# PRUNED UNIVERSAL SYMBOL SEQUENCES FOR LZW BASED LANGUAGE IDENTIFICATION

*S.V. Basavaraja*[*] *and T.V. Sreenivas*[†]

[*]Applied Research Group, Satyam Computer Service Limited, Bangalore.
[†]Dept of ECE, Indian Institute of Science, Bangalore, India.
Email: basavaraja_sv@satyam.com, tvsree@ece.iisc.ernet.in

## ABSTRACT

We present a improved language modeling technique for Lempel-Ziv-Welch (LZW) based LID scheme. The previous approach to LID using LZW algorithm prepares the language pattern table using LZW algorithm. Because of the sequential nature of the LZW algorithm, several language specific patterns of the language were missing in the pattern table. To overcome this, we build a universal pattern table, which contains all patterns of different length. For each language it's corresponding language specific pattern table is constructed by retaining the patterns of the universal table whose frequency of appearance in the training data is above the threshold. This approach reduces the classification score (Compression Ratio [LZW-CR] or the weighted discriminant score [LZW-WDS]) for non native languages and increases the LID performance considerably.

**Index Terms :** Language modeling, PRLM, Pattern table, LZW-CR, LZW-WDS.

## 1. INTRODUCTION

Usually LID is performed by tokenizing the input signal first followed by building a language models for these tokens. The common tokenization for spoken language identification is that of phonemes of one or more languages [1]. The language models are often stochastic models viz., unigram, bigram distributions [1][2] ergodic-HMM [3][4], duration models, etc. In [5] two approaches are proposed for language modeling one is modified bigrams with context-mapping matrix and another one is language model based on binary decision trees. Various architectures of LID have been proposed [2], viz. (i) PRLM (phone recognition followed by language model), (ii) PPRLM (parallel PRLM), (iii) PPR (parallel phone recognition). In this paper, we use the PRLM architecture because of its simplicity.

Stochastic models proposed for LID are typically Markov models of small orders (unigram, bigram, trigram etc.). The language discriminability for finite test data is limited by the nature of the model itself, with higher order models likely to do better; but, higher order model estimates are poorer with limited training data.

The LZW based language model for LID proposed in [6] attempts to solve the above limitations by keeping only limited numbers of patterns of different string length for each language. These patterns are automatically derived from the training data of a language, which is assumed to be generalizable to the unseen test data. Once the pattern table is built, for a given test sequence, a compression ratio (LZW-CR) or weighted discriminant score (LZW-WDS) is computed [6] and the highest score decides the language-ID.

The LZW technique ([8]-[10]) allows the basic structural unit of the language to be of variable length. So the technique captures the advantages of higher order models with much less training data. Also the LZW technique, builds the pattern tables regardless of the frequency of occurrence of the patterns. A pattern $T_i$ occurring most in language $L1$ will also be present in the pattern table of language $L2$ even if it occurs only once in the training data of language $L2$. As a result, more than 50% of the patterns in each pattern table are not language specific; these add confusability to the LID task, thereby inhibiting the performance.

In [7] we propose two solutions for overcoming this limitation. It builds LZW pattern tables as before but then make these pattern tables more language specific by pruning it (Fig. 1). Two pruning techniques were discussed. First one is language specific (LS-LZW) pruning. Here, for each language pattern table, only those patterns which are unique to that pattern table are retained; i.e. a pattern $T_i$ present in language $L1$ pattern table is retained only if $T_i$ doesn't appear in any of the other pattern tables. Second one is length frequency (LF-LZW) pruning. Here a pattern in the pattern table is retained only if the product of its frequency of occurrence in that language training data and its length is more than a threshold. After pruning both of these techniques use the same LZW-CR and LZW-WDS method of scoring for identifying the language of a test sequence. By pruning the pattern tables it was shown that the performance of LID has improved along with a reduction in complexity.

Although the LZW based language modeling has shown promise for LID, for very short test sequences and limited

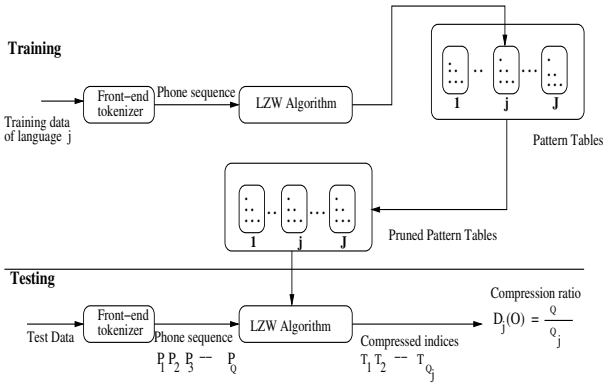


**Fig. 1**. *Block Schematic showing the Training and Testing Phase of LS-LZW-CR based LID*

training data, there is scope for improvement. In spite of using variable length token patterns, we recognize that the pattern tables need to be further enriched. We address this problem in this paper and propose a method of forming universal string or pattern table of a certain maximum length and then pruning it to form language specific pattern table for each language.

