PPRLM Optimization for Language Identification in Air Traffic Control Tasks

R. Córdoba, G. Prime, J. Macías-Guarasa, J.M. Montero, J. Ferreiros, J.M. Pardo

Speech Technology Group. Dept. of Electronic Engineering. Universidad Politécnica de Madrid
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040-Madrid, Spain
cordoba@die.upm.es
http://www-gth.die.upm.es

Abstract

In this paper, we present the work done in language identification for two air traffic control speech recognizers, one for continuous speech and the other one for a command interface. The system is able to distinguish between Spanish and English. We will confirm the advantage of using PPRLM over PRLM. All previous studies show that PPRLM is the technique with the best performance despite of its drawbacks: more processing time and labeled data is needed. No work has been published regarding the optimum weights which should be given to the language models to optimize the performance of the language recognizer. This paper addresses this topic, providing three different approaches for weight selection in the language model score. We will also see that a trigram language model improves performance. The final results are very good even with very short segments of speech.

1. Introduction

Automatic language identification (LID) has become an important issue in recent years in speech recognition systems. Each day, more and more recognition systems are multilingual and need to know in a very short time the language of the caller to an automatic system to use the appropriate recognition system specific to that language.

To do language identification, first we have to identify which factors are more critical to distinguish between languages. We can identify two main factors of differentiation: the realization of allophones and sounds (some allophones exist in one language but not in other languages) and information related to the sequence of allophones, which has demonstrated to be vital: some sequences of allophones do not exist in one language (or occur very little), so the identification of those sequences is crucial for LID.

Many techniques have been suggested in recent years for this task. The most widespread technique is the phone-based approach, like Parallel phone recognition followed by language modeling (PPRLM) [1], which classifies languages based on the statistical characteristics of the phone sequences and has a very good performance.

Another popular technique is a simple GMM classifier. This technique addresses the first differential factor between languages: every language has sounds that are specific to it. Its main advantage is that we do not need labeled data to train the classifier, so it is a very cheap system. Its main drawback is its low performance, due to the fact that it does not deal with any information regarding the sequence of sounds (the second main factor of differentiation between languages.)

In recent years, some techniques have been proposed that try to take the advantages from both techniques: a GMM classifier called "GMM tokenizer" [2][3][4]. In this approach, the output of the classifier (for each frame, the tokenizer outputs the index of the Gaussian component scoring highest in the GMM computation), is used as input to a "language model" (LM) module, where the sequence of the different indexes is learnt. This technique uses both acoustic information and sequence information, so it seems to be suitable and has the same advantages than the GMM alone: no need for labeled data and is faster that the phone-based approaches. Nevertheless, in all previous studies the performance of this technique is worse than PPRLM, but has one advantage: the combination of PPRLM and this technique improves the overall result. So, it offers complementary information to the task, but with the cost of CPU time due to the use of PPRLM.

In summary, there is a general agreement that PPRLM is the best option if you look for performance and have labeled data available to model the phone recognizers. In fact, it has been widely used for speaker recognition with very good results [5], especially in mismatch conditions.

So, in this paper, we are going to focus on the most promising technique, PPRLM, and we will compare it with PRLM. Also, we will provide some clues to find the best way to optimize the weights used in the language models. This work has been done under the project INVOCA, for the public company AENA, which manages Spanish airports and air navigations systems [6].

The paper is organized as follows. A brief overview of the PPRLM system and the proposals for its optimization is given in section 2. Then, we present the database used in the experiments and the general conditions of the experiments in section 3, followed by the results of the experiments in section 4. The conclusions are given in Section 5.

2. Language Identification Technique

2.1. PPRLM versus PRLM

The main objective of this technique is to model the frequency of occurrence of different phone sequences in each language. These systems have two stages:

1. In the first stage, a phone recognizer takes the speech utterance and outputs the sequence of phonemes corresponding to it.
2. In the second stage, a language model module scores the probability that the sequence of phonemes corresponds to the language.

For the phone recognizer, we have followed the classical approach: context-independent phone models using continuous HMMs with multiple Gaussians.

There are two alternatives:

- Use just one phone recognizer. The advantages are speed, as there is only one recognition process, and we only need labeled data in one language. Its main drawback: it only models the phonemes specific to that language, so its performance is going to be low. It is called PRLM (phone recognition followed by language modeling).
- Use several phone recognizers modeled for different languages. The advantage is that using many recognizers we can cover most of the phonetic realizations of the languages. Its main drawback is speed: processing time is multiplied by the number of recognizers. It is called PPRLM (parallel PRLM).

