Odyssey 2010
The Speaker and Language Recognition Workshop
28 June – 1 July 2010, Brno, Czech Republic

Coping with Two Different Transmission Channels in Language Recognition

Florian Verdet[1,2], *Driss Matrouf*[1]
Jean-François Bonastre[1], *Jean Hennebert*[2]

[1]Université d'Avignon et des Pays de Vaucluse, Laboratoire Informatique d'Avignon, France
[2]Département d'Informatique, Université de Fribourg, Fribourg, Switzerland
florian.verdet@univ-avignon.fr, driss.matrouf@univ-avignon.fr,
jean-francois.bonastre@univ-avignon.fr, jean.hennebert@unifr.ch

Abstract

This paper confirms the huge benefits of Factor Analysis over Maximum A-Posteriori adaptation for language recognition (up to 87% relative gain). We investigate ways to cope with the particularity of NIST's LRE 2009, containing Conversational Telephone Speech (CTS) and phone bandwidth segments of radio broadcasts (Voice Of America, VOA). We analyze GMM systems using all data pooled together, eigensession matrices estimated on a per condition basis and systems using a concatenation of these matrices. Results are presented on all LRE 2009 test segments, as well as only on the CTS or only on the VOA test utterances. Since performances on all 23 languages are not trivial to compare, due to lacking language–channel combinations in the training and also in the testing data, all systems are also evaluated in the context of the subset of 8 common languages. Addressing the question if a fusion of two channel specific systems may be more beneficial than putting all data together, we study an oracle based system selector. On the 8 language subset, a pure CTS system performs at a minimal average cost of 2.7% and pure VOA at 1.9% $minC_{avg}$ on their respective test conditions. The fusion of these two systems runs at 2.0% $minC_{avg}$. As main observation, we see that the way we estimate the session compensation matrix has not a big influence, as long as the language–channel combinations cover those used for training the language models. Far more crucial is the kind of data used for model estimation.

1. Introduction

The focus of this paper is language recognition, which consists in processing a speech signal to detect which language the speaker is talking in.

It is obvious that the data we observe includes not only useful information, but also information that does not help in the task of language recognition. The unwanted information covers speaker specificities including vocal tract configuration, current emotion or health status. It covers as well recording conditions with background noise, microphone setup, transmission channel and speech signal encoding. We propose here to qualify this perturbing information as *session* dependent. The data we observe is then composed of useful information, the information that depends on the language, and useless or even perturbing information: the information that depends on the session.

The feature extraction and modeling strategy (e.g. with GMMs) should attempt to focus on the useful information and to discard the language independent perturbing information. However, usual feature extraction approaches can only partially discard perturbing information related to the recording setup and transmission channel. Furthermore, a lot of the speaker dependent specificities are kept in the features. The systems proposed in this work keep track of this session variability. This helps distinguishing the language dependent information.

For this, we need several sessions for every language (the more the better). Typically, each utterance recording can be seen as a different session. In order to detach language dependent information from session variability, we consider the language part being the information that is common to all sessions (utterances) of a given language and the residuals being the session variability.

In testing stage, where we are left alone with only one session, the perturbing variability contained in the data is estimated and removed based on the variability seen in the training data. What remains should hopefully emphasize the useful part of the information and thus, the classification should be more precise.

These principles can actually be implemented using the *Factor Analysis* (FA) approach. FA has triggered significant advances in speaker verification as described in [1, 2]. The context of language recognition is sensibly different from speaker recognition. Each class has far more training data than is usually the case for speaker verification. Bigger models can thus be estimated more robustly as some analysis in [3] show on different model sizes. We have less classes (only some languages instead of a lot of speakers), but there are a lot of different sessions for each language. This big speaker variability may be caught at the same time as the finer (e.g. session) variabilities.

In 2009, it was the first time that NIST's Language Recognition Evaluation (LRE) [4] included data of two quite different conditions. Namely the classical Conversational Telephone Speech (*CTS*) and the major part coming from telephone bandwidth parts of Voice Of America (*VOA*) radio broadcasts. This LRE disposes of a total of 23 target languages.

This paper will analyze different ways to cope with the challenge induced by these two channel conditions. As pointed out in recent works [3], FA systems presented here will show a very important gain over traditional MAP adaptation systems. These FA systems require a projection matrix that keeps track of the session and channel variabilities observed in the training data. Since the present evaluation setup contains two rather distinct channel conditions (CTS and VOA), this paper will study, amongst others, the possibility to estimate one compensation matrix for each of these.

In regard to system setup, one possibility is to build one unique system including data from both conditions pooled together. But initially, it may be a better idea to cope with these two conditions separately up to a certain point and then putting them together. There are several possible setups: we may es-

timate one FA session variability matrix using both, CTS and VOA data and then train condition dependent models using this common compensation matrix. Another strategy could be to estimate session variability matrices separately and train the language models using both conditions (this case is not handled here because it does not seem optimal to use some training material of a condition which the compensation was not designed for). Finally, we may estimate separate matrices and estimate the models using only data of the same condition as used for the matrix.

We will thus have a look at systems with a global approach putting all data together and systems handling these conditions separately. These systems will be evaluated following the NIST LRE 2009 protocol with its 23 languages, as well as on a common subset of 8 languages since we do not have training and testing data for both channels in all 23 languages (more on this in Section 4.2). Each system will also be evaluated on the CTS segments of these 23 and 8 language evaluation sets only and on the VOA utterances only. We will further present results for a fusion of channel-dependent systems. This fusion is done by selecting the scores of one or the other system based on an oracle revealing the true channel of each test utterance.