## 2. PATTERN TABLE BUILDING

### 2.1. Training Stage

In training, we do not use LZW algorithm to build pattern tables corresponding to each language. First an universal table is constructed by keeping all the patterns of different size. Consider the total number of phonemes or tokens is $N$, i.e. $p_1, p_2, ...p_N$. The universal table $U$ is constructed by the combination of these phonemes. We call this combination of phonemes as pattern $T_i$. where $1 \leq i \leq M$, $M$ is the maximum number of patterns in $U$. If the size of the pattern is varied from 1 to $K$, then

$$U = C_1 \cup C_2 \cup C_3 \cup ....C_{K-1} \cup C_K \tag{1}$$

where $C_i$ is the set of all symbol sequences of length $i$; thus the total number of patterns in $U$ is $M$=$N + N^2 + ... + N^K$.

Since the universal table is huge and not language specific it does not help LID. We construct a language specific pattern table $S_j$ for language $j$ ($1 \leq j \leq J$, where $J$ is the total number of languages) from $U$ by pruning. We assign a weight factor to each pattern $T_i$ in $U$; the weight factor is the number of times the pattern $T_i$ has appeared in the training data of language $j$. Now $S_j$ will contain pattern $T_i$ of $U$ only, if the weight factor is more than a fixed threshold. Pruning is repeated using the respective language training data. We call this type of building a language models as pruned universal symbol (PUS) language models.

By comparing these language specific pattern tables with the pattern tables obtained by LZW [6][7], we see that several additional patterns are included because of the substrings of the universal set.

### 2.2. Testing

Using the language specific pattern tables from the training stage, we obtain a language identification score by applying LZW algorithm for a test sequence; two such scores have been identified [6].

#### *2.2.1. Compression Ratio (CR)*

For the test sequence, each newly found pattern is coded by its index in the pattern table. Since the test sequence of patterns is represented by a sequence of indices, the algorithm achieves compression. The test sequence is separately compressed using the pattern table of each language. For the given test sequence, if the pattern table is representative of its language and if the test sequence contains patterns unique to that language, the compression ratio will be high. Conversely, if the phoneme sequence does not correspond to the language of the pattern table, the phonemes get coded individually resulting in a low compression ratio. We define the compression ratio as the ratio of the number of phonemes in the test sequence to the number of indices obtained after LZW compression.

#### *2.2.2. Weighted Discriminant Score (WDS)*

In WDS, a weight factor is assigned to each pattern in $S_j$ using the training data of language $j$. For a pattern $T_i$ of length $s$, the weight factor $L_j(T_i)$ for language $j$ is calculated as:

$$L_j(T_i) = \frac{N_{T_i}}{N_s} \tag{2}$$

where $N_{T_i}$ denotes the number of times the pattern $T_i$ occurred in the training data and $N_s$ denotes the total number of patterns of length $s$ in the training data. The weight factors are normalized as below:

$$W_j(T_i) = \frac{L_j(T_i)}{\sum_{k=1}^{N_j} L_j(T_k)} \tag{3}$$

such that

$$\sum_{i=1}^{N_j} W_j(T_i) = 1 \tag{4}$$

where $N_j$ denotes the number of a patterns in $S_j$.

For a test sequence $O$ its discriminant score $D_j(O)$ for a language $j$ is calculated as follows. The test sequence $O$ is converted into a sequence of patterns by using pattern table of language $j$. Let $T_i$, $i = 1, ..., Q_j$ denote these patterns. Now the discriminant score of $O$ for language $j$ is defined as the product of the weight factors of the individual patterns.

$$D_j(O) = \prod_{i=1}^{Q_j} W_j(T_i) \quad (5)$$

The assumption here is that, the individual patterns are independent. If the patterns are not independent they would not have occurred separately in the pattern table. i.e. if a pattern $T_k$ and $T_l$, $k, l = 1..Q_j$ are not independent of each other, then $T_k$ and $T_l$ would not be separately present but the concatenated pattern $T_kT_l$ would be present in the pattern table. This assumption will hold good when the training data for building the pattern tables contain most of the valid patterns occurring in the language.

