The Albayzin 2012 Language Recognition Evaluation Plan (Albayzin 2012 LRE)

Luis Javier Rodríguez-Fuentes[1], Niko Brümmer[2], Mikel Penagarikano[1], Amparo Varona[1], Mireia Diez[1], and Germán Bordel[1]

[1] GTTS, University of the Basque Country UPV/EHU, Spain
[2] AGNITIO Research, South Africa

luisjavier.rodriguez@ehu.es

Abstract The Albayzin 2012 Language Recognition Evaluation (Albayzin 2012 LRE) is supported by the Spanish Thematic Network on Speech Technology (RTTH)[3] and organized by the Software Technologies Working Group (GTTS)[4] of the University of the Basque Country, with the key collaboration of Niko Brümmer, from Agnitio Research, South Africa, for defining the evaluation criterion and coding the script used to measure system performance. The evaluation workshop will be part of *IberSpeech 2012*, to be held in Madrid, Spain from 21 to 23 November 2012[5].

Keywords: Language Recognition, Albayzin Evaluations, Iberspeech 2012

1 Introduction

As in previous Albayzin LRE editions, the goal of this evaluation is to promote the exchange of ideas, to foster creativity and to encourage collaboration among research groups worldwide working on language recognition technology. To this end, we propose a language recognition evaluation similar to those carried out in 2008 and 2010, but under more difficult conditions. This time the application domain moves from TV Broadcast speech to any kind of speech found in the Internet, and no training data will be available for some of the target languages (aiming to reflect a common situation for low-resource languages).

The change in the application domain pursues two objectives: first, the task should reflect a practical application (in this case, indexing of multimedia content in the Internet); and second, the task should be challenging enough for state-of-the-art systems to yield a relatively poor performance. Results attained in the Albayzin 2010 LRE showed that a possible key to define such a challenging task may be acoustic variability (channel, noise, music, overlapping speakers, etc.), which is inherent to some media (such as the videos posted by people in the Internet) [1].

[3] http://lorien.die.upm.es/~lapiz/rtth/
[4] http://gtts.ehu.es
[5] http://iberspeech2012.ii.uam.es

Audio signals for development and evaluation will be extracted from YouTube videos, which will be heterogeneous regarding duration, number of speakers, ambient noise/music, channel conditions, etc. Besides speech, signals may contain music, noise and any kind of non-human sounds. In any case, each signal will contain a minimum amount of speech. As for previous evaluations, each signal will contain speech in a single language, except for signals corresponding to Out-Of-Set (OOS) languages, which might contain speech in two or more languages, provided that none of them are target languages.

Overall, the Albayzin 2012 LRE introduces some interesting novelties with regard to previous editions (see [2,3] for reference) and NIST Language Recognition Evaluations[6]. The task can be still described as spoken language recognition, but the type of signals used for development and test, the number and identity of target languages and the evaluation criterion are significantly different. In the following sections, all these issues are addressed in detail.

2 The language recognition task

The language recognition task is defined as follows: given a segment of speech and a set of n languages of interest (target languages), produce a likelihood score for each target language plus an additional score for the Out-Of-Set (OOS) language class, based on an automated analysis of the data contained in the segment. Although hard language classification decisions are not required, the likelihood scores are required to be well-calibrated so that they could be used to make Bayes decisions. In closed-set language recognition tests, the last score will not be used to compute performance.

System performance will be evaluated with a calibration-sensitive, multi-class cross-entropy criterion, as explained in Section 4 and Appendix A.

3 Test conditions

3.1 Closed-set vs Open-set

Depending on the set of languages that are allowed to appear in the audio signal, two types of recognition tests are defined:

- In *closed-set recognition*, only target languages are expected to appear in the audio signal. In this case, system performance is computed on the subset of test segments containing speech in one of the target languages.
- In *open-set recognition*, the audio signal may contain any language, either a target language or OOS languages. In this case, system performance is computed on the whole set of test segments, including those containing OOS languages.

This way, systems could be designed specifically or optimized for closed-set or open-set recognition, and research groups could submit separate results for each condition. As we explain in Section 5, whereas the training set will not provide data for OOS

[6] http://nist.gov/itl/iad/mig/lre.cfm

languages, both the development and evaluation sets will include segments in OOS languages (with different distributions). The set of OOS languages will not be disclosed to participants.

3.2 Plenty of Training vs Empty Training

Two different conditions are defined depending on the availability of training materials for target languages. This way, we aim to check to what extent the availability of training materials (and thus specific models) for target languages affects system performance. In fact, two separate tasks are defined depending on this condition, since they involve disjoint sets of target languages:

- The first condition, called *Plenty of Training*, involves 6 target languages (those used for the Albayzin 2010 LRE): Castilian Spanish, Catalan, Basque, Galician, Portuguese and English. For all of them, a large amount of training data (around 18 hours of speech per language) will be supplied, specifically speech signals recorded from TV broadcasts used to build the Kalaka-2 database [4]. Development signals (YouTube audio) will be also supplied, both for target languages (between 100 and 200 signals per language) and for Out-Of-Set languages (around 500 signals), to allow tuning systems for open-set tests.
- The second condition, called *Empty Training*, involves 4 target languages: French, German, Greek and Italian, for which no training materials will be supplied. Only development signals (YouTube audio) will be supplied, both for target languages (between 100 and 200 signals per language) and for Out-Of-Set languages. Under this condition, the training and development data supplied for target languages in the *Plenty of Training* condition can be also used. Note also that the set of development signals provided for OOS languages is shared by both conditions.

In both cases, participants are only allowed to use the data provided for this evaluation. Thus, participants cannot benefit from other databases available to them nor will they have to invest time in collecting data. Systems will be built strictly from the data provided, which should be seen as a common starting condition, necessary for the comparison of systems to depend only on the applied technologies. The only exception to this rule and for the sole purpose of preventing some approaches to be penalised, is that auxiliary subsystems trained on external data (e.g. phonetic decoders) are allowed.