All reported experiments are conducted using the free software framework MISTRAL [5], which uses the ALIZE library [6]. The evaluation protocols are the ones of NIST Language Recognition Evaluation 2009. System performance is measured, as in LRE 2009 [4], in $\%\ minC_{avg}$ and may be depicted in the form of mean DET curves.

Section 2 gives a description of the general working of our FA systems along with the way they are scored and evaluated. In Section 3, we present all the data that was used for training and for testing. The different systems analyzed in this work are stated in Section 4, together with some explanations on the reason of testing on an 8 language subset. The elucidations on the experimental setup in Section 5 are succeeded by the results of the various single systems and their fusion in Section 6 and give place to conclusions and some lookout in Section 7.

2. Basic Factor Analysis system

2.1. GMM system using a UBM

To keep our models general enough to cover also feature vectors which have not been seen during training, the models are based on a *Universal Background Model* (UBM). The parameters of the UBM are estimated taking as much and as different data as possible from a large set of languages. For a baseline system, the model for each language is then obtained through a *Maximum A-Posteriori* (MAP) *adaptation* of this very general UBM towards the training data [7]. For the means (μ) of the Gaussians, the MAP adaptation can be expressed as a weighted sum of the UBM means and the means of one language's training data (factor α being the importance of the training data means):

$$\mu_{language} = (1-\alpha) * \mu_{ubm} + \alpha * \mu_{data} \qquad (1)$$

This keeps both, a certain generality coming from the UBM side and a good fitting to the training data of the language.

2.2. GMM-UBM with Factor Analysis

For simplicity and because it has proven to work well, we operate the factor analysis solely on the means of the Gaussian mixtures and do not touch their variances [2]. If we concatenate all the means of one model, we obtain one big *mean super-vector* (SV). The basic factor analysis (FA) formula can be stated as:

$$m_{observed} = m_{ubm} + Dy_{language} + Ux_{session} \qquad (2)$$

where m are mean supervectors, y is the part which is specific to the language, weighted by D (a diagonal matrix), and Ux is the session variability, which is included in the observed data but which we do not want to be included in the language model. The Factor Analysis model assumes that the session variability is located in a low-dimensional subspace. This subspace is generated by the vector columns of the U matrix. x are the session factors in this subspace. The data we observe is thus modeled as being of a global base (m_{ubm}) with a language specificity (Dy) and some session variability (Ux).

2.2.1. Estimating eigensession compensation matrix

The matrix U, here called *eigensession matrix*, is common to all languages. It is iteratively estimated using expectation maximization (EM) algorithm with maximum likelihood (ML) optimization criterion. Each step, the different $x_{session}$ (variability) vectors are estimated, then a $y_{language}$ is estimated for each language (using the new x) and finally U is estimated globally, based on these x and y. Since x and y also depend on U, the process is iterated. The step by step algorithm is described in [2].

2.2.2. Training language models

The model for each language, which is stored at training stage, is the $m + Dy$ part of factor analysis formula (2). Very similar to MAP adaptation, it is a weighted combination of the UBM mean m and the mean supervector Dy estimated using all training data corresponding to the target language, D being the weighting factor. It can be seen as MAP adaptation operating on session-compensated (Ux part removed) information.

When estimating the model for a given language, the per session x are estimated. Using the stored U matrix, the Ux part is subtracted from the data's mean supervector and y gets estimated. Finally, the remaining $m + Dy$ part is stored as the language's model means (see [2] for details). To obtain full language model parameters, mixture weights and covariances are taken unchanged from UBM.

2.2.3. Testing using compensated models

We assume that the observed feature vectors contain session perturbation (speaker and channel). In testing stage, different strategies may be applied to cope with this:

Feature normalization An often used strategy [8] consists in normalizing each feature vector by removing the session effect, which is the Ux component weighted by the feature vector's posterior probability against hypothesized language's model. x is estimated using statistics of the testing utterance as detailed in [2]. Once the feature vectors are normalized, they are tested against the model of the hypothesized language. We may sketch this by:

$$feat - Ux_{utterance} \quad | \quad m + Dy_{language}$$

So, we try to remove the impurity (normalizing by Ux component) to match them up with the clean models issued from training stage (which do not include session variability Ux). Most of the FA using systems of LRE 2009 participating sites apply session compensation on the feature level [13].

Model compensation Instead of removing session variability from the features, we may likewise compensate this estimated variability on the model itself and leave the features untouched:

$$feat \quad | \quad m + Dy_{language} + Ux_{utterance}$$

We simply add the Ux component to the model (we're still in mean supervector context). So that un-normalized test data is tested against a modified model. This is not the same as saving the full model (including Ux) at training stage, since here, x is estimated on the current test utterance only. This strategy will be used for the systems presented here.

2.3. Scoring

Scores are normalized separately for each test utterance among all languages. This is done by dividing each score (usually the likelihood of the test utterance being of a given language) by the sum over the scores obtained against all language models.