From the set of languages $J$, the language ID for the test sequence is $j^*$

$$j^* = \arg\max_{j \in J}(D_j(O)) \quad (6)$$

## 3. EXPERIMENTS AND RESULTS

The experiments for the LID task are performed on the 6 language OGI-TS data base, which contains manually labeled phonetic transcriptions. The 6 languages are : English, German, Hindi, Japanese, Mandarin and Spanish. The manual labels are subjected to simulated errors, using a random error model. The OGI-TS database uses transcription based on the multi-language motivated Worldbet [11]. The transcriptions of the *story-bt* sentences of the OGI-TS database uses 923 symbols in all, from the 6 languages. The phonetic detail is made explicit by use of diacritics. The diacritics are merged into the base labels leaving us with approximately 150 symbols. By grouping together similar sounding phonemes, this is further reduced to 50 language-independent phonetic units. The resulting 50 units, which include several silence and non-speech units, are shown in Table 1.

**Table 1**. *50 size Phone Inventory including non-speech symbols.*

| | |
|---|---|
| vowel(14) | i, 3r, I, u, E, >, @, &, o, a, 8, e, 2, ax |
| semivowels(4) | w, l, r, j |
| diphthongs(8) | ai, ei, ou, au, iu, Eax, oi, uax |
| nasals(3) | m, n, N |
| fricative(9) | f, s, sh, v, z, h, D, G, T |
| affricate(3) | dZ, ts, cC |
| stops(6) | b, d, g, k, p, t |
| non speech(3) | pause, line,breath,smacking noise, other noises |

Training data of each language consists of $20,000$ phonemes. Size of the universal table $U$ is $M$=$63,77,550$ and maximum length of the pattern is $K$=4. After pruning, the average size of the language specific pattern table $S_j$ is $1,650$. We have used threshold for pruning as 4; i.e. pattern should appear in training data at-least 5 times.

Now, we present the results of spoken language identification on the 6 languages of the OGI-TS database. Each *story-bt* utterance is at least 45 sec long and is spoken by a unique speaker. We divide the utterances of each language into two parts, training speakers and testing speakers (mutually exclusive). The *story-bt* being extempore and free, makes the LID task text independent and speaker independent. To simulate the real tokenization we introduce a controlled amount of token errors to manually assigned phonetic labels. Noisy tokenization is realized by first generating one random variable for each token. This random variable takes on values 1 and 0 with probabilities $p$ and $1-p$ respectively, where $p$ is the induced artificial error rate. For each token, if the value of the corresponding random variable is 1, then that token is replaced by any one of the other tokens (all with equal probability). On the other hand, if the value of this random variable is 0, then that token is left unmodified. Thus, after this process we get a noisy tokenization of the speech utterance with an error rate of $p$. Noise is added to the training tokens as well as to the test phonemes. We have generated token sequences with 30% error (corresponding to typical phoneme error rates of an automated front end) to test the language model performance.

The performance of LID system with random substitution error presents the lower bound of the LID system accuracy, as the random substitution errors does not maintain any particular form of patterns. In case of phonetic similarity based substitution errors there is more chance that some kind of a pattern is maintained, like phoneme "k" is always misplaced with "g". It is possible to have a good LID accuracy with a wrong but consistent phonetic recognizer, because the LID system requires only consistent labeling, e.g., pattern "aka" is always recognized as "aga". Since we train the system with "aga" itself, so LID system considers "aga" as one of the pattern in that language. If the pattern "aka" appears in test data, but phonetic recognizer will convert it into "aga".

Test utterances have lengths varying from 20 to 300 phonemes. LID performance of the proposed PUS language modeling technique using compression ratio (CR) or using weighted discriminant score (WDS) are compared with previous techniques namely Bigram, basic LZW with compression ratio (LZW-CR) and with weighted discriminant score (LZW-WDS), Language Specific LZW with compression ratio (LS-LZW-CR) and with weighted discriminant score (LS-LZW-WDS) and Length Frequency product based LZW with compression ratio (LF-LZW-CR) and with weighted discriminant score (LF-LZW-WDS), the LID task has been performed and the results averaged over the 6 languages for an error probability $p = 0.3$ is reported in Table 2 and 3 . The graphical illustration of the average LID performance for $p$ = 0.3 is also shown in Fig. 2 (CR as the measure) and Fig. 3 (WDS as the

