Scoring Algorithms for Wordspotting Systems

Robert W. Morris and **Jon A. Arrowood** and **Peter S. Cardillo**
Nexidia Inc.
3060 Peachtree Rd Suite 730
Atlanta, Georgia 30305-2240
{rmorris,jarrowood,pcardillo}@nexidia.com

Mark A. Clements
Center for Signal & Image Processing
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
clements@ece.gatech.edu

Abstract

When evaluating wordspotting systems, one normally compares receiver operating characteristic curves and different measures of accuracy. However, there are many other factors that are relevant to the system's usability for searching speech. In this paper, we discuss both measures of quality for confidence scores and propose algorithms for producing scores that are optimal with respect to these criteria.

1 Introduction

In order to evaluate any system, it is useful to have objective quality measures that can be automatically applied to systems for comparison. For wordspotting systems, these measures are oriented towards recall accuracy. Most of these measures are based on receiver operating characteristic (ROC) curves and functions of these curves. However, there are many other factors that are relevant to the systems usability.

When a user enters a query to the Nexidia wordspotter (Clements et al., 2001), the system returns a sorted result list that marks the times where the query matches the audio. In addition, scores are associated with each result. These scores are related to the likelihood that the tagged audio matches the query. Although this score gives an indication of the strength of the match, users have had difficulty interpreting the scores.

We found that most users want to use the score in one of two ways. The first application is to provide a score threshold for monitoring applications. Alternatively, people also assume that the score reflects the probability that the tagged audio segment is actually a match.

However, without any objective quality measure of these scores, it was difficult to evaluate different score generation algorithms. In this paper, we discuss both measures of quality for confidence scores and propose algorithms for producing scores that are optimal with respect to these criteria.

2 Assumptions

In order to derive a scoring algorithm, a key assumption must be made by the wordspotting algorithm: each match must have a numeric score associated with it. In addition, there must be some theoretical basis for an additive decomposition of this score. This decomposition is given by

$$R^{(q)} = \sum_{l=1}^{L} R_l^{(q)}, \tag{1}$$

where $R^{(q)}$ is the score returned by query q, and $R_l^{(q)}$ is the score associated with the lth phoneme in the query. With this assumption, we also assume that these components can be modeled with a Gaussian distribution with dependence on whether the match is truly a hit or a miss. The distributions are then given by

$$R_l^{(q)}|\text{Hit} \sim \mathcal{N}\left(\mu_H\left(S_l^{(q)}\right), \sigma_H^2\right) \tag{2}$$

$$R_l^{(q)}|\text{Miss} \sim \mathcal{N}\left(\mu_M\left(S_l^{(q)}\right), \sigma_M^2\right), \tag{3}$$

where $S_l^{(q)}$ is the lth phoneme in query q. In this model, the means, μ are dependent on the phoneme, but the variance, σ^2, is not. Using the additive model, the raw scores are distributed by

$$R^{(q)}|\text{Hit} \sim \mathcal{N}\left(\sum_{l=1}^{L}\mu_H\left(S_l^{(q)}\right), L\sigma_H^2\right) \tag{4}$$

$$R^{(q)}|\text{Miss} \sim \mathcal{N}\left(\sum_{l=1}^{L}\mu_M\left(S_l^{(q)}\right), L\sigma_M^2\right). \tag{5}$$

3 Performance Measures

We propose two scoring evaluation measures. In each of these methods, the raw score is modified by some scoring function $F()$. The first measure evaluates a scoring algorithms usefulness for setting detection thresholds. This method assumes that the scoring function calculates the cdf of the missed score distributions. The measurement is based on the Kolmogorov-Smirnov test statistic, which is given by

$$KS = \max_i \left| F\left(R_M^{(i)}\right) - \frac{i}{N} \right| \tag{6}$$

where $R_M^{(i)}$ are the raw scores for the false alarms in descending order.

A metric for measuring scoring algorithms based on result confidence is given by

$$B = \frac{1}{N_H}\sum_{n=1}^{N_H}\left[1-F\left(R_H^{(n)}\right)\right]^2 + \frac{1}{N_M}\sum_{n=1}^{N_M}\left[F\left(R_M^{(n)}\right)\right]^2, \tag{7}$$

where N_M and N_H are the number of hits and misses. This value is equal to zero when all hits are scored to one and all misses are scored as zero. On the other hand, B is equal to 0.5 if $F(R)$ is set to 0.5 regardless of the input.

4 Algorithms

If one is interested in setting a detection threshold based on false alarms per hour, then one can set the score using the cumulative density function of the misses. This yields the score

$$\begin{aligned} F_C\left(R^{(q)}\right) &= \Pr\left(x < R^{(q)}\right) \\ &= Q\left[\frac{1}{\sqrt{L}\sigma_M}\left(R^{(q)} - \sum_{l=1}^{L}\mu_M\left(S_l^{(q)}\right)\right)\right], \end{aligned} \tag{8}$$

where Q is the cdf of the unit normal distribution. To set a threshold for K false alarms per hour, then the threshold should be set to

$$\alpha = 1.0 - \frac{K}{K_T}, \tag{9}$$

where K_T is the range of false alarms per hour that the miss model is trained.