3.3 Primary vs contrastive systems

Unlike previous editions of the Albayzin LRE, neither the duration nor the acoustic conditions (presence of background noise or music, etc.) of test segments will be taken into account to define different evaluation tracks. There will be just 4 tracks, combining the two tasks described in Section 3.2 and the two recognition modalities described in Section 3.1:

- Plenty of Training (6 target languages), Closed-Set Recognition (briefly, PC)
- Plenty of Training (6 target languages), Open-Set Recognition (briefly, PO)

– Empty Training (4 target languages), Closed-Set Recognition (briefly, EC)
– Empty Training (4 target languages), Open-Set Recognition (briefly, EO)

The first track (PC) is mandatory, which means that participants must submit at least one complete set of recognition results for that condition. Note that a complete set of results in the PC track comprises the 6 scores yielded by the system plus a fake score for OOS languages (just to fit the file format, as specified in Section 6.1) for each one of the test segments. The PO, EC and EO tracks are optional.

Participants can submit multiple sets of recognition results (each corresponding to a different system) for each track, but they are required to specify one of them as *primary system*, the remaining being *contrastive systems*. To determine the ranking in each track (in terms of the primary evaluation measure, as defined in Section 4), only primary systems will be taken into account, though all the submitted systems will be evaluated and presented in tables and graphs.

4 Evaluation of system performance

For the Albayzin 2012 Language Recognition Evaluation, we follow the example of the upcoming 2012 NIST Speaker Recognition Evaluation[7], and we change the submission format to a probabilistic form. In the previous Albayzin LREs (as well as in all previous NIST LREs and SREs), participants had to submit the outputs of their systems in the form of hard decisions. This year, the format will be soft, probabilistic decisions in the form of calibrated language log-likelihoods.

The new submission format will be accompanied by a new primary evaluation criterion, called *multiclass cross-entropy*. This criterion simultaneously evaluates discrimination and calibration, so that highly discriminative systems, with well-calibrated outputs, will perform well. The primary criterion will be accompanied by a secondary criterion, where the evaluator optimally re-calibrates submissions. The secondary criterion can be used for comparison of the discrimination, when calibration is not of immediate interest and also to analyze the quality of calibration. This evaluation recipe was first published in [5].

4.1 Evaluation task and submission format

As noted above, in this LRE there will be two separate tasks involving either $n = 6$, or $n = 4$ target languages. Given n target languages, we define $m = n + 1$ *language classes*, denoted L_i, for $i = 1, 2, \ldots, m$, where the first n classes are the n target languages, while L_m is the class of all OOS languages.

The evaluation data will consist of some large number, T, of audio segments, denoted $s_1, s_2, \ldots, s_T$. Participants must assume that every segment contains either one of the target languages or an OOS language, as defined above. If this assumption turns out after the fact to be untrue for some segments, for example where segments contain no speech, then such segments will be removed when tallying the final evaluation criteria.

[7] See the SRE'12 Evaluation Plan at `http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v11-r0.pdf`.

The task is to recognize the language class to which each segment, s_t, belongs. But we do not require hard language classifications. Instead, we require for every s_t, m *language log-likelihoods* of the form:

$$\ell_{it} = k_t + \log P(s_t|L_i), i = 1, 2, \ldots, m \tag{1}$$

We require the log-likelihoods to be *finite*: $-\infty < \ell_{it} < \infty$. Here k_t is an arbitrary real constant that may depend on the segment, s_t, but not on the class, L_i. We use *natural logarithms* throughout.

In summary, when T speech segments are given, the submission will be a m-by-T matrix of language class log-likelihoods. Each column of this matrix is a log-likelihood-vector, $\boldsymbol{\ell}_t = (\ell_{1t}, \ell_{2t}, \ldots, \ell_{mt})$, which represents the soft, probabilistic classification given by the system under evaluation, for segment s_t.

4.1.1 Open-set vs Closed-set For simplicity, we specify only the above m-class submission format. For the open-set condition, the whole m-by-T matrix will be used for evaluation. For the closed-set condition, the matrix will be edited: all the columns where the audio segment does not contain one of the n target languages will be disregarded, and the bottom row with the likelihoods corresponding to the L_m class (OOS languages) will be also disregarded.

4.1.2 Discussion It may seem surprising to some that we do not ask for language posteriors of the form $P(L_i|s_t)$ to be submitted instead. However, such posteriors must depend implicitly on some prior distribution over language classes. By definition, this prior cannot be extracted from the speech. In some areas of machine learning, for example in a phone recognizer, it would make sense for the recognizer to effectively learn this prior from development data. But in the context of language recognition, the proportions of languages in the development data could be very different from the priors that one would want to use in applications. The prior will also vary between applications. For this reason, we prefer the above-defined, prior-independent likelihood format for submission.

It must further be mentioned that the m-component log-likelihood vector, $\boldsymbol{\ell}_t$, is a redundant representation. To see this, choose without loss of generality, $k_t = -\log P(s_t|L_m)$, which would zero the last component, leaving only $m - 1$ components. We could do the same with any of the other components. But any such $(m - 1)$-component representation must be asymmetrical and is therefore less intuitive and more difficult to work with in practice.

Finally, we choose to use a logarithmic format, because experience has shown that scores from typical classifiers behave like (shifted and scaled) log-likelihoods. The logarithmic format therefore preserves the look and feel of typical scores.

4.2 Primary evaluation criterion

The primary evaluation criterion to be defined below can be interpreted as a form of empirical, multiclass cross-entropy. To compute it, the evaluator specifies a prior distribution over language classes, so that Bayes' rule can be used to map the submitted

log-likelihoods to language class posteriors. The goodness of these posteriors is then evaluated via the logarithmic cost function. A weighted average of the logarithmic cost over all audio segments forms the cross-entropy criterion.

