A Discriminative Text Categorization Technique for Language Identification built into a PPRLM System

M. A. Caraballo, L. F. D'Haro, R. Cordoba, R. San-Segundo, 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
{macaraballo,lfdharo,cordoba,lapiz,pardo}@die.upm.es

Abstract

In this paper we describe a state-of-the-art language identification system based on a parallel phone recognizer, the same as in PPRLM, but instead of using as phonotactic constraints traditional n-gram language models we use a new language model which is created using a ranking with the most frequent and discriminative n-grams between languages. Then, the distance between the ranking for the input sentence and the ranking for each language is computed, based on the difference in relative positions for each n-gram. The advantage of the proposed ranking is that it is able to model reliably longer span information than in traditional language models and that with less training data it is able to obtain more reliable estimations. In the paper, we describe the modifications that we have made to the original ranking technique, i.e., different discriminative formulas to establish the ranking, variations of the template size and a penalty for out-of-rank n-grams. Results are presented on a new and larger database. The test database has been significantly increased using cross-fold validation for more reliable results.
Index Te rms: Language Identification, n-gram frequency ranking, text categorization, PPRLM

1. Introduction

Currently, one of the most used technique in Language identification (LID) is the phone-based approach, like Parallel phone recognition followed by language modeling (PPRLM)[1]. In PPRLM, the language is classified based on statistical characteristics extracted from the sequence of recognized allophones.

In spite of the high LID accuracy results obtained by PPRLM, the accuracy is reduced due to the presence of bias in the scores generated by each recognizer and because PPRLM does not model correctly long-span dependencies (i.e. to use high order n-gram language models) probably due to an unreliable estimation of the n-gram probabilities. In order to solve the first problem, we decided to use a GMM classifier and a normalization procedure called differential scores. Regarding the second problem, we decided to use a ranking of occurrences of each n-gram with higher n-grams, in a similar way to [2] and [3] where the ranking is applied to written text. Although the information source is very similar to PPRLM (frequency of occurrence of n-grams), results are much better, as we will see.

This paper is a continuation of the work done in [4] and [5] but tested on a new database with more languages and including new modifications to the ranking algorithm. Section 2 describes the system setup and basic techniques. In Section 3 the basic n-gram ranking technique and the new discriminative n-gram ranking are described, together with the results considering all the new alternatives considered. Finally, conclusions and future works are presented in Section 4.

2. System description

2.1. Database

For this work we have used the C-ORAL-ROM database [6], which consists of spontaneous speech for 4 main Romance Languages: Spanish, French, Portuguese, and Italian. This database is made of 772 spoken texts with more than 120 hours of speech and around 300K words for each language. The database transcriptions and annotations were validated by both external and internal reviewers. The database includes recordings in two different types: formal and informal (equally distributed). The formal recordings consist of three different contexts: natural (e.g. political speech, teaching, preaching, etc.), media (e.g. talk shows, news, scientific press, etc), and telephone (e.g. private and human-machine). The informal recordings include monologues, dialogues, and conversations in familiar and public contexts.

Next, we describe the main changes that we made to the database in order to adapt it to our experiments and recognition system: a) Most of the sound files were sampled to 22,050 Hz @ 16 bits and some others to 11 KHz @ 16 bits, all of them were sub-sampled to 8 KHz @ 16 bits in order to use them with the acoustic models of our recognizer. b) Some recordings in the database were too long (i.e. longer than 10 minutes) so they were splitted into shorter files. This way, we also eliminated noised and difficult to recognize sections, c) Finally, we generated random recording lists in order to avoid any kind of bias at training. Table 1 shows the number of sentences in the database that we have finally used. The average sentence length is 6.2 seconds.

	Spanish	**French**	**Italian**	**Portuguese**
Sentences	17634	16474	19074	17946

Table 1: *Number of sentences by language*

2.2. 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 c0 and their first and second-order differentials, giving a total of 3 streams and 39 parameters per frame. We have used two phoneme recognizers, for Spanish and English, with context-independent continuous HMM models. For Spanish, we have considered 49 allophones and, for English, 61 allophones, all with 3 states. All models use 10 Gaussians densities per state per stream.

The performance of phoneme recognizers is very low for several reasons: a) there is a mismatch between the recognizers' languages and the 4 languages to be identified; b) the recordings still contain different kind of noises, background music, etc., and very spontaneous speech; c) the acoustic models were not adapted to this database. So, there is

a clear mismatch in the languages and in the channel conditions. The good thing of using this setup is that improvements obtained with our techniques will be more evident, as we will see.

