EVALUATION OF CONFIDENCE MEASURES FOR LANGUAGE IDENTIFICATION

Kay Berkling, Doug Reynolds, Marc Zissman *

M.I.T. Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9185, USA
kay,dar,maz@sst.ll.mit.edu

Abstract

In this paper we examine various ways to derive confidence measures for a language identification system [3], using phone recognition followed by language models, and describe the application of an evaluation metric [1] for measuring the “goodness” of the different confidence measures. Experiments are conducted on the 1996 NIST Language Identification Evaluation corpus (derived from the Callfriend corpus of conversational telephone speech). The system is trained on the NIST 96 development data and evaluated on the NIST 96 evaluation data. Results indicate that we are able to predict the performance of a system and quantitatively evaluate how well the prediction holds on new data.

1. INTRODUCTION

In the development of a typical language identification system (LID), the error rate is measured on evaluation data, characterizing the system by a single number. However, during operation, it can be very useful to have a finer-grain measure of system performance, such as a confidence measure associated with each decision. The confidence measure is defined as a number between 0 and 1, reflecting the probability of correct system performance on a given test token. Derivation of a meaningful confidence measure is achieved in two steps. After training any basic recognition system (e.g., speech, language, speaker recognition), a confidence measure is derived from a separate training set, taking into account the performance of the system. Different methods for constructing confidence measures are possible, including the types of features extracted from the system and their usage. While speech recognition systems have a variety of word level features to extract and apply for this purpose, the language identification system is limited to phoneme level features in this paper. Our primary focus is using the output scores from the LID system to compute confidence scores. In addition, several phoneme-based features, such as durations and frequency of occurrence, are extracted in order to study their contribution to the “goodness” of the confidence measure.

This paper is organized as follows: In Section 2 we provide a brief review the LID system used in this work. We describe the various methods for deriving confidence scores from selected features in Section 3 and evaluate the methods using a confidence evaluation measure in Section 4. In Section 5 we evaluate the utility of additional features in the derivation of a confidence score.

2. LID SYSTEM

The language identification system used in this study is called the PRLM-P system (phone recognition followed by language models) [3]. Figure 1 shows a block diagram of the language identification system for a two-language, single-phone recognizer case. In this approach, incoming speech is first tokenized in terms of phones from a given language via a phone recognizer. During training, an interpolated language model of unigrams \(u()\) and bigrams \(b()\) is derived for a language to be recognized using phone sequences from training speech in that language. During recognition, the score of a test utterance \(U\), tokenized as a sequence of \(T\) phones, \(p_{h1}, p_{h2}, \ldots, p_{hT}\), against a language model \(L\) is calculated as

\[
l(U|L) = \frac{1}{T} \prod_{t=2}^{T} \lambda_2 \cdot b(p_{hat}|p_{hat-1}, L) + \lambda_1 u(p_{hat}|L) + \lambda_0
\]

(1)

In general, multiple phone recognizers (frontends) can be used to tokenize the speech, requiring a set of language models for each language to be recognized (one set for each phone recognizer). Thus, for each input utterance, we have a \(NL \times PR\) vector of language model scores for all \(NL\) languages to be recognized coming from each of the \(PR\) phone recognizers. This vector is then processed by \(NL\) Gaussian classifiers (backend), trained on development data using cross-validation. These \(NL\) scores from the backend are further normalized as

\[
s(U|L_j) = \frac{l(U|L_j)}{\sum_i^n l(U|L_i)}
\]

(2)

Finally, the utterance is classified as being in language \(L_{win}\) with the corresponding score \(s = s(U|L_{win})\), where

\[
L_{win} = \arg\max_j s(U|L_j).
\]


---

*This work was sponsored by the Department of Defense under Air Force Contract F19628-95-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Air Force.
The confidence score generator is a separate path which takes in inputs from various points in the LID system to synthesize a meaningful confidence score for each decision. In the next section we describe three poolings of LID scores into sets are used to derive the confidence scores.

3. CONFIDENCE SCORE DERIVATION

As stated earlier, the confidence score should reflect the probability of correctness of the system's decision. In this section, we examine the straightforward approach of using estimated probability of correctness given systems scores as a confidence measure. Three methods of pooling LID scores into sets are used to derive the confidence measures.