In order to facilitate comparison of the cross-entropies across different tasks, which have different perplexities, we show how to present cross-entropy in the form of *relative confusion*, a measure closely related to perplexity . The following subsections give more detail.

4.2.1 Prior Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_m)$ represent a prior distribution over the $m = n + 1$ language classes, so that $\pi_i = P(L_i|\boldsymbol{\pi})$. We specify:

$$\boldsymbol{\pi} = \left(\frac{1 - \pi_m}{n}, \ldots, \frac{1 - \pi_m}{n}, \pi_m \right) \tag{2}$$

For the closed-set condition, we specify $\pi_m = 0$. For the open-set condition, we specify $\pi_m = \frac{1}{m}$.

4.2.2 Posterior Given a log-likelihood-vector, $\boldsymbol{\ell}_t = (\ell_{1t}, \ell_{2t}, \ldots, \ell_{mt})$, submitted by a system under evaluation, the evaluator calculates the posterior distribution:

$$P(L_i|\boldsymbol{\ell}_t, \boldsymbol{\pi}) = \frac{\pi_i \exp(\ell_{it})}{\sum_{j=1}^{m} \pi_j \exp(\ell_{jt})} \tag{3}$$

Note that the arbitrary constant k_t, as defined in (1), cancels. Also note that when $\pi_m = 0$, then the corresponding likelihood, ℓ_{mt}, is effectively disregarded.

The mapping (3) is the familiar softmax function. We say that $\boldsymbol{\ell}_i$ favours class i, when $\ell_{it} + \log \pi_i \gg \ell_{jt} + \log \pi_j$, for all $j \neq i$, in which case $P(L_i|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \approx 1$ and $P(L_j|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \approx 0$.

In what follows, we shall refer to the whole posterior distribution as:

$$\boldsymbol{\Pi}_t = \big(P(L_1|\boldsymbol{\ell}_t, \boldsymbol{\pi}), \ldots, P(L_m|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \big) \tag{4}$$

4.2.3 Logarithmic cost function For every audio segment, s_t, the system under evaluation submits the log-likelihood-vector, $\boldsymbol{\ell}_t$. The evaluator has access to the *true class label* for segment s_t, which we denote $L_{\text{true}(t)} \in \{L_1, \ldots, L_m\}$. This allows the evaluator to compute a measure of goodness for $\boldsymbol{\ell}_t$, in the form of the *logarithmic cost function*:

$$C_{\log}(\boldsymbol{\Pi}_t|L_{\text{true}(t)}) = -\log P(L_{\text{true}(t)}|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \tag{5}$$

To get a feeling for this cost function, note that when $\boldsymbol{\ell}_t$ favours the correct language class, then $P(L_{\text{true}(t)}|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \approx 1$, so that $C_{\log}(\boldsymbol{\Pi}_t|L_{\text{true}(t)}) \approx 0$. But if $\boldsymbol{\ell}_t$ favours an incorrect class, then $P(L_{\text{true}(t)}|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \approx 0$ and so $C_{\log}(\boldsymbol{\Pi}_t|L_{\text{true}(t)}) \gg 0$.

For the interested reader, appendix A gives further analysis and motivation of the logarithmic cost function.

4.2.4 Multiclass cross-entropy We form our *primary evaluation criterion*, known as *multiclass cross-entropy*, by a weighted average of the logarithmic cost:

$$\mathcal{C}_{\text{mce}} = \sum_{i=1}^{m} \frac{\pi_i}{\|\mathcal{T}_i\|} \sum_{t \in \mathcal{T}_i} -\log P(L_i|\boldsymbol{\ell}_t, \boldsymbol{\pi}) \tag{6}$$

where $\mathcal{T}_i$ is the subset of indices for segments of class i. By $\|\mathcal{T}_i\|$ we mean the number of segments of language class i. Note that for the closed-set case, when $\pi_m = 0$, all segments of class L_m are effectively ignored[8].

The default system. We already know that the logarithmic cost function is small for a good $\boldsymbol{\ell}_t$ and large for a bad $\boldsymbol{\ell}_t$. Let us now examine a convenient reference value to help to further understand how cross-entropy behaves.

We define *the default system* as the one that cannot make up its mind about the language class and outputs $\ell_{it} = k_t$ for every t. This gives $P(L_i|\boldsymbol{\ell}_t, \boldsymbol{\pi}) = \pi_i$ for every i, t. In other words, the default system effectively ignores all information in the audio signal, so that the posterior and prior are the same. The cross-entropy for the default system gives the reference value[9]:

$$\mathcal{C}_{\text{def}} = \sum_{i=1}^{m} -\pi_i \log \pi_i \tag{7}$$

which is just the prior entropy[10].

If a submitted system has $\mathcal{C}_{\text{mce}} \geq \mathcal{C}_{\text{def}}$, then it does not improve upon the default system. We would expect good systems to have $\mathcal{C}_{\text{mce}} < \mathcal{C}_{\text{def}}$.

Confusion. To facilitate interpretation of cross-entropy, we define the *confusion* of the system under evaluation as:

$$F_{\text{mce}} = \exp(\mathcal{C}_{\text{mce}}) - 1 \tag{8}$$

similarly, the *prior confusion* (confusion of the default system) is:

$$F_{\text{def}} = \exp(\mathcal{C}_{\text{def}}) - 1 \tag{9}$$

Since cross-entropy is non-negative, so is confusion—a perfect system would have zero confusion. To get an intuitive feeling for confusion, consider the prior confusion for the closed-set case where have a flat prior over n classes, so that $\mathcal{C}_{\text{def}} = \log n$ and $F_{\text{def}} = n - 1$. This can be interpreted as the number of *wrong* alternatives. Notice that confusion is closely related to perplexity (the *total* number of alternatives).[11] Here we choose to use confusion, rather than perplexity, in order to facilitate comparison across

[8] When $\pi_m = 0$, $P(L_m|s_t, \boldsymbol{\pi}) = 0$ and $-\log P(L_m|s_t, \boldsymbol{\pi}) = \infty$, but we can use the limit: $\lim_{\pi_m \to 0} \pi_m \log P(L_m|s_t, \boldsymbol{\pi}) = 0$.