In order to increase the reliability of the results presented in the next sections, we performed a cross-fold validation, dividing all the available material in 9 subsets: 5 subsets to estimate the LMs, 2 subsets to estimate the Gaussian classifier, 1 subset for development, and 1 subset for test.

2.3. Description of PPRLM

Nowadays, PPRLM is the most popular approach to language identification. The main objective of PPRLM is to model the frequency of occurrence of different allophone sequences in each language. The technique can be divided into two stages. First, several parallel phone recognizers take the speech utterance and outputs a sequence of allophones corresponding to the phone sets used for each one. Second, the sequence of allophones is used as input to a bank of n-gram language models (LM) in order to capture phonotactics information. In this stage, the language model scores the probability that the sequence of allophones corresponds to a given language.

The main advantages of PPRLM are: a) Since it uses many recognizers, it is possible to cover most of the phonetic realizations of every language. b) It is possible to have phone recognizers modeled for languages different to the languages that we have to identify, which is especially useful in situations when the training data is not enough to obtain reliable language dependant models. On the other hand, PPRLM presents two major weaknesses: a) The presence of bias in the log-likelihood scores generated by each combination of the *N* recognizers and *M* language models and, b) the data sparcity and limitations of the n-grams LMs to model long span information.

The bias problem is mainly due to the differences between the allophone dictionaries and training data used by each recognizer [1]. In [7] two solutions for this problem are described. The first solution is called bias removal; it consists on a normalization procedure using as LM score the calculated score minus the average score in the training data. Then, the language is identified using a Maximum Likelihood Classifier. The second solution is to use another kind of classifier, such as Gaussian, K nearest-neighbor, or Support Vector Machine (SVM) classifiers. The advantage of using these classifiers is that the classification is not based on using an absolute discriminant function, and therefore it is not affected by the bias. In our system, given the good results obtained in [8], we decided to continue using a Gaussian Classifier. These classifiers also benefit from applying normalization of the scores (e.g., the T-norm normalization). In our system, we use what we call "differential scores", which is a similar normalization.

Regarding the problems with the LMs, the data sparcity is difficult to solve because it would require new training data (i.e., obtaining new recordings or using an external corpora with the same dictionary of phonemes used in our platform) consisting of a sequence of recognized phonemes. Regarding solutions for the problem of including long span information to the language models, in [9] they describe slight improvements on the LID rate when using the skip-gram technique, and in [3] they present LID experiments on written text for six languages using three different kinds of LM: Markov models, trigram frequency vectors, and n-gram text categorization, with good results for the last technique. In our case, we have used and extended the n-gram text categorization technique [2].

2.4. Gaussian classifier for LID

As mentioned above, the general PPRLM approach has a bias problem in the log-likelihood score for the languages considered. To tackle this issue, we proposed in [10] to use a Gaussian classifier instead of the usual decision formula applied in PPRLM. With all the scores provided by every LM in the PPRLM module we prepare a score vector. With all the sentences in the training database we estimate a Gaussian distribution each language. In recognition, the distance between the input vector of LM scores and the Gaussian distributions for every language is computed, using a diagonal covariance matrix, and the distribution which is closer to the input vector is the one selected as identified language. Besides, the Gaussian classifier allows us to increase the number of Gaussians to better model the distribution that represents our classes.

One important conclusion of our work in [10] is that, instead of absolute values, we need to use differential scores: the difference between the score obtained by one LM and the average score obtained by the other 'competing' languages: (SC_i' = SC_i – Aver(SC_j, j≠i)) in Figure 1. We applied it to unigram, bigram and trigram separately, with 8 scores x 3 n-grams = 24 features in total in the feature vector.

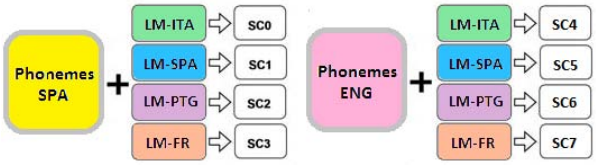

Figure 1: *PPRLM scores used for the LID system*

The average result in LID for PPRLM is **35.8%** error rate. It is a bad result, but, as we mentioned in Sections 2.1 and 2.2, the performance of the acoustic models is really poor and the sentences average length is short.

3. N-Gram Frequency Ranking

In this section we will describe the original text categorization technique and the modifications that we have made to improve it, as well as the selection of the most discriminative n-grams.

3.1. Description of the Basic Technique

In [2], an interesting technique that combines local information (n-grams) and long-span information (collected counts from the whole utterance) is described. In summary, for training the original technique proposes the creation of a ranked template with the N (typically 400) most frequents n-grams (up to n-grams of order five) of the character sequences in the train corpus for each language sorted by occurrence and then orthographically in case two or more n-grams contain the same occurrence (e.g., positions 10 and 11 in Figure 2).