**Table 2**. *Average LID accuracy for $p = 0.3$ (30% tokenization noise)using compression ratio as distance measure. Method 1: Bigram, Method 2: LZW-CR, Method 3: LS-LZW-CR, Method 4: LF-LZW-CR, Method 5: PUS-CR*

| Test size | Method 1 | Method 2 | Method 3 | Method 4 | Method 5 |
|---|---|---|---|---|---|
| 20 | 66.05 | 38.5 | 49.15 | 46.73 | 50 |
| 40 | 66.04 | 60.78 | 70.27 | 68.53 | 71.54 |
| 60 | 66.49 | 71.41 | 80.73 | 79.64 | 81.10 |
| 80 | 71.89 | 80.19 | 87.60 | 86.09 | 87.91 |
| 100 | 73.78 | 86.73 | 91.26 | 90.42 | 91.08 |
| 150 | 78.48 | 92.00 | 97.06 | 95.91 | 95.63 |
| 200 | 81.20 | 95.59 | 98.09 | 97.44 | 97.35 |
| 250 | 82.30 | 97.19 | 99.28 | 98.89 | 98.97 |
| 300 | 86.35 | 98.48 | 99.26 | 99.26 | 99.92 |

**Table 3**. *Average LID accuracy for $p = 0.3$ (30% tokenization noise)using WDS as distance measure. Method 1: Bigram, Method 2: LZW-WDS, Method 3: LS-LZW-WDS, Method 4: LF-LZW-WDS, Method 5: PUS-WDS*

| Test size | Method 1 | Method 2 | Method 3 | Method 4 | Method 5 |
|---|---|---|---|---|---|
| 20 | 66.05 | 76.40 | 77.32 | 78.84 | 80.39 |
| 40 | 66.04 | 89.34 | 90.39 | 90.78 | 91.78 |
| 60 | 66.49 | 94.5 | 95.42 | 95.01 | 96.69 |
| 80 | 71.89 | 97.00 | 97.67 | 97.56 | 98.23 |
| 100 | 73.78 | 98.60 | 98.23 | 98.62 | 98.98 |
| 150 | 78.48 | 99.53 | 99.53 | 99.39 | 99.63 |
| 200 | 81.20 | 99.71 | 99.60 | 99.71 | 99.80 |
| 250 | 82.30 | 99.64 | 99.86 | 100 | 100 |
| 300 | 86.35 | 100 | 100 | 100 | 100 |

measure).

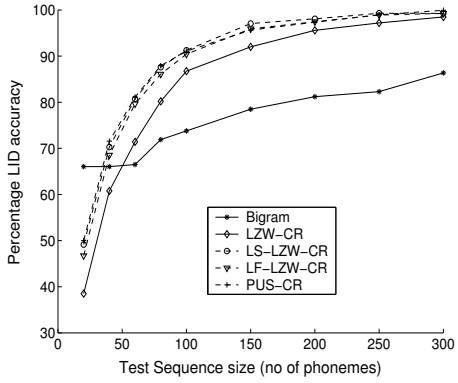


**Fig. 2**. *Average LID accuracy for $p = 0.3$ using Compression Ratio (CR) as a measure.*

From the tables, it is clear that the new language model consistently improves the performance for all cases; error rate reduction of around 10% is seen for small test size cases. Also, the WDS out performs CR in all test cases and all language models. However, we still need about 10 Sec of speech for around 99% LID performance.

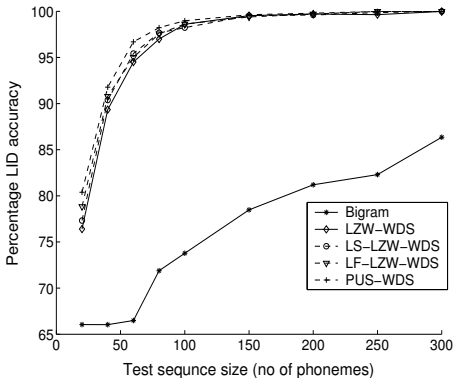


**Fig. 3**. *Average LID accuracy for $p = 0.3$ using Weighted Discriminant Score (WDS) as a measure.*

In order to verify the performance of the proposed LID system for the errors introduced by segmentation and phone recognizer. We have introduced 10%deletion error by deleting the manually labeled segments randomly, 10% of insertion errors are introduced by inserting a random label between two manually labeled segments and 30% phone recognizer error is introduced as discussed above. By comparing the results of Table 4 and 5 with only noisy tokenization (only phone recognition error, Table 2 and 3 ), we can see that there is overall deterioration of all techniques, particularly for short test sequences. However, among the three language models, we can see that LZW-WDS is significantly more robust to tokenization, deletion and insertion errors. The traditional bigram language model is much poorer than both the LZW based models. We can see that LZW-CR approach is quite poor for short test sequences.

