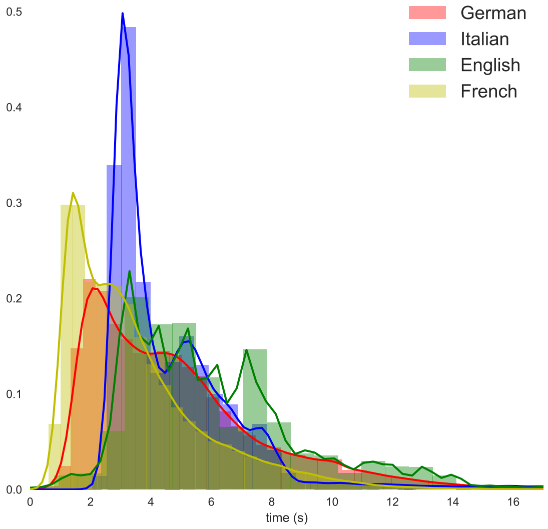

Fig. 5. Utterance duration distribution per language.

Model	C_{avg}				LER in %			
utterance-length	avg	$< 3\,s$	[3, 10] s	$> 10\,s$	avg	$< 3\,s$	[3, 10] s	$> 10\,s$
UBM/GMM-IV-LR	2.40	4.33	1.43	1.29	3.60	8.45	1.84	0.87
UBM/GMM$_P$-IV-LR	1.38	1.72	1.34	1.55	2.09	2.80	1.82	1.46
UBM/GMM$_L$-IV-LR	**0.90**	**1.30**	**0.72**	**0.71**	**1.36**	**2.47**	**0.96**	**0.72**

Table 2. LID performance for 3 different utterance-length conditions, and the average performance overall data.

ing training costs in mind, we built our method in such a way that 21 stacked MFCC frames will lead to 1 BNF (i.e. by segmenting speech and zero padding if necessary). Assuming 13 dimensional MFCCs and 80 dimensional embedding layers, the DNN front-end can be seen as a function $Y = F(X)$, where Y is an output matrix of size $(N/21, 80)$ and an input matrix X of the size $(N, 13)$. N is the number of frames of the speech segment. This frame sub-sampling leads to a faster training of the UBM. Not only should the embeddings hold more language information but the neural network training itself is significantly faster.

4 Experiments

During the development, we used two GPU GTX 1080 TI with 12 Intel cores I7 Xseries. The implementation of the models was done using Kaldi[1] toolkit. For training and test-

[1] http://kaldi-asr.org/doc/

Model	**Total in hours**	$\frac{Total}{Total_{Baseline}}$
UBM/GMM-IV-LR	63.75	100%
UBM/GMM$_P$-IV-LR	183.75	288%
UBM/GMM$_L$-IV-LR	70.35	110%

Table 3. Systems' training time comparison.

ing, we used respectively 50 K and 10 K utterances from each language which resulted in a training set of 200 K and a testing set of 40 K utterances. Input speech was characterized by 13 dimensional MFCCs with a frame rate of 10 ms, applying 25 ms hamming windows. Voice Activity Detection (VAD) was applied after MFCC feature extraction to remove non-speech frames. Both MFCC and VAD modules were borrowed from Kaldi [13]. **UBM/GMM-IV-LR**, **UBM/GMM$_P$-IV-LR** and **UBM/GMM$_L$-IV-LR** were built with a 1'024 UBM/GMM and 400 dimensional I-vectors. I-vector extractors were trained with 5 iterations using the Kaldi routine from "lre07" example. DNNs were trained with a learning rate of 10^{-4}, patches of size 64 were used. DNNs were trained with cross-entropy loss and the Adam Optimizer.

The "Speechdat" datasets [2] are telephone recordings from both fixed and mobile networks. Each dataset holds the same amount of male and female speakers. The datasets have a vast coverage of speaking styles (e.g., short commands, carefully pronounced speech, spontaneous speech). The audio files are recorded in A-law, 16 bit, 8 kHz format.

The languages used in this work are German, Swiss-French, Italian and British-English. Table 1 presents the amount of data points (utterances) we have used for each language as well as the total hours of speech. In order to build the **UBM-GMM$_P$-IV-LR** baseline, an ASR system was initially developed to obtain a phone alignment, further used to train the phone-DNN front-end. For each language dataset, minor modifications were made to the universal Speech Assessment Method Phonetic Alphabet (SAMPA[2]) dictionary. A diagram shown in Figure 4 presents the language manifolds in the phone space resulting in 102 classes. The figure reveals that phone classes are not the most language discriminative units to identify the language.

Figure 5 shows the utterance duration distribution of each language set. Class distributions are unbalanced within the 3 speech duration groups, used to evaluate LID systems (i.e. inferior to 3 s, between 3 s and 10 s, superior to 10 s). Keeping the goal to detect the spoken language in real-time, the SpeechDat appears to be an appropriate corpus, as recordings are mostly represented by short utterances.