During the evaluation, a dynamic ranked template is created for the phoneme sequence of the recognized sentence following the same procedure. Then a distance measure (OOP, Out-Of-Place) is applied between the input sentence template and each language dependent template previously trained. The distance for a given ranking T is calculated using Eq. 1.

$$d^T = \frac{1}{L}\sum_{i=1}^{L} abs(pos\ w_i - pos\ w_i^T) \qquad \text{Eq. 1}$$

Where *L* is the number of n-grams generated for the input sentence. If an n-gram does not appear in the global ranking (meaning that it has not appeared in training or it is not in the top n-grams selected) it is assigned a maximum distance: the

size of the ranking. The selected language is the one that presents the higher correlation between templates (i.e., the lower distance).

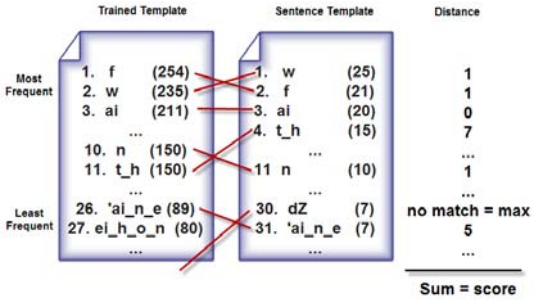

Figure 2: *Example and calculation of distance score using a ranking of n-grams as proposed by [2]*

Figure 2 shows an example of one of the templates created in our system using the English phoneme set and the template created for the unknown sentence.

3.2. Our baseline for N-Gram Ranking

In [5] we described several modifications that we made on the basic technique proposed in [2]. Below we provide a brief description of the most important ones.

Our first variation is what we called the "golf score". As the number of occurrences of the n-grams in the input sentence is very low, most n-grams have the same number of occurrences and should have the same position in the ranking. It is the same as a ranking in golf (the sport): all players with the same number of strokes share the same position. Figure 3 shows an example of the modification applied to the original template using the proposed "golf" score. Using this technique we obtained a 2.5% relative improvement.

For the second modification we thought that having only one global ranking was not efficient since, in general, the top positions were always devoted to unigrams & bigrams, which we already knew that were less discriminative for LID. So, we decided to have different rankings for each n-gram order (besides that, the procedure is the same). As the ranking size for unigram and bigram will be different between languages, we need an additional normalization in the distance measure, i.e., we divide it by the number of items in the set for that n-gram order. We also increased the template size to an optimum of 3000, which is the baseline for this paper.

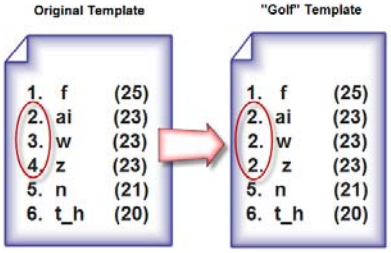

Figure 3*: Ranking template modification with "golf score"*

3.3. N-Gram Discriminative Ranking

Inspired in the work of [11], where better LID results could be obtained using the most discriminative units, we thought that we should introduce the same concept in the ranking creation process; therefore, we decided to give more relevance (higher positions) in the ranking to the items that are actually more specific to the language that is being identified, i.e. n-grams with a high frequency in one language but with zero or low frequency in the competing languages.

In our work we propose a variation of tf-idf. After the original global rankings are created, we have the number of occurrences of each n-gram: $n_1(w)$ = occurrences of n-gram w in the current language, and $n_2(w)$ = the average occurrences of w in the competing languages, where T are the ranking templates created for each language.

$$N_1 = \sum_{\forall w: w \in T_1} n_1(w) \qquad N_2 = \frac{1}{|T-1|} \sum_{\forall w: w \in T; T \neq T_1} n_2(w) \qquad \text{Eq. 2}$$

As the number of total occurrences will be different for each language and n-gram order, before the subtraction a normalization is needed to have comparable amounts. Being N_1 the sum of all occurrences for the current language and N_2 the average for the competing languages (see Eq. 2):

$$n_1'(w) = \frac{n_1(w) \times N_2}{N_1 + N_2} \qquad n_2'(w) = \frac{n_2(w) \times N_1}{N_1 + N_2} \qquad \text{Eq. 3}$$

Another important issue is to use a threshold value for these normalized values (Eq. 3), i.e., to discard the n-grams that were below a threshold as non-representative. In our case, we obtained an optimum using 9-9-3-3-2 (threshold values for each n-gram starting at unigram). Then, we considered several alternative formulas with the same philosophy as tf-idf for the final number of occurrences used to assign the final position in the ranking (which we will call n_1'').