[9] Again, for the case $\pi_m = 0$, we use $\lim_{\pi \to 0} \pi \log \pi = 0$.

[10] Shannon's entropy.

[11] perplexity = confusion + 1

different tasks with different prior perplexities. We do this by defining *actual relative confusion*:[12]

$$F_{\text{act}} = \frac{F_{\text{mce}}}{F_{\text{def}}} \tag{10}$$

(We call this criterion 'actual', to contrast with an auxiliary relative confusion criterion, F_{dis}, to be defined below.) The relative confusion is the factor by which the system has changed (hopefully reduced) the prior confusion. The reference value for relative confusion is 1. Badly calibrated systems that have relative confusion greater than one are doing worse than the default system. We expect good systems to have relative confusion below 1. A perfect system would have relative confusion of zero.

4.3 Auxiliary evaluation criterion

The evaluator can *recalibrate* the log-likelihoods, ℓ_{it}, submitted by a system as:

$$\ell'_{it} = \alpha\ell_{it} + \beta_i \tag{11}$$

where α and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)$ are calibration constants. In vector notation, (11) becomes: $\boldsymbol{\ell}'_t = \alpha\boldsymbol{\ell}_t + \boldsymbol{\beta}$.

Let $\mathcal{C}'_{\text{mce}}$ denote the cross-entropy obtained by applying (6) to ℓ'_{it}. The auxiliary evaluation criterion is defined as:

$$\mathcal{C}_{\text{min}} = \min_{\alpha,\boldsymbol{\beta}} \mathcal{C}'_{\text{mce}} \tag{12}$$

This minimization is just a form of multiclass logistic regression. This is an unconstrained, convex minimization problem and can be performed using standard numerical optimization algorithms. Note that if we choose $\alpha = \beta_i = 0$, the default system results, so that if the calibration parameters are freely optimized, we must have $\mathcal{C}_{\text{min}} \leq \mathcal{C}_{\text{def}}$.

Below, we use the auxiliary criterion, $\mathcal{C}_{\text{min}}$, for two purposes: comparison of discrimination and analysis of calibration.

4.3.1 Comparison of discrimination between systems Let us assume for argument's sake that the evaluee used a restricted affine transformation of the same form as (11) in order to transform raw uncalibrated scores to calibrated log-likelihoods[13]. If the evaluator now recalibrates, the composition of those two transforms is still of the same form. By applying (11), the evaluator therefore effectively removes the calibration done by the evaluee, recovering the raw scores, to which a new, optimal calibration transform is then applied. This gives the interpretation that $\mathcal{C}_{\text{min}}$ is the best performance that the evaluee could have obtained with an optimal calibration of this form. We can

[12] If we had used relative perplexity, the perfect system would have had relative perplexity of $\frac{1}{n}$, for n alternatives, which does not facilitate comparison across tasks having a different number of alternatives.

[13] Practical experience has shown that this is indeed a good strategy.

therefore use this criterion to compare the discrimination potential of systems, regardless of problems that may have been introduced by bad calibration. Again, we find it convenient to do so via relative confusion. Let us define the minimized confusion as $F_{\text{min}} = \exp(\mathcal{C}_{\text{min}}) - 1$ and the relative version as:

$$F_{\text{dis}} = \frac{F_{\text{min}}}{F_{\text{def}}} \tag{13}$$

where 'dis' is mnemonic for discrimination. As pointed out above, we must have $F_{\text{dis}} \leq 1$.

4.3.2 Analysis of calibration For one system at a time, we can also compare F_{act} to F_{dis} to judge how good the calibration of that system was. We define the *calibration loss* as the additional relative confusion introduced by suboptimal calibration:

$$F_{\text{cal}} = \frac{F_{\text{act}} - F_{\text{dis}}}{F_{\text{dis}}} \tag{14}$$

The optimization guarantees that $F_{\text{cal}} \geq 0$, with perfect calibration at zero.

4.3.3 Factorization The above definitions provide the following factorization of the primary criterion, F_{act}:

$$F_{\text{act}} = (1 + F_{\text{cal}})F_{\text{dis}} \tag{15}$$

where $F_{\text{cal}} \geq 0$ and $0 \leq F_{\text{dis}} \leq 1$. This emphasizes the sensitivity of the primary criterion to both discrimination and calibration. To get a small value for F_{act}, a system needs to minimize both factors.

4.4 Language pairs

The log-likelihood submission format makes it possible to apply a variety of different evaluation criteria to the same submission. We can use this opportunity to focus, for example, on *pairs of languages*. To do this, we choose some $1 \leq i < j \leq n$ and set $\pi_i = \pi_j = \frac{1}{2}$ and the rest of the prior components to 0. Then we can proceed as before, calculating cross-entropy for each of the $\binom{n}{2}$ language pairs. Some language pair performance analysis of this kind may be done by the evaluator, but the focus of this evaluation is on the full multiclass recognition problem, rather than on pairs.

4.5 Summary of evaluation criteria

The primary evaluation criterion for comparison between systems is F_{act}, which is sensitive to both calibration and discrimination. It is calculated by applying equations (3), (6), (7) , and (10), with the prior $\boldsymbol{\pi}$ specified in section 4.2.1.

The secondary criterion for comparison between systems is F_{dis}, which is sensitive only to discrimination, because calibration is redone by the evaluator. It is calculated by applying equations (11), (12) and (13).