System performance can be enhanced by powering each score with a constant K. Matějka et al. describe in [9] that this procedure attempts to introduce some correction to the assumption of the frames being independent to each other.

$$\widehat{score}_l(utterance) = \frac{score_l(utterance)^K}{\sum_{i \in L} score_i(utterance)^K}$$

l being the hypothesized language and i being each language in turn. In our case, a K of 35 has been chosen. This is based on consequent observations of the impact of different Ks (in steps of 5) prior to our participation to the Language Recognition Evaluation 2009.

2.4. Evaluation

System performance is measured using *minimal average cost* ($minC_{avg}$). It is the detection system choosing the decision threshold in such a way that the average expected cost of misses (utterances not recognized as being of the true language) and false acceptances (mistakenly detecting the presence of a language) among all target/non-target language pairs is minimal (see Section 4.1f of the LRE 2009 plan [4] for a description).

In our case, a false negative (a miss) and a false positive (false acceptance) decision have the same cost and the prior of a target trial is 0.5. The cost function that will be minimized is thus:

$$C_{avg} = \frac{1}{N_L} \sum_{l \in L} \Big[0.5 \cdot P_{Miss}(l) + \frac{0.5}{N_L - 1} \sum_{k \neq l \in L} P_{FA}(l,k) \Big] \tag{3}$$

where N_L is the number of languages in our (closed-) set, P_{Miss} is the probability that a language model misses a match (false negative) and $P_{FA}(l,k)$ is the probability that an utterance of language l is mistakenly recognized as being of language k. These probabilities are calculated in function of a global threshold. It is thus the mean over all target languages of its probability to be missed and its average probability to be detected by a false language model.

For evaluations using test utterances of one channel condition only and not having all languages present in the test data set (in the context of 23 languages), this cost function turns to

$$C_{avg} = \frac{1}{N_{L_T}} \sum_{l \in L_T} \Big[0.5 \cdot P_{Miss}(l) + \frac{0.5}{N_{L_M}\text{-}1} \sum_{k \neq l \in L_M} P_{FA}(l,k) \Big] \tag{4}$$

where L_T is the set of languages in the test data set (also called *target languages*) and L_M is the set of languages for which we have models (*non-target languages*). For the 23 language evaluations on a per-channel basis, these two sets are not identic. Since the test set is not complete, we have less languages in the test set than we have models. In a general manner, $N_{L_T} \leq N_{L_M}$.

3. Data parts

3.1. Training data

Training material is drawn from various sources. Let us define the different data sets as follows:

3.1.1. CTSsmall

The following corpora are used, providing data for the *CTS* condition:

- All three parts (*train*, *devtest* and *evltest*) of the CallFriend[1] [10] corpus. Each of these three parts of the corpus contains 20 complete two-ended, half-hour conversations per language. The CallFriend corpora of 8 languages[2] are used, including available dialects.
- The 120 Indian English recordings of NIST's LRE 2005 development data[3].
- The full conversations of the LRE 2007 evaluation data[4] for 9 languages[5].

Each language has between 40 and 317 segments representing between 2.7 and 58.6 hours of speech. In total for 11 languages, we have 312 hours in 1867 segments.

3.1.2. CTS

This set includes *CTSsmall*, augmented by the 10 and 30 second evaluation segments (ranging from 284 to 1934 segments per language) of LRE 2005[6] for 6 languages[7].

For this set, the 3 second Indian English segments of LRE 2005 development have been avoided, since we added other utterances of this language. Each language has between 40 and 2253 segments representing between 2.7 and 64.8 hours of speech. In total for 11 languages, we have 337 hours in 7870 segments.

3.1.3. VOA

This data comes from the Voice Of America 3 (VOA3) data set[8] and is used together with the *phone*/*wideband* and *speech*/*non-speech* segmentation provided by NIST for the LRE 2009 campaign and which were built by the Brno University of Technology (BUT) lab [11]. Each language is represented by 347 to 400 artificial speakers (described in Section 5.1), summing up to 3.0 to 27.9 hours of speech. In total for 22 languages, we have 333 hours across 8632 segments.

[1] LDC1996S*.

[2] English, Farsi (Persian), French, Hindi, Korean, Mandarin, Spanish and Vietnamese. Both dialects for English, Mandarin and Spanish (having thus 40 conversations).

[3] lid05d1 aka. NIST-R103 aka. LDC2006E104 aka. LDC2009R31, part of LDC2008S05 and LDC2009E41.

[4] LDC2009R31.

[5] Cantonese, English, Indian English, Korean, Mandarin, Persian (Farsi), Russian, Spanish and Vietnamese.

[6] lid05e1 aka. NIST-R104-1.1 aka. LDC2006E105, part of LDC2008S05 and LDC2009E41.

[7] English, Hindi, Indian English, Korean, Mandarin and Spanish.

[8] LDC2009E40 (which includes also the VOA2 set).

3.1.4. VOA10k

This set contains the data of the *VOA* set and a lot of additional VOA3 data for a total of 141 599 segments (1111 to a maximum of 11 029 per language). This data set is only used for training the Universal Background Model.

3.2. Testing data

Tests are conducted on NIST-LRE 2009 data [4]. This evaluation set is composed of 41 794 utterances containing nominally 3, 10 and 30 seconds of speech each. Our processing did not detect any speech in 106 of these files. The other sum up to 133.3 hours of speech.

The primary condition aggregates just utterances of the 23 languages (closed-set condition) with a total of 31 178 utterances. We focus mainly on the 30 second ones, which comes down to 10 571 files giving that many target trials and thus 232 562 non-target trials. There are between 315 and 1015 testing files per language.