1	$n_1'' = (n_1' - n_2') / (n_1' + n_2')$
2	$n_1'' = n_1' * (n_1' - n_2') / (n_1' + n_2')$
3	$n_1'' = \log(n_1') * (n_1' - n_2') / (n_1' + n_2')$
4	$n_1'' = \text{sqrt}(n_1') * (n_1' - n_2') / (n_1' + n_2')$
5	$n_1'' = n_1' * (n_1' - n_2') / (n_1' + n_2')^2$
6	$n_1'' = \text{abs}(n_2' - n_1) / \text{sum all lang}(n_1)$

In Table 2 we can see the results in LID error rates for the 6 different formulas considered. We present the results for each n-gram order alone and, in the final column, the result for the fusion of all n-gram classifiers. This way, we can see the relevance of each n-gram alone. The first line is our baseline experiment: no discriminative ranking, "golf score", independent templates for each n-gram with 3,000 units.

Formula	1-gram	2-gram	3-gram	4-gram	5-gram	All
No discrim.	74.4	43.1	38.7	44.3	58.6	34.40
1	52.6	37.6	32.9	34.4	49.6	**24.93**
2	53.8	40.6	35.8	39.2	56.9	32.93
3	53.4	38.6	32.4	35.3	54.8	29.10
4	53.4	39.7	34.2	35.9	56.1	30.38
5	**52.6**	**37.6**	**32.8**	**34.4**	**49.6**	**24.91**
6	52.7	39.3	32.8	34.4	49.4	25.23

Table 2: *Error rates for the different formulas.*

We can see that the discriminative ranking means an outstanding improvement (from 34.40 to 24.91, 27.6% relative improvement) and without it, results are similar to PPLRM, although slightly better (34.4 vs. 35.8%). We also observe in the table that, as could be expected, the trigram is the most powerful classifier. But what is extremely interesting is that the 4-gram is very close in performance, so it is a clear advantage over PPRLM, where obtaining reliable estimates for 4-gram is difficult and requires a huge training database.

The best result corresponds to the formula 5. Its advantage is that it normalizes the values between 1 and -1: 1 means that the n-gram appears in the current language but not in the other competing ones ($n_2'=0$), indicating that it is especially relevant for that language; -1 means just the opposite ($n_1'=0$), so the n-gram does not appear in the current language.

3.4. Influence of the template size

In Table 3 we can see the effect of the template size for the best configuration (formula 5 in previous section). So, the best results correspond to a template size equal to 5000, although it begins to saturate, and the improvements are only obvious in 4-gram and 5-gram, which could be expected because is where more units are left out the template. So, an alternative that we are considering is to have different sizes for each n-gram. Obviously, this optimum will depend in the number of allophones in each language, so some fine tuning will be needed for another setup.

Template size	1-gram	2-gram	3-gram	4-gram	5-gram	All
500	53.6	40.0	52.6	57.9	66.7	36.70
1000	53.0	38.8	44.8	47.5	61.0	31.62
2000	52.8	37.9	36.5	39.5	53.6	27.13
3000	52.7	37.6	32.8	34.4	49.6	24.91
4000	52.7	37.5	32.8	**34.0**	48.2	24.70
5000	52.7	37.5	32.8	34.0	**48.0**	**24.68**

Table 3: *Error rates for different template sizes.*

3.5. Influence of out-of-rank n-grams

One issue that we have to take into account is that for high order n-grams the amount of out-of-rank units increases. Our first approach was to assign these units the last position in the template (the template size). But it is clear that some penalty can be applied for those cases, so we decided to multiply the last position by a factor greater than 1 for out-of-rank units. In Table 4 we can see the results. The baseline uses 3,000 units.

We can see that there is an optimum for the penalty 1.7, with improvements from 3-gram to 5-gram, as could be expected (unigram and bigram have almost no out-of-rank units). Obviously, improvements saturate for large penalties. Another interesting result is that this penalty is more effective than just increasing the template size (which could be an alternative): in 3-gram, 31.8 (1.7 penalty) vs. 32.8 (5,000 units) in Table 3. And we obtain similar gains for 4-gram and 5-gram (slightly less). The probable reason is that just increasing the template size includes very unreliable n-grams, especially for trigram.