For a given system, the calibration loss is $F_{\text{cal}} = \frac{F_{\text{act}} - F_{\text{dis}}}{F_{\text{dis}}}$.

All of F_{dis}, F_{act} and F_{cal} can be expressed conveniently as percentages. The first is bounded by 100%, but the others can be arbitrarily large for badly calibrated systems.

A MATLAB toolkit will be made available for participants to calculate these criteria.

5 Data

5.1 Training data

Training data provided for this evaluation amount to around 108 hours of speech, with 18 hours on average for each one of the 6 target languages considered in the Plenty-of-Training condition. Speech files have been extracted from the materials used to produce KALAKA-2 (the database created for the Albayzin 2010 LRE) [3]. All of them are continuous excerpts (of different durations) from multi-speaker TV broadcast recordings, featuring various speech modalities and diverse environment conditions.

Broadcasts were recorded through a home connection to cable TV, by means of a Roland Edirol R-09 ultra-light digital audio recorder[14], and stored in CD quality (16 bits/sample, 44.1 kHz, stereo) audio files. Recordings were done at different times: April-September 2008 (Basque, Catalan, Galician, Spanish, English and Portuguese); October-November 2008 (English), April-May 2010 (English and Portuguese) and August-September 2010 (Basque, Catalan, Galician and Spanish). Audio signals were downsampled to 16 kHz, left and right channels being averaged into one single channel, and finally stored in WAV files (PCM, 16 kHz, single channel, 16 bits/sample), by means of *SoX*[15].

The training dataset consists of two disjoint subsets, including clean speech (around 86 hours) and noisy speech (around 22 hours), respectively. Clean-speech segments are high SNR speech signals, maybe with short fragments of noisy and/or overlapped speech. Noisy-speech segments include noisy and/or overlapped speech, maybe with short fragments of clean speech. Different and variable types of noise may appear: street, music, cocktail party, laughs, clapping, etc. However, telephone-channel speech has not been included in any case. Most speech overlaps appeared in hot spots of informal debates in late night shows, magazines, etc. which, on the other hand, featured clean-channel and quiet-background (studio) conditions. In all cases, each segment contains speech in a single language.

5.2 Development and evaluation data

The development and evaluation datasets will be similar in size and structure. Each one will consist of between 100 and 200 audio segments per target language, plus additional segments in OOS languages (needed to tune and evaluate systems in the open-set condition), amounting to around 2000 segments. Audio segments have been extracted from YouTube videos, and then audited and labelled by human *experts*. Each segment

[14] http://www.roland.com/products/en/R-09

[15] http://sox.sourceforge.net

is between 30 and 120 seconds long and is guaranteed to include above a minimum amount of speech (from one or more speakers) either in one of the target languages or in one or more OOS languages. Note that some segments may feature quite challenging background and/or channel conditions. Speech segments will be given random names, so that language labels are kept undisclosed.

6 Information for participants

All the registered participants will be given access to a web page and permission to download the training and development data, along with a keyfile and a scoring script (needed to evaluate system performance during the development phase). Registration involves the commitment to use data exclusively for research purposes, distribution being allowed only with explicit permission. After the evaluation, the registered participants are allowed to use the data to develop or evaluate their own systems, provided that they acknowledge that use by means of a suitable reference:

> **KALAKA-3.** Speech database created for the Albayzin 2012 Language Recognition Evaluation. Produced by the Software Technologies Working Group (GTTS), University of the Basque Country UPV/EHU, Spain.

After receiving training and development data, participants will have more than two months for system development. The evaluation dataset will be also released via web. There will be three weeks to process evaluation data and send back recognition results (see Section 6.4 for details). The keyfile will be released two weeks after the deadline for submitting recognition results.

The ranking of primary systems in all conditions will be determined by taking into account the primary evaluation criterion, F_{act}, as defined in Section 4. As noted above, all the participants must submit recognition results for a primary system in the Plenty-of-Training Closed-set (PC) condition, which is mandatory.

6.1 Data organization

6.1.1 Training data The training dataset will consist of two elements:

- *data* - a directory containing speech segments (WAV files) for the 6 target languages defined for the Plenty-of-Training condition. Data are organized into 6 folders (one per target language), each one with two sub-folders, containing clean and noisy speech, respectively.
- *doc* - a directory containing documentation: the evaluation plan, authoring information, the conditions for using the database, etc.

6.1.2 Development and evaluation data The development and evaluation datasets will both consist of four elements:

- *seg.ndx* - a text file containing the list of segments to be used in all the tests.

– *data* - a directory containing audio segments (WAV files), their names being random alphanumeric strings followed by the *.wav* extension.
– *scoring* - a directory containing the script that must be used to measure system performance, along with a readme file to help using it and the keyfile[16].
– *doc* - a directory containing documentation: the evaluation plan, authoring information, the conditions for using the database, etc.

6.1.3 System output format Recognition results for any given condition must be sent in a text file with a line per test segment, each line consisting of $n + 4$ blank-separated fields, where n is the number of target languages, which depends on the task ($n = 6$ for *Plenty-of-Training* and $n = 4$ for *Empty-Training*). These fields are defined for this evaluation as follows:

1. A code indicating the task for which the system is designed: **Plenty** or **Empty**.
2. A code indicating whether the system is designed to deal just with target languages (**Closed**) or also with OOS languages (**Open**).
3. The name of the audio segment (without the *.wav* extension).
4. A sequence of $m = n + 1$ blank-separated scores: one per target language plus the score of OOS languages, in the following order:
   – *Plenty-of-Training*: Basque, Catalan, English, Galician, Portuguese, Spanish, OOS.
   – *Empty-Training*: French, German, Greek, Italian, OOS.