Using PPRLM, we can even have phone recognizers modeled for languages different than the languages that have to be identified, but obviously if there is a match between the input language and the language of the models the performance will be better, because you can model explicitly the phonetic variations of each language. In our case, as we want to identify English and Spanish and we have labeled data for both of them, the best option is to use PPRLM with phone recognizers trained for English and Spanish. This way, we can model the phonetic differences between them, e.g. in Spanish we only have five vowels, we have the vibrant R which is much stronger than the English one, etc.

2.2. N-gram language modeling

The sequence of phonemes generated by the phone recognizers is used as input to a language model module.

In the training stage, this input is used to model the frequencies of occurrence of n-grams (histograms), one per language. The assumption is that different languages will have different n-gram histograms.

In the recognition stage, interpolated n-gram language models are used to approximate the n-gram distribution as the weighted sum of the probabilities of the n-grams considered. In our case, we have considered up to trigrams. Even though previous studies [1] discouraged the use of trigrams, as we will see, we get better results using them. For a sequence of three symbols we use the formula:

$$S(w_t, w_{t-1}, w_{t-2}) = \alpha_3 \cdot P(w_t \mid w_{t-1}, w_{t-2}) + \alpha_2 \cdot P(w_{t-1} \mid w_{t-2}) + \alpha_1 \cdot P(w_{t-2}) + \alpha_0 \cdot P_0 \quad (1)$$

where $\{w_{t-2}, w_{t-1}, w_t\}$ are the consecutive symbols observed in the phone stream, the P's are ratios of counts observed in the training data, e.g.:

$$P(w_t \mid w_{t-2}, w_{t-1}) = \frac{C(w_{t-2}, w_{t-1}, w_t)}{C(w_{t-2}, w_{t-1})} \quad (2)$$

where $C(w_{t-2}, w_{t-1})$ is the number of times symbol w_{t-2} is followed by symbol w_{t-1}, and $C(w_{t-2}, w_{t-1}, w_t)$ is the number of occurrences of trigram $\{w_{t-2}, w_{t-1}, w_t\}$, both considering all the training text. α_3, α_2, and α_1 are the weights associated to each n-gram. P_0 is a fixed term equal to the inverse of the number of phonemes. α_0 is just a smoothing factor and has a very low value (0.001). As you can see, the formula is just a weighted sum of the contributions of the different n-grams.

The log likelihood that the interpolated trigram language model for language l, λ_l, produced the phone sequence W is:

$$L(W \mid \lambda_l) = -\sum_{t=1}^{T} \log S(w_t, w_{t-1}, w_{t-2} \mid \lambda_l) \quad (3)$$

For language identification, the maximum-likelihood classifier is used, which decides that the language of the unknown utterance is given by the formula:

$$\hat{l} = \arg\min_l L(W \mid \lambda_l) \quad (4)$$

2.3. Techniques for weight selection

We have considered the following alternatives for weight selection in equation (1). Basically, we need to know the relative optimum weight of the three n-grams considered: unigram, bigram and trigram.

2.3.1. Weight selection based in LM observation

The motivation of this technique is to give a bigger weight in equation (1) to the n-gram that provides the biggest difference in score between the Spanish LM and the English LM, considering sentences not used in the LM computation.

For example, with the unigram, we can use the following equations to compute the unigram efficiency for the recognition of Spanish:

$$E_{Spanish}(w_t) = \left(P_{Spanish}(w_t) - P_{English}(w_t)\right) \cdot P_{Spanish}(w_t)$$
$$E^{UNI}_{Spanish} = \sum_i E_{Spanish}(w_i) \quad (5)$$

where $P_{Spanish}(w_t)$ and $P_{English}(w_t)$ are the LM scores of unigram w_t. i represents all sequences of unigrams in the database.

Similar equations can be applied to the bigram and trigram, e.g., for the bigram:

$$E_{Spanish}(w_{t,}, w_{t-1}) = \left(P_{Spanish}(w_t \mid w_{t-1}) - P_{English}(w_t \mid w_{t-1})\right) \cdot P_{Spanish}(w_t, w_{t-1})$$
$$E^{BI}_{Spanish} = \sum_i E_{Spanish}(w_{t,}, w_{t-1}) \quad (6)$$

We repeat this process for the recognition of English and the 3 n-grams and the result is a total of six efficiency values.

This method has two advantages: it helps to decide the weights and it helps to identify which phonemes or sequences of phonemes are really useful in language identification. Setting a threshold, our software shows which phonemes provide the highest differentiation between languages

2.3.2. Weight selection based in likelihood histograms

In this method we compare the histograms with the scores given by the Spanish LM and the English LM for the same validation sentences. We can assume that the bigger the difference between these histograms the better will be the language identification. We can see in the next Figures an example of the histograms.