1. Scores are pooled according to winner (point A in Figure 1). The target set $T$ contains scores of the correctly identified utterances. The background set $B$ contains scores of the incorrectly classified utterances (see Figure 2).

$$
T[L] = \{s(U|L) : L = L_{\text{win}} \cap L = L_{\text{input}}\}
$$

$$
B[L] = \{s(U|L) : L = L_{\text{win}} \cap L \neq L_{\text{input}}\}
$$

$$
T = \bigcup_{L=1}^{L_{\text{NL}}} \{s(U|L) : L = L_{\text{win}} \cap L = L_{\text{input}}\} \quad (4)
$$

$$
B = \bigcup_{L=1}^{L_{\text{NL}}} \{s(U|L) : L = L_{\text{win}} \cap L \neq L_{\text{input}}\}
$$

where $L_{\text{input}}$ is the actual language of the utterance.

2. Scores are pooled according to the input (point B in Figure 1). The target set contains all scores where the input and the language model correspond to the same language, the background set contains all scores where the input does not correspond to the language model (see Figure 3).

$$
T[L] = \{s(U|L) : L = L_{\text{input}}\}
$$

$$
B[L] = \{s(U|L) : L \neq L_{\text{input}}\}
$$

$$
T = \bigcup_{L=1}^{L_{\text{NL}}} \{s(U|L) : L = L_{\text{input}}\} \quad (5)
$$

$$
B = \bigcup_{L=1}^{L_{\text{NL}}} \{s(U|L) : L \neq L_{\text{input}}\}
$$

3. The third method does not separate target and background but pools all winning scores into a single set $S$ regardless of whether or not the input utterance was correctly or incorrectly classified (see Figure 4).

$$
S[L] = \{s(U|L) : L = L_{\text{win}}\}
$$

$$
S = \bigcup_{L=1}^{L_{\text{NL}}} \{s(U|L) : L = L_{\text{win}}\} \quad (6)
$$

In Methods 1 and 2, the confidence score for an utterance, classified as language $L$, is calculated by assuming Gaussian distributions of target and background sets. During testing, posterior probabilities are derived in two ways:

**Language Independent:**

$$
\text{Conf} = \Pr(L_{\text{win}} = L_{\text{input}} | s) = \frac{\Pr(T)p(s|T)}{\text{Norm}}
$$

$$
\text{Norm} = \Pr(T)p(s|T) + \Pr(B)p(s|B)
$$
Language Dependent:

\[
\text{Conf}[L_{\text{win}}] = \frac{\text{Pr}(L_{\text{win}} = L_{\text{in}} | s, L_{\text{win}})}{\text{Norm}} = \frac{\text{Pr}(T[L_{\text{win}}])p(s|T[L_{\text{win}}])}{\text{Pr}(B[L_{\text{win}}])p(s|B[L_{\text{win}}])}
\]

\[\text{Norm} = \text{Pr}(T[L_{\text{win}}])p(s|T[L_{\text{win}}]) + \text{Pr}(B[L_{\text{win}}])p(s|B[L_{\text{win}}])\]

(8)

The evaluation measures given in Table 1 clearly indicate that confidence measures derived with Method 1 track the performance of the system best. The better performance on the evaluation data may be due to the use of more training data in deriving backend and confidence score parameters.

Table 1: Evaluation of confidence scores. (cv) denotes results using cross validation on the development data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test-set</th>
<th>Language Dependent</th>
<th>Language Independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dev (cv)</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>eval</td>
<td>0.28</td>
<td>0.27</td>
</tr>
<tr>
<td>2</td>
<td>dev (cv)</td>
<td>-2.11</td>
<td>-2.22</td>
</tr>
<tr>
<td></td>
<td>eval</td>
<td>-1.66</td>
<td>-1.66</td>
</tr>
<tr>
<td>3</td>
<td>dev (cv)</td>
<td>0.18</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>eval</td>
<td>0.21</td>
<td>0.23</td>
</tr>
</tbody>
</table>