An example of how the system output should look like is shown in Figure 1. In the example, each line contains the likelihood scores corresponding to the 6 target languages handled in the *Plenty-of-Training* condition, followed by the likelihood score corresponding to OOS languages. Since the system operates in closed-set mode, it does not really compute any likelihood score for OOS languages; instead, it provides a fixed value (0.0000) just to fit the file format specification.

```
Plenty Closed xxyyffaa -0.5678 -0.0034  0.6723  1.4332 -7.0032  5.0065 0.0000
Plenty Closed gghhjjbb  0.3421 -0.9734 -1.5671 -3.0087 -9.3215 -3.7666 0.0000
Plenty Closed ppllkkaa -1.7608  0.4559 -0.3946 -0.0004  2.1444 -0.0342 0.0000
Plenty Closed qqttzzii  0.1799 -5.5911 -0.0795  0.1005  0.1871  -6.998 0.0000
```

Figure 1. An example file containing the recognition results of an hypothetical system in the *Plenty-of-Training Closed-set* (PC) condition, for a test set consisting of 4 audio files: *xxyyffaa.wav*, *gghhjjbb.wav*, *ppllkkaa.wav* and *qqttzzii.wav*.

[16] Obviously, the keyfile for the evaluation dataset will be released later.

6.2 Submissions

6.2.1 Submission procedure Recognition results in the format described above, along with a file describing the system or systems, will be sent as attached files by e-mail to the following address:

luisjavier.rodriguez@ehu.es

Filenames will be built according to the following pattern:

<GroupID>_<ConditionID>_<SystemID>.out

where *GroupID* is the name or acronym of the research group, *ConditionID* is one of the values PC, PO, EC or EO, and *SystemID* is a code identifying the system as primary (pri) or contrastive (con1, con2, etc.). For instance, if the research group GTTS sends recognition results for two systems, one primary and one contrastive, in the PC condition, their filenames would be named as follows:

GTTS_PC_pri.out
GTTS_PC_con1.out

6.2.2 System description Research groups must provide a **PDF file** with the description of each submitted system. If multiple systems are submitted for a particular test condition, the description must explicitly designate one of them as the primary system, the remaining being contrastive systems. The system description should give the readers a good sense of what the system is about, keeping in mind the following guidelines:

- Write for your audience. Remember that the reader is not you but other system developers who may not be familiar with your technique/algorithm. Clearly explain your method so they can understand what you did.
- A superficial description would leave other system developers clueless of what you did. Be as complete as possible, but not to the extent of including pseudo-code. Include all the relevant information, in such a way that other groups can build the system on their own.
- Include references to techniques, algorithms, subsystems, etc. used by your systems but not described in detail in the document.
- Avoid jargon and abbreviation without any prior context.

To keep formal homogeneity, it is *mandatory* to edit system descriptions by means of the IberSpeech 2012 paper submission templates (either WORD or LaTeX), available at the following site: **http://iberspeech2012.ii.uam.es/**. The system description should, at least, include the following sections:

1. Introduction
2. System A (name of the submitted system)
   (a) System description
   Clearly describe the methods and algorithms used in system A.

(b) Train and development data
Describe all the data and/or systems directly or indirectly used in developing system A, including the source, acquisition conditions, size, publishing year and any other pertinent information.

(c) Processing speed
Compute the speed of language recognition, in terms of the Real-Time factor (×RT), defined as the total amount of CPU time required to do the processing divided by the total amount of processed audio. Include the specs for the CPU and the memory used. Note that the CPU time required to perform language recognition includes acoustic modeling, decision processing and I/O and is measured in terms of elapsed time on a single CPU, start to finish. Systems that are not completely pipelined are not penalized, however, and time intervening between separate processes need not be included in tallying elapsed time.

3. System B (name of another submitted system)
This section is similar to section 2 but for another system. If system B is a contrastive system, note the differences from the primary system. A new section should be added for each submitted system.
4. References
List of papers relevant to the techniques, algorithms, data, etc. used by the submitted systems.

6.3 Summary of rules

We summarize here the basic rules and restrictions that must be observed by all participants:

– Interested groups must register for the evaluation before July 16th 2012, by contacting the organizing team at:

luisjavier.rodriguez@ehu.es

with copy to the Chairs of the Albayzin 2012 Evaluations:

javier.gonzalez@uam.es
javier.tejedor@uam.es

and providing the following information:

- Group name
- Group ID
- Institution
- Contact person
- Email address
- Postal address

– Starting from June 20th 2012, and once registration data are validated, the training and development datasets will be released via web (only to registered participants). A *wiki* will be activated, featuring public and restricted pages, aiming to improve communication and collaboration between the organizing team and research groups participating in the evaluation.

– Registered groups commit themselves to use the provided data only for research purposes, distribution being allowed only with explicit permission of the Albayzin 2012 LRE organizing team. Registered participants are allowed to use the data to develop or evaluate their own systems, provided that they acknowledge that use by means of the following reference:

> **KALAKA-3.** Speech database created for the Albayzin 2012 Language Recognition Evaluation. Produced by the Software Technologies Working Group (GTTS), University of the Basque Country UPV/EHU, Spain.

– The evaluation dataset is planned to be released by September 3rd 2012. Recognition results along with system descriptions must be submitted to the organizing team by the established deadline: September 24th 2012, 24:00 GMT+1, according to the data format and submission procedure specified in Sections 6.1 and 6.2.

– For each test segment, the information available to the system is limited to that specified in Section 2. In particular:

  - Each test segment must be processed on an independent way, that is, without any information about other segments.
  - Listening to the evaluation data or any other human interaction with the data is not allowed before all test results have been submitted.

– Recognition results (according to the format specified in Section 6.1.3) must include all the test segments, for whatever test condition.

– Participants may submit results for different (contrastive) systems. However, for each test condition for which results are submitted, there must be one (and only one) primary system.

– Each submission must include a group identifier, a test condition identifier (PC, PO, EC or EO) and a file of recognition results for each system.