We apply several performance metrics in this paper. First, Language Error Rate (LER) is computed for each language (i.e. specifically for all 3 speech duration groups, as well as the overall LER). Then we apply the Cost average (C_{avg}), suggested by the NIST evaluation plan [15]. Finally, we also present the training time for each model.

[2] https://www.phon.ucl.ac.uk/home/sampa/

5 Results

Overall, the proposed model **UBM/GMM$_L$-IV-LR** outperforms the **UBM/GMM-IV-LR** and **UBM/GMM$_P$-IV-LR** baselines in terms of LER and C_{avg} and **UBM/GMM$_P$-IV-LR** in terms of the computational load. Table 2 shows the detailed performance of the evaluated models in terms of C_{avg} and LER. As can be seen, significantly better scores are achieved in terms of C_{avg} and LER for all 3 utterance-length conditions.

5.1 Computational costs

Table 3 shows the computational costs required to train the LID front-ends (i.e. in hours). Obviously, **UBM/GMM-IV-LR** is the most efficient model since no bottleneck features are extracted on top of MFCCs. Nevertheless, **UBM/GMM$_L$-IV-LR** requires only 10% more time while LER is reduced by half. **UBM/GMM$_L$-IV-LR** is lighter than **UBM/GMM$_P$-IV-LR** because the DNN is trained solely on four targets rather than 102 phone targets. The 96% reduction of the number of DNN targets in **UBM/GMM$_L$-IV-LR** led to a reduction in training time by 94%, compared to **UBM/GMM$_P$-IV-LR** baseline.

6 Conclusion

This paper investigates fully adapted embeddings spaces for language identification. We hypothesised that extracting BNFs by using phone-embeddings was a sub-optimal approach. Instead, we presumed the extracted language-embeddings would be more representative for the LID. The results of our experiment validated our hypothesis, as language-DNN front-end significantly increases the LID performance as well as is less computationally expensive. In average, 35% relative reduction in both C_{avg} and LER is achieved across all 3 utterance-length conditions.

References

1. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19(4), 788–798 (2011)
2. Elenius, K., Lindberg, J.: SpeechDat Speech Databases for Creation of Voice Driven Teleservices. Phonum 4, Phonetics pp. 61–64 (1997), `http://www.speech.kth.se/prod/publications/files/538.pdf`
3. Fér, R., Matějka, P., Grézl, F., Plchot, O., Cernocký, J.H.: Multilingual bottleneck features for language recognition. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2015-January, 389–393 (2015)
4. Glembek, O., Burget, L., Matjka, P., Karafit, M., Kenny, P.: Simplification and optimization of i-vector extraction. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4516–4519 (May 2011)
5. Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., Moreno, P.J.: Automatic Language Identification using Long Short-Term Memory Recurrent Neural Networks. Proc. of Interspeech pp. 2155–2159 (2014)

6. Kenny, P.: Joint factor analysis of speaker and session variability: Theory and algorithms. Tech. rep. (2005)
7. Kramer, M.A.: Nonlinear principal component analysis using auto-associative neural networks. AIChEJ.37(2) pp. 233 – 243 (1991)
8. M. Dez, A. Varona, M. Peagarikano, L. J. Rodrguez-Fuentes, and G. Bordel: On the use of phone log-likelihood ratios as features in spoken language recognition. SLT pp. 274–279 (2012)
9. Martinez, D., Plchot, O., Burget, L., Glembek, O., Matějka, P.: Language recognition in ivectors space. In: Twelfth Annual Conference of the International Speech Communication Association (2011)
10. Matejka, P., Cumani, S., Ondel, L., Mounika, K.V., Silnova, A., Rohdin, J.: BUT-PT System Description for NIST LRE 2017 (748097) (2017)
11. Matejka, P., Zhang, L., Ng, T., Mallidi, S., Glembek, O., Ma, J., Zhang, B.: Neural Network Bottleneck Features for Language Identification. Odyssey, the speaker and language recognition workshop pp. 299–304 (June 2014)
12. Povey, D., Chu, S.M., Varadarajan, B.: Universal background model based speech recognition. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 4561–4564. IEEE (2008)
13. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society (Dec 2011), iEEE Catalog No.: CFP11SRW-USB
14. Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., Khudanpur, S.: Deep neural network-based speaker embeddings for end-to-end speaker verification. In: 2016 IEEE Spoken Language Technology Workshop (SLT). pp. 165–170 (Dec 2016)
15. US department of commerce, N.: The 2007 NIST Language Recognition Evaluation Plan (LRE07). NIST Web document pp. 1–5 (2007), https://catalog.ldc.upenn.edu/docs/LDC2009S04/LRE07EvalPlan-v8b-1.pdf