5. ADDITIONAL FEATURES

To understand how additional features help with predicting the probability of correct language identification, new features are extracted from different places (points C, and D) within the system as indicated by Figure 1. The vector of phoneme frequency of occurrence is a feature vector, where each element represents the occurrence frequency of a particular phoneme normalized with respect to the total number of phonemes occurring in an utterance. One such feature vector is created for each of the frontends. Phoneme duration is a feature vector, where each element represents the average duration of a particular phoneme normalized with respect to the average duration of all phonemes in the utterance. One such feature vector is created for each of the frontends. Unigram scores are derived in the same way that PRLM scores are derived in Equation 1 but with \( \lambda_2 = 0 \). Normalized unigrams are derived by multiplying each frontend unigram score by the factor given in Equation 12, which effectively normalizes out differences due to the different number of phones in the utterances:

\[
\frac{N!}{\prod_p p_j!} \prod_p u(p|L)^{n_p}
\]

This feature vector consists of \( NL \) scores. One such vector is derived for each of the \( PR \) phone recognizers. The Dist feature vector consists of the distance calculated between the winning score and all language models as given in Equation 13. Here, \( p(ph) \) is the unigram probability of phoneme \( ph \) in the given utterance, and \( P \) is the number of different phonemes occurring in the training set:

\[
dist[L] = \sum_{ph=0}^{P} (u(ph|L) - p(ph))\log\frac{u(ph|L)}{p(ph)}
\]

(13)
For each of these features, a classifier is built in the same manner as described in Section 2 by using cross validation on the development test set. For each classifier, the score corresponding to the winning language model, as determined by the PRLM-P system, is used for determining the final confidence score. A weighted sum of the individual confidence scores (as shown in Equation 14) from each of the features is presented as the final confidence score. This is described by Equation 14 where \( c_i^{1(0)} \) if the corresponding classifier agrees (disagrees) with the LID system.

\[
\text{Confidence} = \frac{\sum_{f} \text{Features} c_i f w_i \text{Conf}(f)}{\sum_{f} \text{Features} c_i f w_i}
\]  

where: \( w_{PRLM} = 11, w_{Norm} = 3, w_{Dia2} = 2, w_{Dar(Hindi)} = 4, w_{Freq(English)} = 3, \) and \( w_{Uni} = 2 \). These weights are chosen to optimize the confidence evaluation measure on the dev set. By using confidence scores that are derived from the weighted set of features and not only the PRLM feature, the confidence evaluation measure shows a slight improvement as shown in Table 2. As we can see from the table and Figure 5, the small differences in the evaluation measure do not seem to warrant the effort of extracting additional features.

Table 2: Evaluation of confidence scores derived from additional features. (cv) denotes results using cross validation on the development data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test-set</th>
<th>Language Dependent</th>
<th>Language Independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRLM</td>
<td>dev (cv)</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>Method 1</td>
<td>eval</td>
<td>0.28</td>
<td>0.27</td>
</tr>
<tr>
<td>PRLM+Features</td>
<td>dev (cv)</td>
<td>0.27</td>
<td>0.22</td>
</tr>
<tr>
<td>Method 1</td>
<td>eval</td>
<td>0.28</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Figure 5: Performance vs. confidence measure when using PRLM only vs. using PRLM with additional Features.

However, when we use the scores to determine the performance on a subset of data we do find that the confidence score using additional features has an advantage over using only PRLM scores. Figure 6 shows how well a confidence score that is chosen as a cutoff score from the development test set generalizes to the evaluation test set. The plot shows a desired target line, depicting perfect performance prediction on a given subset of the evaluation test set based on the development set. Results show that we are better able to predict performance when using confidence scores that use the additional features. This indicates that confidence scores with additional features are more robust over data sets thus allowing better threshold setting for flagging errors during operation.

6. CONCLUSION

We have shown that out of the three methods that were introduced, confidence measures are best extracted using the winning scores and modeling true winners vs. false winners to estimate the probability of correct recognition. Using this confidence measure estimator, we are able to predict the probability of correct LID on a file-by-file basis. We also introduced the use of additional features to improve the confidence measure. We found these additional features did not improve the confidence measure as evaluated by Equation 10, but were useful for predicting performance on a subset of evaluation data. In future work we plan to study the effect of using the additional features for out-of-set language detection.

7. REFERENCES