– Research groups must provide a description of the submitted systems, according to the guidelines given in Section 6.2. For the sake of formal homogeneity, **it is mandatory** to edit system descriptions by means of the IberSpeech 2012 paper submission templates available at **http://iberspeech2012.ii.uam.es/**.

– Each participating site is required to send one or more representatives to the Evaluation Workshop, to be held in Madrid (Spain) as part of IberSpeech 2012 (November 21-23, 2012). Representatives will be expected to give a presentation of their systems and to participate in discussions on the current state of the technology and future plans. The workshop will be open to participants in the Albayzin 2012 LRE and to researchers registered to IberSpeech 2012.

– This plan might be modified due to new restrictions or unplanned needs, to detected errors or inaccuracies. Updated versions of this plan, if any, will be announced through the IberSpeech 2012 website and e-mailed to the registered participants.

6.4 Schedule

- *May 18, 2012*
  - The evaluation plan is released through the website of IberSpeech 2012.
  - Registration for Albayzin 2012 LRE opens.
- *June 20, 2012*.
  - The training and development datasets are released via web.
  - A *wiki* is activated to improve communication and collaboration between the registered participants and the organizing team.
- *July 16, 2012*.
  - Registration for Albayzin 2012 LRE closes.
- *September 3, 2012*.
  - The evaluation dataset is released via web.
  - System submission (via e-mail) opens.
- *September 24, 2012 (24:00, GMT+1)*.
  - Deadline for submitting system results.
  - Deadline for submitting system descriptions.
- *October 15, 2012*.
  - Preliminary results in all conditions and the keyfile for the evaluation dataset are released to participants through the wiki.
- *November 21-23, 2012*.
  - **IberSpeech 2012, Madrid, Spain.**
  - *Albayzin 2012 LRE Workshop*: presentation of systems and discussion of results.
  - Plenary session: summary of results.

Appendices

A Motivation and analysis of the logarithmic cost function

Here we provide further motivating analysis of the logarithmic cost function, which forms the basis of all our evaluation criteria. In particular, we show that:

- The logarithmic cost is a *proper scoring rule*, and as such belongs to a family of cost functions which are well suited towards evaluating the goodness of probabilistic inferences.
- The logarithmic cost has an interpretation in terms of an expected value of the more traditional misclassification cost.

This appendix is based on Niko Brümmer's Ph.D. Thesis [6], in which the interested reader may find further explanations, motivations, derivations and references.

A.1 Logarithmic cost as proper scoring rule

A *proper scoring rule* is a special cost function that measures the goodness of *probability distributions* relative to a truth reference. The function maps the distribution and truth reference to a real-valued cost, where smaller costs indicate 'better' distributions. The defining property of a proper scoring rule is that its expected value is minimized when the distribution being scored is the same as the distribution w.r.t. which the expectation is taken. Proper scoring rules encourage *honesty* (calibration) and *diligence* (discrimination) in the person (or machine) whose goodness is being judged by such rules. We demonstrate these properties for the logarithmic cost function.

Proper scoring rules are traditionally defined in terms of scoring the goodness of *probability distributions*, whereas in the rest of this document we were interested in evaluating the goodness of *log-likelihood-vectors*. However, if the prior is given, as it is here, then there is in essence a one-to-one mapping between probability distributions and log-likelihood-vectors. So here we can follow the traditional notation in order to analyse proper scoring rules in terms of probability distributions.

Let there be m language classes. Given a speech segment, let a language recognizer (machine or human) calculate to the best of his/her/its ability, a (posterior) probability distribution over language classes, denoted $\mathbf{p} = (p_1, \ldots, p_m)$. Now for this trial, the recognizer could submit $\mathbf{p}$ as calculated, but perhaps there might be a good reason to instead submit some other distribution $\mathbf{q} = (q_1, \ldots, q_m)$. What can we expect if we calculate $\mathbf{p}$, but submit $\mathbf{q}$ to evaluation by logarithmic cost? Since the recognizer does not know the true language class and its best distribution for the class is $\mathbf{p}$, its expectation should be based on $\mathbf{p}$. The expected cost when submitting $\mathbf{q}$ would then be the *cross-entropy*:

$$\begin{aligned}\sum_{i=1}^{m} p_i C_{\log}(\mathbf{q}|L_i) &= \sum_{i=1}^{m} -p_i \log q_i \\ &= \sum_{i=1}^{m} -p_i \log p_i + \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} \\ &= \mathrm{H}(\mathbf{p}) + \mathrm{KL}(\mathbf{p}\|\mathbf{q})\end{aligned} \tag{16}$$

where $\mathrm{H}(\mathbf{p})$ is Shannon entropy and $\mathrm{KL}(\mathbf{p}\|\mathbf{q})$ is KL-divergence. Since KL-divergence is non-negative and reaches zero if and only if $\mathbf{q} = \mathbf{p}$, the expected value is minimized uniquely at $\mathbf{q} = \mathbf{p}$. To optimize the expected cost, the recognizer should therefore submit $\mathbf{p}$ as calculated and not some other distribution. This is the mechanism by which the logarithmic cost (and any other proper scoring rule) ensures *honesty* in the recognizer.

The same decomposition shows that honesty is not enough. Merely calculating some default $\mathbf{p}$ and then submitting it is not a winning strategy. The recognizer (or its designer) must also work *diligently* to minimize the entropy, $H(\mathbf{p})$, of its output, so that the posterior uncertainty about the language class is as small as possible.

A.2 Logarithmic cost interpreted as expected misclassification cost

Let us now construct a different proper scoring rule, induced by a more traditional, application-based cost function. Then we show how this is related to the logarithmic cost function.