This set of utterances comprises data drawn from CTS and from VOA sources [4, 12]. There are 8708 testing files in 10 languages[9] originating from CTS sources. Thereof 3081 for the 30 second condition with 32 to 625 for each language. Drawn from VOA are 22 470 testing files with 7490 of 30 seconds. We count between 27 and 399 testing utterances for each of 22 of the 23 languages.

4. System descriptions

4.1. Universal Background Model

The Universal Background Model of a preceding experiment (using all CallFriend data of the 7 NIST LRE 2005 languages) has served as starting point for initializing the new one. It has then been trained iteratively by an EM/ML algorithm. During the 6 iterations, the amount of training data has been increased from 65 mio up to 787 million speech frames, which represents 2185 hours and 143 466 speakers. For this, the training data parts *VOA10k* and *CTSsmall*, described in Section 3.1, have been used, thus including six times more VOA3 data than standard phone data.

The same UBM has served for all experiments. This is required if we want to concatenate eigensession matrices (for the systems described in Section 4.6).

4.2. Setup particularity

We have CTS training data available for only 11 of the languages[10] and VOA data for 22 of the 23 languages[11] included in NIST LRE 2009. The fact that we do not have training data of both conditions for every language poses some troubles. To be still able to obtain a full set of 23 language models, we will use data of the other condition where needed (Section 6.2.1 will introduce the augmented data sets).

Similarly, on the testing segments side, some condition–language combinations are missing. This presents a more consequent problem. CTS test utterances are available for 10 languages[12] only (of which 9 correspond to the available CTS training data languages[13]). VOA test utterances are present for the same 22 languages as the training data. So we end up comparing system performances on a different number of languages. But this is not a big problem since the evaluation is a *detection* task, which answers binary questions (in contrast to an identification task that has to select one language out of a set) and under which the average expected cost of a system delivering random decisions is 50%, independent of the number of classes.

There are only 8 languages that we find everywhere, in CTS and VOA, in training and in testing data. So results will also be presented for this subset of 8 common languages.

4.3. System using pooled data

For this system, all training data, thus the *CTS* and the *VOA* data sets defined in Section 3.1, is used for estimating the session compensation matrix as well as the language models.

4.4. pure CTS system

The CTS system uses only standard phone data (*CTS* data set) for estimating the session compensation matrix. The language models are then estimated using this pure CTS session compensation matrix and the *CTS* training data set.

4.5. pure VOA system

The pure VOA system estimates a session compensation matrix on VOA data only (*VOA* data set), which are phone calls transmitted over broadcast. This matrix serves then for estimating language models using this same *VOA* data set.

4.6. System with merged session compensation matrices

For this system, the session compensation matrix is built concatenating the two condition dependent matrices of the CTS and the VOA systems described above to form a matrix of double rank (80 instead of 40). The difference to an eigensession matrix of the same rank 80 is that, in the present case, there are 40 channels assured for each condition, whilst for the pooled case, the channels are allotted dynamically - some channels may even be similarly present in both conditions. Two sites participating in NIST LRE 2009 have used a similar strategy of stacking session compensation matrices[14].

Using this merged (thus dual-condition) matrix, we may train the language models using either training data pooled together or we may build condition specific models using either data set *CTS* or *VOA*.

5. Experimental setup

The framework used for all experiments is principally the free software MISTRAL [5, 6]. For cepstral feature extraction, SPro4 [14] has been used.

In a first step the systems are evaluated on the 8-language subset sketched in Section 4.2 and which includes Cantonese, English, Hindi, Korean, Mandarin, Persian, Russian and Vietnamese.

The experiments are also run in the 23-language context of NIST's Language Recognition Evaluation (LRE)

[9] Cantonese, English, Hindi, Indian English, Korean, Mandarin, Persian, Russian, Urdu and Vietnamese

[10] Spanish, English, Korean, Mandarin, Hindi, Indian English, Cantonese, French, Persian (Farsi), Russian and Vietnamese.

[11] The missing language is Indian English.

[12] Cantonese, English, Hindi, Indian English, Korean, Mandarin, Persian, Russian, Urdu and Vietnamese.

[13] The above except Urdu.

[14] ATVS, Universidad Autonoma de Madrid, Spain and IFLY, iFlyTekSpeech Lab, EEIS University of Science and Technology of China systems described in [13].

2009 [4], which comprises Amharic, Bosnian, Cantonese, Creole (Haitian), Croatian, Dari, English (American), French, Georgian, Hausa, Hindi, Indian English, Korean, Mandarin, Pashto, Farsi (Persian), Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu and Vietnamese.

5.1. Parametric setup

In our experiments, we use *Shifted Delta Cepstra* (SDC) parameters in the configuration 7-1-3-7 (in concordance with other researches in this domain [15, 16, 8, 9]). This means we're having 6 cepstral MEL-scale coefficients plus energy (cepstra and energy are kept in the parametric vector) and seven delta blocks stacked, each block calculated on frames $t-1$ and $t+1$ with a t shifted by 3 each time. This yields feature vectors of size 56.

Energy based *speech detection* is conducted on all utterances to spot speech and non-speech parts. All features are then *normalized* in such a way that the features containing the speech part of one utterance have an average of 0 and a variance of 1.