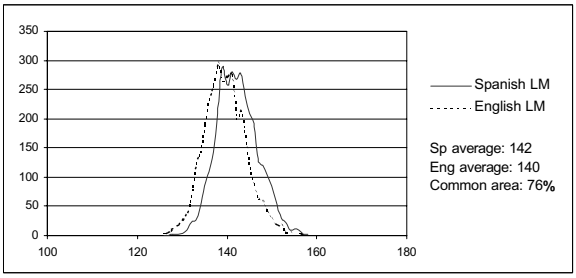

Figure 1: Histogram for unigram LM & Spanish speech

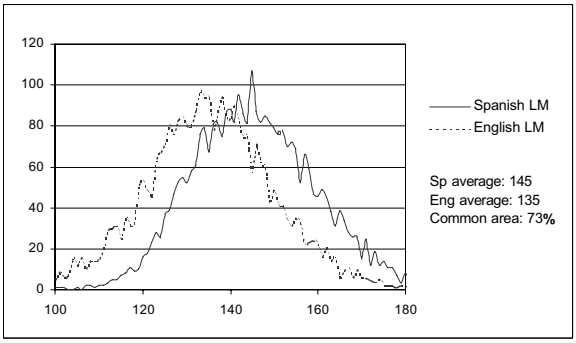

Figure 2: Histogram for trigram LM & Spanish speech

We can see that the discrimination is bigger for the trigram LM, as the common area is smaller. So, we would give a bigger weight to the trigram in this case. We compute these histograms for both languages and all n-grams.

2.3.3. Empirical weight selection

This is just an obvious procedure. We just do an exhaustive weight scanning to find the optimum values. This technique can be extremely costly as three variables have to be found.

3. System Setup

3.1. Databases used

We have used two different databases:

- An isolated speech database, used in a command interface to control the air traffic controller position. In fact, it contains some compound words, which are especially useful, as we need long utterances in our language identification task.
- A continuous speech database, which consists of very spontaneous conversations between controllers and pilots. For speech recognition it is a very difficult task.

We have one big drawback with these databases: all speakers are native Spanish. So, many of them do not reflect all the phonetic variations in English. We will see that this is a decisive factor in all cases for English identification, not for Spanish. In the second database, we have a second drawback: the controllers use to mix Spanish for greetings and goodbyes even when the rest of the sentence is in English. Also, many company names and airports have the Spanish pronunciation embedded in the English conversation.

We have separated each database in three sets:

- A training set, used to generate HMM acoustic models.
- A set dedicated to train the language models.
- A third set dedicated to the validation of all alternatives.

One important aspect of the design is that we need an important part of the data for the second set, because language models are more critical for language identification than acoustic models. In fact, in our first approach we had dedicated a big part of the database to train the acoustic models, but results were low. We decided to take some data from the first set and dedicate it to LM training with a good improvement. It is not important that acoustic models make more mistakes, because language models can 'learn' these mistakes.

In next Tables, we can see the sizes of all databases.

Table 1. Database for isolated speech (words / hours)

	Spanish	English
HMM training set	20,380 / 10.4	11,589 / 5.6
LM training set	10,220 / 5.2	9,097 / 4.4
Validation set	2,919 / 1.5	3,183 / 1.4

Table 2. Database for continuous speech (sentences / h.)

	Spanish	English
HMM training set	4,026 / 7.1	2,200 / 4.7
LM training set	503 / 0.9	500 / 1.0
Validation set	500 / 0.9	453 / 0.9

3.2. General conditions of the experiments

The system uses a front-end with PLP coefficients derived from a mel-scale filter bank (MF-PLP), with 13 coefficients including c_0 and their first and second-order differentials, giving a total of 39 parameters per frame.

For the phone recognizers, we have used context-independent continuous HMM models. For Spanish, we have considered 49 different allophones and, for English, 61 different allophones. So, we have tried to cover all possible phonetic variations in both languages, specially including allophones which do not exist in the other language. All models use 10 Gaussians densities per state per stream.

The results using semicontinuous HMMs were 15% worse than the ones presented in the paper.

4. Experiments and Results

4.1. Command interface (Isolated speech)

4.1.1. Weight selection based in LM observation

With this database the following efficiencies (results from equations 5, 6) are computed:

Table 3. Grammar efficiency for the Spanish PRLM

	Spanish	English
Unigram	2	4
Bigram	32	4
Trigram	75	27
All grammars	109	36

We can see that the efficiency of the Spanish grammars is much bigger, as could be expected, as Spanish sentences are used. The table also shows that trigrams should be given the highest weight. For the English sentences:

Table 4. Grammar efficiency for the English PRLM

	Spanish	English
Unigram	1	2
Bigram	10	12
Trigram	40	52
All grammars	51	66

Now, the table shows that the efficiency of English is only slightly bigger than the Spanish one. This shows the problems we mentioned in section 3.1: the speakers are non-native.