If one is looking at a list of scores, one might be interested in the probability that the score was generated by a true match. By Bayes law, the conditional probability can be calculated by

$$\begin{aligned} F_B\left(R^{(q)}\right) &= \Pr\left(\text{Hit}|R^{(q)}\right) \\ &= \frac{P_H p(R^{(q)}|\text{Hit})}{P_H p(R^{(q)}|\text{Hit}) + (1-P_H)p(R^{(q)}|\text{Miss})}, \end{aligned} \tag{10}$$

where P_H is the prior probability of a hit.

5 Model Training

Each of the scoring methods described above require models of how the phonemes relate to the scores through the parameters: μ_M, μ_H, σ_M^2, and σ_H^2. For this purpose, a series of hits and misses over the desired range of false alarms rates must be collected from the wordspotter. With these scores, it is possible to train the miss and hit models independently. For this reason, only the miss model training is described here.

Given the model in Equation 5, the following distribution holds with N observations:

$$\begin{aligned} &p(\mathbf{R}|\mathbf{S}, \mu_M, \sigma_M^2) = \\ &\quad \prod_{n=1}^{N}\mathcal{N}\left(R^{(n)} - \sum_{l=1}^{L}\mu_M\left(S_l^{(n)}\right), L\sigma_M^2\right). \end{aligned} \tag{11}$$

The maximum likelihood solution for μ_M and σ_M^2 is a difficult optimization problem. However, if the phoneme components $R_l^{(n)}$ from Equation 1, the distribution simplifies to observations of the Gaussian components. By using the Expectation Maximization (EM) algorithm, the overall likelihood in Equation 11 can be iteratively maximized (Dempster et al., 1977).

Similarly, the training problem can also be viewed in a Bayesian framework, where a Minimum Mean Squared Error (MMSE) estimate can be calculated. Like the maximum likelihood estimate, this requires an iterative method where the components of the score are generated. This can be computed by a Gibbs sampler (Gamerman, 1997).

In addition to providing a mechanism for creating meaningful scores, these models can be useful for other purposes. For example, one can analyze the mean vectors to determine which phonemes provide better discrimination for wordspotting. These can also be used to diagnose problems in performance that are phoneme specific.

6 Results

The experiments for this algorithm were conducted using the Nexidia wordspotting system trained on broadcast quality North American English speech. The effect of using different scoring algorithms was accomplished using a nine hour subset of the HUB-4 1996 North American English broadcast corpus. This data was chosen since this corpus is widely available and is disjoint from the training data used for the wordspotter. From this corpus, 8500 search terms were randomly selected from the transcripts. These queries were equally distributed in length from 4 to 20 phonemes, and then split into a testing and training set. For each search term, results ranging from the top score down to the 90th false alarm were collected. The results from the training terms were then used to train the score models using both the EM algorithm and a Gibbs sampler.

These trained models were then then used to generate both F_B and F_C for all of the test queries. In addition, the "Standard" scores were generated. These scores are what the Nexidia wordspotting product reveals to the users, and are calculated by scaling the raw scores by the number of phonemes and mapping these from zero to one.

The resulting scores from these tests are listed in Table 1. As expected, the CFAR based score performed well on the KS metric, while the Bayesian score was more accurate on the B measure. Both of these methods performed much better than the previous ad-hoc "Standard" method. However, performance improvements on one measure resulted in very poor scores on the other. This is due to the fact that the objective of each measure is very different. In addition, the estimation scheme had little effect on the overall scores. Since the EM algorithm requires a small fraction of the computation that the Gibbs sampler requires, this method is preferable.

Table 1: Comparison of different scoring algorithms based on two scoring measurements

Algorithm		**Performance Measure**	
		KS	B
Gibbs	CFAR	0.312	0.350
	Bayes	0.790	0.197
EM	CFAR	0.322	0.351
	Bayes	0.789	0.196
Standard		0.633	0.496

To illustrate the differences between the three scoring algorithms, the hits and misses were also collected and plotted in Figure 6. In each subplot, there are histograms of the hits and misses. In all three cases, most of the hits tend to have scores close to one. However, the misses in the standard scoring scheme are concentrated from 0.5 to 0.8. When the Bayes scoring method is used, half of the hits are very close to 1.0, while half of the misses are very close to 0.0. The other half of the scores are distributed along the score range. Finally, the misses from the CFAR scoring algorithm are distributed evenly along entire range of scores. Because the normal score assumption does not strictly hold, this distribution is not perfectly flat at the start and the end, but it is fairly close.