For VOA3 training data, the following particular step has been inserted. Since Factor Analysis' session compensation requires the concept of individual "*speaker*" utterances (what would be recording sessions in a speaker verification context), we emulated the speakers as follows. For this VOA3 data, NIST provided two kinds of labels [11]: *speech*/*non-speech* and *phone*/*wideband*. We assume all *phone* labeled segments within the same longer *speech* segment belonging to the same real speaker (thinking of a possible dialog between a moderator (wideband) and a phone-caller). We thus concatenate the *phone&speech* segments to longer "speaker" segments, which are subsequently used as individual training files. The speech activity detection and $0|1$ normalization steps are applied prior to this concatenation. We have recently seen that other researchers do a similar concatenation step [11].

The evaluation data has been processed in a similar manner (bypassing the concatenation of artificial speakers). Our speech activity detection did not detect any speech in 106 of the evaluation files. For these files all scores are set to a constant value, so the system makes 50% errors on these trials (due to the flat prior and equal costs, as stated in Sect. 2.4).

6. Results

The results presented here feature full systems using mixtures of 2048 Gaussians. All results are for 30 second segments, NIST LRE 2009's closed-set primary condition. Results are first presented on the 8-language subset and then on all 23 languages in order to match the NIST LRE 2009 protocol.

We define tests of *matching condition* as CTS test segments tested on the *CTS* system or VOA segments on the *VOA* system. Similarly, *cross condition* tests are tests where the segment condition does not match the channel condition on which the system was trained.

6.1. Evaluation on 8 languages

We observe that both data parts (*CTS* and *VOA*) have about the same total amount of training data, but for *CTS* distributed on only half the number of languages, benefiting thus in average from twice as much data each (30.6 vs. 15.1 hours). So VOA seem to be the easier tests, since trained on less data in average, they give a far better performance.

6.1.1. MAP adapted GMM-UBM system

We will use these GMM-UBM systems as baseline to compare the FA systems to. The GMM-UBM language models are obtained by simple MAP adaptation with a factor α of 14.0, where only the mean values are changed (neither Gaussians' weights nor variances are adapted).

While seeing the GMM-UBM system as baseline, it obtains, evaluated on 8 languages, 18.21% $minC_{avg}$ when trained on *CTS* data and 18.31% with *VOA* data. Using all data for training, its mean average cost is at 16.91%. Table 1 highlights the number of tests totally and on a per condition basis as well as the results of the MAP adapted UBM GMMs.

Table 1: *MAP adapted GMM-UBM systems evaluated using only the* 8 *mutual languages on all test segments and on a per-condition basis, in* $\%$ $minC_{avg}$

system	data for model estimation	8 languages closed-set 30s tests all	CTSonly	VOAonly
nb of testing languages		8	8	8
total number of test files		4635	2475	2160
per language		315–1015	52–625	27–397
MAP	CTS&VOA	16.91	19.69	14.27
MAP	CTS	18.21	19.64	19.27
MAP	VOA	18.31	24.62	12.27

6.1.2. UBM-based factor analysis systems

For the different Factor Analysis systems we analyze here, the eigensession matrices are obtained using the unique UBM and are set to have a *rank* of 40 (number of session factors in x). They are iteratively estimated during 14 iterations. In the case of concatenated compensation matrices, we have a rank of 80.

For training each language model, statistics over training data against the UBM are collected and x and y are estimated. The model, being the $m + Dy$ part of the Factor Analysis formula (2), gets then fashioned in one step.

Evaluated on the 8 language subset, the base factor analysis system pooling together the training data of both conditions performs at 2.23% $minC_{avg}$. Evaluating only on the CTS test utterances yields 3.03% $minC_{avg}$ and on the VOA files 1.75% $minC_{avg}$.

Table 2: *Different UBM-based FA systems evaluated on all test segments and on a per-condition basis, in* $\%$ $minC_{avg}$

data for estimating U matrix	models	8 languages closed-set 30s tests all	CTSonly	VOAonly
pooled	CTS&VOA	2.23	3.03	1.75
pure CTS	CTS	3.06	2.71	4.28
pure VOA	VOA	6.78	10.15	1.90
merged U	CTS&VOA	2.33	3.06	1.69
merged U	CTS	3.32	2.61	4.76
merged U	VOA	6.78	10.34	1.92

Table 2 shows the results on all FA systems. The number of tests is indeed the same as indicated in Table 1. As expected, channel condition dependent systems are better on matching tests than on cross condition tests. The pure *VOA* system presents a striking difference between matching tests with 1.90 and cross condition tests with a cost as high as 10.15% $minC_{avg}$!

Let us have a look at following (pessimistic approach-) sequence for the CTS tests only: Cross-condition MAP is at 24.62% $minC_{avg}$, pure cross-condition FA drops to 10.15% (-59% $relative$), adding some matching channels (merged eigensession matrix) stays similarly at 10.34% $minC_{avg}$. Adding some channel-matching data for language model training improves the performance to 3.06 (another -70% $relative$). So we see that FA is very useful, even for channels purely estimated on sessions of a quite different condition. These results tend also to show that the session variability captured by the eigensession matrix is of global nature, somehow independent of the channel condition. We further see the importance to have data as similar to the testing data as possible – this seems to be even of more benefit for FA (the above -70% $relative$) than for MAP (-20% $relative$). Analogously for VOA tests only, we observe: Cross-condition MAP 19.27% $minC_{avg}$, pure cross 4.28% (-78% $relative$), adding matching channels gives 4.76% and adding matching training data yields 1.69% $minC_{avg}$ (another -64% $relative$).