Using this strategy we can confirm, e.g., that the n-grams more relevant to English identification are:

- With the Spanish PRLM: n, R, 'e, 'e-n, f-l-'a, f, k, etc.
- With the English PRLM: f-l-ai, p, f-l, w-'i-n, u-S, etc.

Of course, these phonemes are biased to the air traffic application we are working with.

4.1.2. Weight selection based in likelihood histograms

Histograms for both languages and the 3 n-grams are computed. To abbreviate, we will just present the common area in PPRLM for all of them:

- For Spanish: 82% (unigram), 74% (big.) and 68% (trig.)
- For English: 84% (unigram), 80% (big.) and 73% (trig.)

So, we get better results for Spanish sentences as before, and trigrams provide the best differentiation.

4.1.3. Language identification experiments

Another drawback of this database is that the average duration of the commands is just 0.95 seconds (about 13 phonemes). So, the results may seem bad, but in fact are really good. In Table 5 we can see the best error rates for isolated speech.

Table 5. Best language identification results (isolated)

α_1	α_2	α_3	Input language	Spanish PRLM	English PRLM	Min error	PPRLM
0.05	0.05	0.90	Spanish	15.21	20.59	5.72	12.64
			English	17.99	20.52	6.28	13.92

The average result is **13.3%** using less than one second of speech. As could be expected, PPRLM gives better results than PRLM alone. 'Min error' gives the minimum score that the PPRLM can obtain: items that fail in both PRLMs. Another important conclusion is that all predictions from the techniques proposed are fulfilled: English works slightly worse than Spanish and the best results are obtained when we give the trigram the biggest weight.

We have decided to present these results using all words from the validation set because this way we have more items in the validation set and obtain a better confidence in the comparisons between the different weights tested. Using just the longest commands (in fact, compound words) the error rate keeps going down: with an average of two seconds of speech the error rate is **2.7%**.

4.2. Continuous speech

The results obtained with the continuous speech database for the weight selection techniques presented in this paper are very similar than those obtained for isolated speech. So, we will just present the language identification results. The average duration of the sentences is 4.6 seconds (about 70 phonemes.)

Table 6. Best language identification results (continuous)

α_1	α_2	α_3	Input language	Spanish PRLM	English PRLM	Min error	PPRLM
0.05	0.05	0.90	Spanish	5.47	6.90	1.22	3.85
			English	22.49	9.13	5.12	10.91

The average result is **7.4%**, which is less than we could expect after the good results obtained in the command interface. But the reason is quite obvious: the performance for English is very bad because speakers mix Spanish and English in their greetings and goodbyes. Another factor that can affect the performance of the system is that this database is extremely spontaneous (they are live recordings from the controllers.)

5. Conclusions

We have obtained very good results with our language identification system using a very short period of time. In isolated speech, we have been able to provide good results using less than one second of speech. With continuous speech, the results are worse due to inconsistencies in the database.

We have shown three methods to decide the optimum weights to use in LM scoring, and one of them provides an insight of what phonetic characteristics are really relevant for language identification.

As future work, we plan to apply the same technique to an English continuous speech database with native speakers.

6. References

[1] Zissman, M.A., "Comparison of four approaches to automatic language identification of telephone speech," IEEE Trans. Speech and Audio Processing, vol. 4(1), pp. 31-44, 1996.

[2] Torres-Carrasquillo, P.A., Reynolds, D.A., Deller Jr., J.R., "Language identification using Gaussian mixture model tokenization", IEEE ICASSP 2002, pp. I-757-760.

[3] Torres-Carrasquillo, P.A., Singer, E., Kohler, M.A., Greene, R.J., Reynolds, D.A., Deller Jr., J.R., "Approaches to Language Identification Using Gaussian Mixture Models and Shifted Delta Cepstral Features", ICSLP 2002, pp. 89-92.

[4] Wong, E., Sridharan, S., "Methods to Improve Gaussian Mixture Model Based Language Identification System", ICSLP 2002, pp. 93-96.

[5] Jin, Q., Schultz, T., Waibel, A., "Phonetic Speaker Identification", ICSLP 2002, pp. 1345-1348.

[6] *INVOCA Project Synopses*. Eurocontrol. Analysis of Research & Development in European Programmes. Available at http://www.eurocontrol.int/eatmp/ardep-arda/servlets/SVLT014?Proj=AEN043