Penalty factor	1-gram	2-gram	3-gram	4-gram	5-gram	All
1.0 (base)	52.6	37.6	32.8	34.4	49.6	24.91
1.35	52.6	37.6	31.8	33.7	48.7	24.57
1.7	52.6	37.5	31.8	**33.4**	48.3	**24.47**
2.0	52.6	37.5	31.8	33.5	48.1	24.48
2.5	52.6	37.4	31.9	33.5	**47.9**	24.51
3.0	52.6	37.4	31.9	33.5	47.9	24.57

Table 4: *Error rates for penalties for out-of-rank units.*

3.6. Stratified rankings

After examining the rankings obtained, we considered the possibility of grouping n-grams with close values in n_1'' value considered for the ranking, so that we "smooth" the ranking.

Total units	Units/ cluster	1-gram	2-gram	3-gram	4-gram	5-gram	All
3000	1 (base)	52.6	37.6	32.8	34.4	49.6	24.91
3000	2	52.6	37.5	31.8	33.4	48.0	**24.54**
3000	3	52.6	37.5	31.9	33.5	47.8	24.55
3000	4	52.6	37.5	32.0	33.5	47.8	24.62
4000	2	52.6	37.5	31.8	33.5	47.6	24.62
6000	2	52.6	37.5	31.8	33.5	47.5	24.62

Table 5: *Error rates for penalties for out-of-rank units.*

In Table 5 we can see the results for different template sizes and number of units in each cluster. E.g. last row means a 3000 cluster template size with 2 units/cluster. Again we can see some improvements, especially for 4-gram and 5-gram.

4. Conclusions and Future Work

We have demonstrated that the n-gram Frequency Ranking approach overcomes PPRLM thanks to the longer span that can be modeled, especially for the great effect of the 4-gram, and partially of the 5-gram. To obtain this improvement, the following issues have been crucial:

- n-gram discriminative rankings with the normalized value for the number of occurrences are able to overcome PPRLM (31.6% relative improvement, 24.47 vs. 35.8).
- The ranking size should be between 3,000 and 5,000 depending on the n-gram order.
- Applying a penalty to out-of-rank n-grams may provide up to 1.7% relative improvement.
- Similar gains can be obtained with the stratified rankings.

As future work, we will consider different template sizes and penalties for the different n-grams to achieve the best result possible in the fusion of all of them.

5. Acknowledgements

This work has been supported by SD-TEAM (TIN2008-06856-C05-03), ROBONAUTA (DPI2007-66846-c02-02), and MA2VICMR (S2009/TIC-1542).

6. References

[1] Zissman, M.A., "Comparison of four approaches to automatic language identification of telephone speech," IEEE Trans. Speech&Audio Proc., v. 4, pp. 31-44, 1996.

[2] Cavnar, W. B. and Trenkle, J. M., "N-Gram-Based Text Categorization", Proc. 3rd Symposium on Document Analysis & Information Retrieval, pp. 161-175, 1994.

[3] Vatanen, t., Väyrynen, J. and Virpioja, S. "Language Identification of Short Text Segments with N-gram Models". Int. Conf. on. Language Resources and Evaluation (LREC'10), 2010.

[4] Cordoba, R., D'Haro, L.F., et al. "n-gram Frequency Ranking with additional sources of information in a multiple-Gaussian classifier for Language Identification". V Jornadas de Tecnología del Habla, pp. 49-52, 2008. Bilbao, Spain.

[5] Cordoba, R., D'Haro, L.F., et al. "Language Identification based on n-gram Frequency Ranking". Interspeech 2007, pp. 354- 357.

[6] Cresti, E. et al. "The C-ORAL-ROM CORPUS. A Multilingual Resource of Spontaneous Speech for Romance Languages". IV Int. Conf. on Language Resources and Evaluation, 2004.

[7] Ramasubramaniam, V., Sai Jayram, A. K. V., and Sreenivas, T. V. "Language Identification using Parallel Phone Recognition. Workshop on Spoken Language Processing". pp. 109-116. 2003.

[8] Cordoba, et al. "Cross-Task and Speaker Adaptation in a Speech Recognition System for Air Traffic Control". IEEE Aerospace and Electronic Systems Magazine, Vol. 21, No 9, pp. 12-17. 2006.

[9] Navratil, and J. Zühlke, W. "Double bigram-decoding in phonotactic language identification". ICASSP, Vol. 2, pp. 1115–1118. 1997.

[10] Córdoba, R., et al. "Integration of acoustic information and PPRLM scores in a multiple-Gaussian classifier for Language Identification". IEEE Odyssey 2006.

[11] Nagarajan, T., and Murthy, H. A. "Language Identification Using Parallel Syllable-Like Unit Recognition". ICASSP, pp. I-401-404. 2004.