For matching condition, CTS test utterances, we have a cost reduction of 86% $relative$ between simple MAP and the pure system. For matching VOA, the cost reduction is of about 84% $relative$.

Globally, we see that performance does not vary much by changing the way the session compensation matrix is estimated. Thus the most important, besides having some compensation matrix, is to have at one's disposal a lot of training data, with a part of it as similar to the testing data as possible.

6.1.3. Oracle based fusion

We observe that performance is very good when eigensession matrix, training data and test part are of matching conditions. This leads to the question if we could not fuse two condition dependent systems by selecting the score that matches the channel to obtain the benefit of both systems. Thus, the fusion of two systems is done employing a channel-based system selector.

This section presents and discusses the results using an *oracle* type selector. The oracle tells us of which condition the test utterance really is. Our fuser then selects the scores of the corresponding channel-dependent system for that utterance. Since this selector is perfect in terms of channel detection, we can interpret the result as being the best performance a fuser based on automatic channel detection can approach.

We note that the result on all 30s tests (grouping CTS and VOA segments) can differ from the calculated linear combination of the corresponding channel performances of the two fused single systems (their sum weighted by the number of tests). This happens since the indicated performance uses the average cost function (C_{avg}) and is thus a mean of language pair (mis-) detection costs.

Table 3: *Fusion of FA systems using an oracle type system selector, in* $\%$ $minC_{avg}$

Fusion of system 1 & system 2	8 languages closed-set 30s tests		
	all	CTS only	VOA only
CTS,CTS&VOA,VOA merged U, CTS&VOA	2.06	2.71	1.90
	2.04	2.61	1.92

Evaluating the fusion (Table 3) of the pure CTS (2.71% $minC_{avg}$ on the matching tests) and the pure VOA (1.90% $minC_{avg}$ on VOA) systems yields a global 2.06% $minC_{avg}$. This result is interesting since it is better than the global results of the system merging the session compensation matrices (-12% $relative$) or the system pooling all data together since beginning (-7.7% $relative$).

The same kind of fusion applied on the merged eigensession matrix-based condition-dependent systems performs at 2.04% $minC_{avg}$, which is even 8.6% $relative$ better than the best single system.

Figures 1 depicts the per-condition trained (and tested) merged-matrix systems along with their fusion.

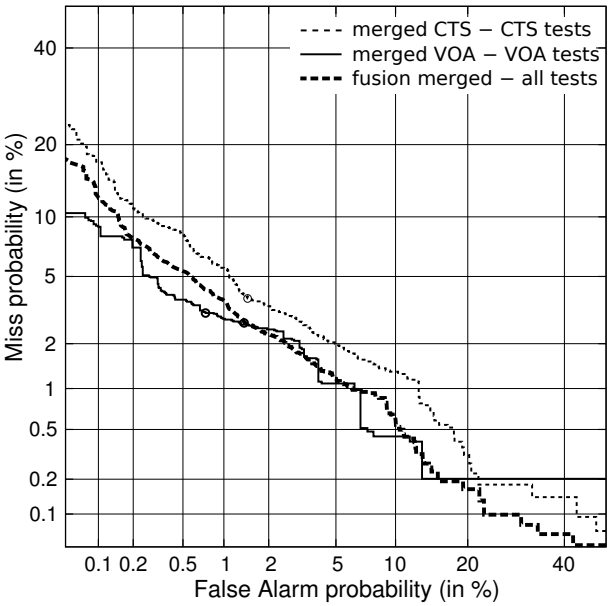

Figure 1: *DET curves for* 8*-language systems based on concatenated eigensession matrix, per-condition trained and tested on the matching tests only and the oracle type fusion of these systems tested on all 30 second segments.*

6.2. Evaluation on 23 languages

This section presents the same systems under NIST LRE 2009 condition, where there are language–condition combinations missing as well in the training as in the testing set. It also shows to which extent the systems are robust enough to recognize languages on a channel condition for which no training data is available.

As described in Section 4.2, NIST's LRE 2009 evaluation concerns 23 languages. Hence, we need some data to train each of the 23 language models. For this, we extend the data sets used for training as follows:

6.2.1. Extended data sets

CTS+ The *CTS+* data set comprises the *CTS* data set for its 11 languages, extended by the *VOA* parts for the other 12 languages.

VOA+ This set analogously contains the *VOA* data set for the 22 languages contained in it plus the *CTS* part for Indian English, which is missing VOA data.

6.2.2. MAP adapted GMM-UBM system

In the 23 language context and still with 2048 Gaussians, they evaluate to 20.60% $minC_{avg}$ when using all data to train the models. Taking the *CTS+* data set for training, it obtains 23.54% $minC_{avg}$ and when trained on *VOA+* 21.62%. The best matching condition MAP system runs at 17.30% $minC_{avg}$. These results are presented in Table 4 – indicating again the number of tests for this 23 language context.