**Table 4**. *Average LID accuracy for 10% deletion error, 10% insertion and 30% tokenization noise using compression ratio as discriminative measure. Method 1: Bigram, Method 2: LZW-CR, Method 3: LS-LZW-CR, Method 4: LF-LZW-CR, Method 5: PUS-CR*

| Test size | Method 1 | Method 2 | Method 3 | Method 4 | Method 5 |
|---|---|---|---|---|---|
| 20 | 62.27 | 27.78 | 39.23 | 33.87 | 36.40 |
| 40 | 63.24 | 46.42 | 59.54 | 55.65 | 57.08 |
| 60 | 64.59 | 59.20 | 71.01 | 68.11 | 68.23 |
| 80 | 67.62 | 67.94 | 77.79 | 75.37 | 75.77 |
| 100 | 69.99 | 73.62 | 82.97 | 79.74 | 80.22 |
| 150 | 77.53 | 82.60 | 90.08 | 87.54 | 88.59 |
| 200 | 80.42 | 87.70 | 92.50 | 92.73 | 91.55 |
| 250 | 84.14 | 90.93 | 94.48 | 94.11 | 94.60 |
| 300 | 86.29 | 93.74 | 96.79 | 95.97 | 94.62 |

**Table 5**. *Average LID accuracy for 10% deletion error, 10% insertion and 30% tokenization noise using WDS as discriminative measure. Method 1: Bigram, Method 2: LZW-WDS, Method 3: LS-LZW-WDS, Method 4: LF-LZW-WDS, Method 5: PUS-WDS*

| Test size | Method 1 | Method 2 | Method 3 | Method 4 | Method 5 |
|---|---|---|---|---|---|
| 20 | 62.27 | 68.18 | 69.46 | 71.31 | 73.72 |
| 40 | 63.24 | 82.99 | 84.34 | 85.46 | 86.47 |
| 60 | 64.59 | 89.97 | 90.80 | 91.12 | 92.24 |
| 80 | 67.62 | 93.41 | 93.84 | 95.01 | 95.86 |
| 100 | 69.99 | 95.98 | 95.53 | 97.01 | 97.70 |
| 150 | 77.53 | 98.39 | 98.29 | 98.89 | 98.40 |
| 200 | 80.42 | 99.37 | 98.88 | 99.89 | 99.71 |
| 250 | 84.14 | 100 | 99.64 | 100 | 100 |
| 300 | 86.29 | 100 | 100 | 100 | 100 |

## 4. CONCLUSIONS

We propose a new technique for building a language specific pattern table, by pruning a universal table of token sequences of different size. The Language specific pattern table is constructed by assigning a weight to each pattern of the universal table and then retaining the patterns whose weight factor is above the threshold. We thus maintain the good language modeling capability by retaining the patterns which are important for LID. So, the proposed PUS technique is able to build more discriminative language models for LID.

## 5. REFERENCES


[1] M.A. Zissman, "Language identification using phoneme recognition and phonotactic language modeling", Proc. ICASSP, pp-3503-3506, April 1995.

[2] M.A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech", IEEE Trans Speech and Audio Processing, Vol. 4, No. 1, pp-31-44, Janauary 1996.

[3] V. Ramasubramanian, A.K.V. SaiJayaram and T.V. Sreenivas, "Language identification using parallel subword recognition - an ergodic HMM equivalence", Proc. Eurospeech, Geneva, pp-1357-1360, September 2003.

[4] S.A.Santosh Kumar and V. Ramasubramanian, "Automatic language identification using ergodic-HMM", Proc. ICASSP, pp-I-609-I-612, April 2005.

[5] Navratil, J. and W. Zuhlke, "Phonetic-context mapping in language identification", Proc. EUROSPEECH, vol. 1, Greece, pp-71-74, September 1997.

[6] S.V. Basavaraja and T.V. Sreenivas, "LZW Based Distance Measures for Spoken Language Identification", Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, Puerto Rico, pp. 1-6, June 2006.

[7] S.V. Basavaraja and T.V. Sreenivas, "Low Complexity LID using Pruned Pattern tables of LZW", Proc. Interspeech 2006 - ICSLP, Pittsburgh, pp. 413-416, September 2006.

[8] Welch, T A, " A technique for high-performance data compression", IEEE Computer. Vol. 17, pp. 8-19. June 1984.

[9] Jacob Ziv and Abraham Lempel, "Compression of individual sequences via variable-rate coding", IEEE Transactions on Information Theory, Vol. IT-24, No. 5, pp. 530-536, September 1978.

[10] Mark Nelson, "LZW Data Compression", Dr. Dobb's Journal, October 1989.

[11] T. Lander, "The CSLU labeling guide", Center For Spoken Language Understanding, Oregon Graduate Institute, May 1997.