We start by specifying a cost function which penalizes misclassification. For now, let the recognizer submit a hard decision, say L_j, which is the *recognized class*. Let the true class be L_i. We define the *weighted misclassification cost function* as:

$$C_{\boldsymbol{\eta}}(L_j|L_i) = \frac{1}{m-1} \begin{cases} 0 & \text{if } j = i, \\ \frac{1}{\eta_i} & \text{if } j \neq i. \end{cases} \tag{17}$$

which is parameterized by $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_m)$, a vector of weights that lives in the interior of the standard simplex, with $0 < \eta_i < 1$ and $\sum_{i=1}^{m} \eta_i = 1$.

If we choose $\eta_i = \frac{1}{m}$, then $C_{\boldsymbol{\eta}}$ is just a scaled version of the familiar *zero-one cost function*, which effectively computes the misclassification error-rate when averaged over a supervised evaluation database. In other words, we have generalized zero-one cost.

Next, we transform $C_{\boldsymbol{\eta}}$ into a proper scoring rule. Let the recognizer now submit a probability distribution, $\mathbf{q} = (q_1, \ldots, q_m)$, instead of a hard decision. The evaluator now evaluates the goodness of $\mathbf{q}$ as the cost of the minimum-expected-cost Bayes decision made with $\mathbf{q}$. This proper scoring rule is defined as:

$$C^*_{\boldsymbol{\eta}}(\mathbf{q}|L_i) = C_{\boldsymbol{\eta}}(L^*_{\mathbf{q}}|L_i) \tag{18}$$

where $L^*_{\mathbf{q}} \in \{L_1, \ldots, L_m\}$ is the Bayes decision made with $\mathbf{q}$:

$$L^*_{\mathbf{q}} = \arg\min_{L_j} \sum_{i=1}^{m} q_i C_{\boldsymbol{\eta}}(L_j|L_i) \tag{19}$$

This is a very natural way to evaluate the goodness of $\mathbf{q}$. The reasoning is that the recognizer output, $\mathbf{q}$, should be designed so that the user of $\mathbf{q}$ can apply it to make minimum-expected-cost Bayes decisions. The consequence of applying $\mathbf{q}$ in this way when the real class is L_i is just $C^*_{\boldsymbol{\eta}}(\mathbf{q}|L_i)$.

Moreover, it is easy to show that this construction satisfies the definition of a proper scoring rule:

$$\sum_{i=1}^{m} p_i C^*_{\boldsymbol{\eta}}(\mathbf{p}|L_i) \leq \sum_{i=1}^{m} p_i C^*_{\boldsymbol{\eta}}(\mathbf{q}|L_i) \tag{20}$$

for any distributions $\mathbf{p}$ and $\mathbf{q}$. Note however, that the logarithmic cost function is a *strictly* proper scoring rule, whose expectation is minimized uniquely at $\mathbf{q} = \mathbf{p}$. In contrast, $C^*_{\boldsymbol{\eta}}$ is non-strict and its expectation does not have a unique minimum. For the honesty-inducing purpose, strictness is to be preferred, but we show how to mend this problem.

If we persevere with $C^*_{\boldsymbol{\eta}}$, we would have to choose some value for $\boldsymbol{\eta}$ in order to define a concrete evaluation criterion. But, to better exercise the decision-making ability of recognizers, we could use instead a *combination* of many different values. This works because convex combinations of proper scoring rules are still proper scoring rules and moreover such combinations can induce strictness. It turns out that a continuous, convex combination over the whole simplex where $\boldsymbol{\eta}$ lives, gives a convenient result. That is, we take the expected value of $C^*_{\boldsymbol{\eta}}$, w.r.t. a uniform distribution over the simplex. Performing this expected-value integral[17], we find the closed-form solution:

$$\int_{\Delta} \Gamma(m) C^*_{\boldsymbol{\eta}}(\mathbf{q}|L_i)\, d\boldsymbol{\eta} = -\log q_i = C_{\log}(\mathbf{q}|L_i) \tag{21}$$

where Δ represents the simplex and $\Gamma(m) = 2 \times 3 \times \cdots \times (m-1)$ is the normalization constant of the uniform distribution over the simplex. As promised, this combination has now resulted in an easily computable, strictly proper scoring rule.

In summary, we have constructed here an interpretation of the logarithmic cost function, which shows that it is closely related to misclassification cost. In particular, this interpretation shows that if we design recognizers to minimize logarithmic cost, we can expect those recognizers to also have small misclassification costs.

References

1. Rodriguez-Fuentes, L.J., Varona, A., Diez, M., Penagarikano, M., Bordel, G.: Evaluation of Spoken Language Recognition Technology Using Broadcast Speech: Performance and Challenges. In: Odyssey 2012: The Speaker and Language Recognition Workshop, Singapore (25-28 June 2012)
2. Rodriguez-Fuentes, L.J., Penagarikano, M., Bordel, G., Varona, A.: The Albayzin 2008 Language Recognition Evaluation. In: Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic (28 June - 1 July 2010)
3. Rodriguez-Fuentes, L.J., Penagarikano, M., Varona, A., Diez, M., Bordel, G.: The Albayzin 2010 Language Recognition Evaluation. In: Proceedings of Interspeech, Firenze, Italia (August 28-31 2011) 1529–1532
4. Rodriguez-Fuentes, L.J., Penagarikano, M., Varona, A., Diez, M., Bordel, G.: KALAKA-2: a TV broadcast speech database for the recognition of Iberian languages in clean and noisy environments. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey (23-25 May 2012)

[17] This, however, is *not* easy. See appendix D of [6].

5. Brümmer, N., van Leeuwen, D.A.: On calibration of language recognition scores. In: Proceedings of Odyssey 2006 - The Speaker and Language Recognition Workshop. (2006) 1–8 Available online: http://niko.brummer.googlepages.com.
6. Brümmer, N.: Measuring, refining and calibrating speaker and language information extracted from speech. PhD thesis, Department of Electrical and Electronic Engineering, University of Stellenbosch (December 2010) Available online: https://scholar.sun.ac.za/handle/10019.1/5139.