Table 4: *MAP adapted systems using all* 23 *languages, on all test segments and on a per-condition basis, in* $\%$ $minC_{avg}$

system	data for model estimation	LRE 2009 closed-set 30s tests all	CTSonly	VOAonly
nb of testing languages		23	10	22
number of test files		10571	3081	7490
per language		315–1015	32–625	27–399
MAP	CTS&VOA	20.60	22.41	21.21
MAP	CTS+	23.54	21.50	27.89
MAP	VOA+	21.62	35.48	17.30

6.2.3. UBM-based factor analysis systems

The base factor analysis system pooling together the training data of both conditions performs at 4.79% $minC_{avg}$. Evaluating only on the CTS test utterances yields 6.71% $minC_{avg}$ and on VOA files 4.57% $minC_{avg}$. Table 5 also shows the results on the other systems.

Table 5: *All UBM-based FA systems, in* $\%$ $minC_{avg}$

data for estimating U matrix	models	LRE 2009 closed-set 30s tests all	CTSonly	VOAonly
pooled	CTS&VOA	4.79	6.71	4.57
pure CTS	CTS+	9.22	7.75	10.06
pure VOA	VOA+	7.95	21.51	3.91
merged U	CTS&VOA	4.47	7.16	4.14
merged U	CTS+	6.23	6.96	6.86
merged U	VOA+	6.16	13.13	3.72

Further on, systems using a merged eigensession matrix (and still trained on one or the other condition only) benefit from the extended channel modeling (80 instead of 40 and more diverse channels): an improvement of -32 and -39% $relative$ $minC_{avg}$ for cross-condition tests and also slightly (-5 to -10% $relative$) for tests with matching condition. This observation cannot be confirmed in the "cleaner" 8 language setup. In that setup, we have all required language–channel combinations in our training data and thus the additional channels are not of much use. An interpretation of this difference between the 8- and the 23-language setups is that the data used for estimating the U matrix has at least to cover the language–channel combinations used for model training. Maybe the session variability mapping in the compensation matrix are fairly on a per-language basis – perhaps even catching some language-dependent part. So adding additional (foreign–condition) channels does not help in a context where training data covers the set of testing languages. But can be useful when correct channels are missing from the training data for some languages.

The drastic degradation compared to Table 2, occurring on the *CTS* systems (doubling or tripling the number of errors) can be explained by the fact that in this 23 language case, not even half of the models could really be trained with *CTS* data (the *CTS+* data set containing *VOA* data for 12 of the languages).

Looking at the performances on all 30 second tests, no single channel dependent system performs as well as systems with models trained on all data together.

6.2.4. Oracle based fusion

The oracle fusion is also applied on the whole set of 23 languages to match LRE 2009 protocol. The results using an oracle as selector for fusion are shown in Table 6.

Table 6: *Fusion of UBM-based FA systems using an oracle type selector, in* $\%$ $minC_{avg}$

Fusion of system 1 & system 2	LRE 2009 closed-set 30s tests all	CTS only	VOA only
CTS,CTS+&VOA,VOA+	4.16	7.75	3.91
merged U, CTS+&VOA+	3.91	6.96	3.72

The fusion of the two pure systems with 4.16% $minC_{avg}$ is slightly outperformed by the fusion of the systems built on a common, merged eigensession matrix with 3.91% $minC_{avg}$ (-6% $relative$). This is similar to the differences that has been observed between unfused pure and unfused common-matrix systems in Table 5. A big part of the easier recognition on VOA tests is still due to the fact that nearly all language–channel combinations are present in the training data. This fused system with a minimal average cost of 3.91% has a cost reduction of 12.6% $relative$ compared to the best single system with its 4.47% $minC_{avg}$.

Table 7: *Oracle type fusion of merged-*U *systems, confusions of language pairs of interest, in* $\%P_{FA}$.

language pairs		detection confusions A as B	B as A
language A	language B	P_{FA}(A,B)	P_{FA}(B,A)
Cantonese	Mandarin	19.58	6.40
Portuguese	Spanish	6.05	7.53
Creole	French	78.73	60.37
Russian	Ukrainian	11.74	71.39
Hindi	Urdu	73.01	94.20
Persian	Dari	63.85	70.69
Bosnian	Croatian	68.45	75.53
English	Indian English	38.39	54.01

Table 7 shows the error rates on the set of language pairs of particular interest (as announced in the LRE Plan [4]) for the merged-U oracle fusion. They are from the 23 language setup since none of the pairs is present in the 8 language subset. The presented values are rates of false positives, P_{FA} of Eq. (3) and may thus exceed 50%. They are calculated using the global threshold that gives 3.91% $minC_{avg}$ for the whole system. All big confusions ($> 33\%P_{FA}$) fall into the presented language pairs. 40.9% of all false positives are made on these language pairs – the remaining have an average P_{FA} of 2.36%.

7. Conclusions and perspectives

In this paper, we investigated ways to cope with the particularity of NIST's LRE 2009 data containing conversational telephone speech (CTS) as well as phone bandwidth parts of radio broadcasts (VOA). All systems were evaluated under full LRE 2009

condition as well as only on the CTS or only on the VOA segments. To avoid bias introduced by testing data without matching training data or vice versa, we evaluated the systems also on a common subset of 8 languages.

We confirmed the benefits of the factor analysis method for language recognition. We compared the results in a first time to the traditional GMM-UBM approach using MAP adaptation. Using FA over MAP generally reduces the expected system costs by at least $60\%\ relative$ and up to $87\%\ relative$.

For FA systems, we see that performance does not vary much by changing the way the session compensation matrix is estimated – on only one condition, pooling data of both conditions together or concatenating the single-condition matrices, the latter generally being slightly better.