7 Conclusions

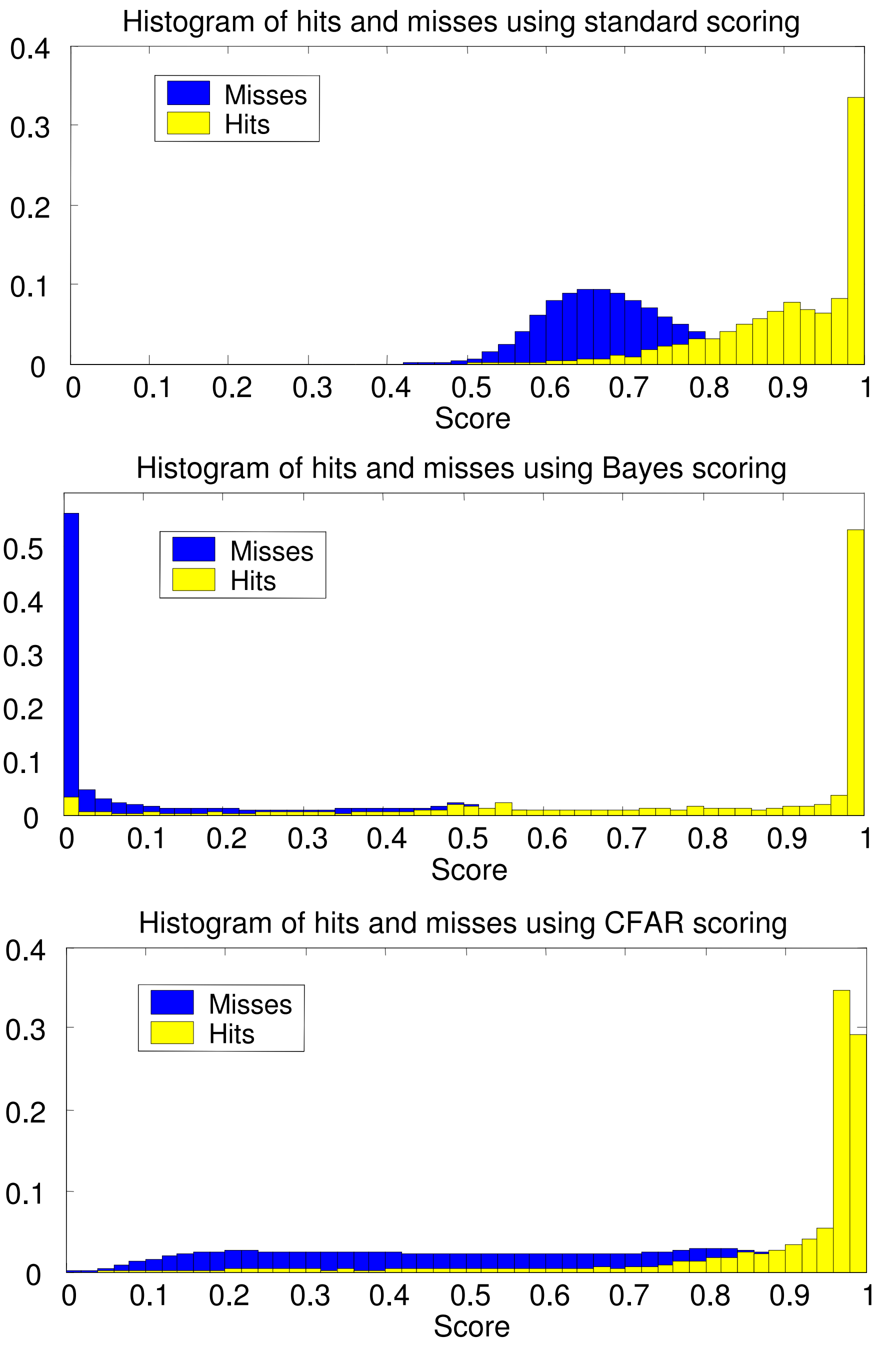
Several methods for for both generating and evaluating scores from wordspotting systems have been proposed. These methods can operate on any system that generates scores where an additive model based on phonemes is valid. The scores that are produced by the algorithms described can be used to both give intuitive confidence levels, as well as provide a simple mechanisms for setting thresholds in monitoring environments. These methods have been shown to provide superior performance when compared to their relevant metrics.

References

M. A. Clements, P. S. Cardillo, and M. S. Miller. Phonetic searching vs. LVCSR: How to find what you really want in audio archives, in *AVIOS 2001*.

Dani Gamerman. 1997. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, volume 1. Chapman & Hall, Boca Raton, FL.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38.



Histogram of hits and misses using standard scoring

Histogram of hits and misses using Bayes scoring

Histogram of hits and misses using CFAR scoring

Figure 1: Comparison of different scoring methods on Broadcast English queries. Scores are derived from results ranging from zero to ten false alarms per hour.