Results obviously show that systems trained on data of one condition only have the best performance on tests of the matching condition: $2.71\%\ minC_{avg}$ for pure CTS and 1.90% for pure VOA systems on 8 languages.

We also show the possible enhancement by fusing two channel dependent systems over systems which pool both data types together. This fusion is analyzed by selecting, for each test utterance, the scores of one or the other system – in our case following an oracle indicating the real channel of an utterance. Evaluated on 8 languages, fusing systems that perform globally at 3.32 and $6.78\%\ minC_{avg}$ puts us to $2.04\%\ minC_{avg}$.

Most observations in the data-complete 8 language context also hold for the lacunary 23 language context – especially the benefit of fusion.

During preparation, we observed that we could re-apply a score normalization step described in Section 2.3 on the fused scores and boost the error rates from 2.04% down to $1.56\%\ minC_{avg}$ ($-24\%\ relative$). This seems to occur due to a non-optimal exponent K. A value of 35 has proven well-suited for experiments on data from CallFriend or previous NIST LREs but seems not the best for this kind of evaluation including data of a quite different nature (VOA). We will have to investigate ways to optimize this K.

The possible gain of the fusion presented herein probed by an oracle system selector has to be assessed using a selector that automatically detects the channel. First attempts indicate a performance of about $2.15\%\ minC_{avg}$ for fusing the pure systems in the 8 language context. This is still $3.5\%\ relative$ better than the best single system and suggests that this type of fusion can be useful.

The analysis presented here can be spun further to find ways to build systems on several automatically detected (think of "clustering") macro-channels. This would allow us to loosen the limitation to the two conditions we had in the present work. Since the fusion enhances the performance (about $-8\%\ relative$ for an oracle based selector), we can hope that we may improve even more by slicing the data into multiple channel classes, each one modeled by an adapted specific system, and then fusing these.

Furthermore, nothing prevents from using a more sophisticated fusing system, once we dispose of the single systems. But the presented system selection is very simple and nevertheless performant.

This work is supported by MOBIO, the European project FP7-2007-ICT-1.

8. References

[1] Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., "Factor Analysis Simplified", in: Proc. of ICASSP '05., vol.1, pp. 637–640, March 18–23, 2005.

[2] Matrouf, D., Scheffer, N., Fauve, B., Bonastre, J.-F., "A straightforward and efficient implementation of the factor analysis model for speaker verification", in: Interspeech 2007, 1242–1245, 2007.

[3] Verdet, F. and Matrouf, D. and Bonastre, J.-F. and Hennebert J., "Factor Analysis and SVM for Language Recognition", in: Proc. of Interspeech '09, pp.164–167, Brighton, UK, 2009

[4] The 2009 NIST Language Recognition Evaluation, evaluation plan, http://www.itl.nist.gov/iad/mig/tests/lre/2009

[5] The MISTRAL project, open source platform for biometrics authentification, http://mistral.univ-avignon.fr

[6] Bonastre, J.-F., Wils, F., Meignier, S., "ALIZE, a free toolkit for speaker recognition", in: Proc. of ICASSP '05, vol.1, pp. 737–740, March 18–23, 2005.

[7] Gauvain, J.-L., Lee, C.-H., "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains", Speech and Audio Processing, IEEE Transactions on., vol.2, no.2, 291–298, April 1994.

[8] Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C., "Compensation of Nuisance Factors for Speaker and Language Recognition", Audio, Speech, and Language Processing, IEEE Transactions on, vol.15, no.7, pp.1969–1978, Sept. 2007.

[9] Matějka P., Burget L., Schwarz P., Černocký J., "Brno University of Technology System for NIST 2005 Language Recognition Evaluation", in: Proc. of Odyssey 2006: The Speaker and Language Recognition Workshop, San Juan, PR, pp.57–64, 2006.

[10] CallFriend corpus, telephone speech of 15 different languages or dialects, http://www.ldc.upenn.edu/Catalog

[11] Plchot, O., Hubeika, V., Burget, L., Schwarz, P., Matejka, P., Cernocky, J., "Acquisition of Telephone Data from Radio Broadcasts with Application to Language Recognition – Technical Report", January 2009, Speech@FIT, Brno, Czech Republic, available online at http://www.nist.gov/speech/tests/lre/2009/radio_broadcasts.pdf

[12] Greenberg, C., Martin, A., "2009 NIST Language Recognition Evaluation - Evaluation Overview", presented at NIST LRE 2009 Workshop, June 24–25, 2009, Baltimore, USA
http://www.itl.nist.gov/iad/mig/tests/lre/2009/lre09_eval_results/NIST_LRE09_workshop-presentation_website.pdf

[13] "NIST Language Recognition Evaluation 2009 – System Descriptions", NIST LRE Workshop, June 24–25, 2009.

[14] SPro, a free speech signal processing toolkit, http://www.irisa.fr/metiss/guig/spro/

[15] Burget, L., Matejka, P., Cernocky, J., "Discriminative Training Techniques for Acoustic Language Identification", in: Proc. of ICASSP '06, Toulouse, F, vol.1, pp.209–212, 14–19 May 2006.

[16] Campbell, W. M., Singer, E., Torres-Carrasquillo, P. A. and Reynolds, D. A., "Language Recognition with Support Vector Machines", in Proc. Odyssey: The Speaker and Language Recognition Workshop in Toledo, Spain, ISCA, pp.41–44, May 31–June 3, 2004.