SPEAKER DETECTION WITHOUT MODELS

Daniel Gillick, Stephen Stafford, and Barbara Peskin

International Computer Science Institute
Berkeley, CA 94704 – USA
{dgillick,sjs,barbara}@icsi.berkeley.edu

ABSTRACT

In order to capture sequential information and to take advantage of extended training data conditions, we developed an algorithm for speaker detection that scores a test segment by comparing it directly to similar instances of that speech in the training data. This non-parametric technique, though at an early stage in its development, achieves error rates close to 1% on the NIST 2001 Extended Data task and performs extremely well in combination with a standard Gaussian Mixture Model system. We also present a new scoring method that significantly improves performance by capturing only positive evidence.

1. INTRODUCTION

Traditionally, speaker detection tasks have involved at most a few minutes of training speech and a minute of test speech. Under these circumstances, the Gaussian Mixture Model (GMM), which lumps all speech frames together to create a generic frame model, is a reasonable approach. It succeeds by virtue of its simplicity. More recent techniques, however, attempt to capitalize on greater amounts of data available through NIST's Extended Data task to capture speaker information that exists in longer speech patterns. Language modeling, duration modeling, and modeling of various prosodic cues have all proven useful [1, 2, 3]. The motivation for our research is that given enough training examples, we can avoid such explicit parameterized modeling altogether, and score test tokens by comparing them directly to similar instances in the training data.

A speaker detection system based on such direct comparison is not unprecedented. Higgins et al. developed a frame-level nearest-neighbor approach that was competitive with a GMM system in the early '90s [4]. Dragon Systems extended this idea to sequences of frames a few years later with encouraging results (see description of the Sequential Non-Parametric system (SNP) in [5]). More recently, the speech recognition community has begun to look into example-based techniques to enhance long-standing HMM standards (e.g. [6, 7]), and in the last few months, it has come to our attention that [8] used a dynamic time warping word-spotting technique to find and compare similar test and training frame sequences.

Our motivating intuition is that if we want to know whether a test speaker is the same as the target speaker, we ought to look for very good acoustic matches that are as long as possible. While we are not convinced of anything by long poor matches (as people say things differently on different occasions), and only somewhat encouraged by short good matches (it is conceiveable that two people could produce very similar frames or even short frame sequences), it is precisely the long good matches that ought to be most useful for speaker detection. With this in mind, we have begun to build a framework for example-based speaker detection.

This research, which represents an expansion of Dragon's SNP system, is still exploratory. Compared with GMM or Hidden Markov Model (HMM) approaches, little is known about the behavior of these example-based systems. In the sections that follow, we describe the design and implementation of our system and then discuss some preliminary but promising results on the NIST 2001 Extended Data task.

2. THE ALGORITHM

We use output from an automatic speech recognizer (ASR) to partition the test and target training speech streams into "tokens". Test-target pairs are then scored, using nearest-neighbor techniques, by measuring the frame-level distances between test tokens and instances of matching target tokens. The following subsections provide greater detail.

2.1. Feature extraction

As a front-end, we create feature vectors with 20 mel-frequency cepstral coefficients (MFCCs), C_0 - C_{19}, and their first derivatives for a total of 40 features per 10-ms frame. Cepstral Mean Subtraction (CMS) is used at the utterance level to correct for some simple channel differences.

2.2. Token labels for data

Using transcriptions and time-alignments from the SRI recognizer [9], we divide the data into tokens. These could be phones or words or word bigrams or anything else. While we know from Dragon's work on the SNP system that short tokens like phones are effective, we hope to show that even more speaker-discriminating power is contained in longer tokens. Note that because we are using ASR output, both the word identity and the alignment information are highly errorful. To get a sense for the costs of these errors, we will contrast systems that use ASR output and force-aligned truth transcripts, below.

2.3. Comparing test and target speakers

Comparing test and target utterances involves pairwise comparisons between each test token and every instance of that token in

This material is based upon work supported by the National Science Foundation under Grant No. 0329258.

the target training data. When we compare two tokens, we use the Euclidian metric to calculate the distance between aligned frames. In keeping with the nearest-neighbor strategy, we retain only the best test-target pairing for each test token.

Before we can take such measurements, though, we need to decide how to line up the frames. The simplest method is to align the first frames of the test and target tokens and the second frames and so on, and stop when we reach the end of the shorter token. We might instead choose to slide the shorter token through the longer token, looking for the best place to start matching one-to-one. A third possibility is to use some sort of dynamic time warping (DTW). DTW is the standard solution to the problem of comparing speech tokens of different lengths since it is assumed that certain sounds (vowels, for example) might be more prone to duration variation than others. Recall, however, that for the purpose of speaker detection, we are really interested in long good matches. Ideally, we would like to find cases where the test token matches the target token exactly, without any stretching or shrinking, as it is reasonable to assume that there is speaker information in the duration of those sounds. We compare these alignment strategies in the next section.

Once we have a score for each test token, we need to decide how to produce an overall score. One method is simply to take an average of the token scores. We will call this the *basic-score*. Specifically, we sum the unnormalized test-token scores and divide by the total number of frames. A second method involves keeping only positive evidence. Since we are intuitively more convinced by really good scores, we use a *hit-score* (HS) to weight good scores more heavily than bad ones. The hit-score for an entire test-target comparison is computed as follows:

$$\text{HS} = \sum_{i \in \text{test tokens}} \frac{\text{number of matched frames in } i}{k^{\text{score}[i]}} \qquad (1)$$

This formulation lets bad (larger) scores drop out as effective zeros and gives exponentially heavier weight to good (smaller) scores. The value of the constant k is estimated empirically to be 2. We compare results obtained from these different scoring methods in the next section.

2.4. Normalizing the scores

It is well known that raw scores need to be normalized so that the scores assigned to various test-target pairs are comparable. Normalization is especially important in our case since we are not adapting from any background model (as is customary with GMM or HMM systems) which naturally tends to center the scores.

We apply two standard normalizations, one to correct for the variability of the test data, and one to correct for the variability of the training data. For the former, a GMM system would subtract from the score of each test-target pair the score that the test segment receives against a background model to create the usual log-likelihood ratio score. Since we have no such background model, we create a "pseudo-speaker" whose speech consists of conversations from a number of different held-out speakers. We subtract the test-pseudo-speaker score from each test-target score. To address target variability, we use ZNORM. Specifically, a set of held-out impostor samples are each scored against the target training data in question. We subtract the mean impostor score from the test-target score and divide by the standard deviation.

3. EXPERIMENTS

Our experiments are based on the Extended Data Task from the NIST 2001 Speaker Recognition Evaluation [10], a text-independent single-speaker detection task using data obtained from the Switchboard-I corpus. This data set consists of about 2400 telephone conversations among 543 speakers (302 male, 241 female) collected in the early 1990s by Texas Instruments. The speakers are divided into 6 independent "splits" so that when testing on one split, the others can be used for normalization.

In the evaluation, 1, 2, 4, 8, and 16 conversation-side training conditions are specified (the average conversation side contains 2 - 3 minutes of speech). We focus our attention on the 8-conversation-side training, which has become a standard for data-intensive algorithms.

3.1. General results

Table 1 shows results for various choices of tokens. DTW is used to determine frame alignments; the final scores are linear combinations of the basic-score and the hit-score. Performance of a typical GMM system (provided by SRI) on this data is included for reference. The Equal Error Rate (EER) and the minimum of the Decision Cost Function (DCF) represent two points on the now standard Detection Error Tradeoff (DET) curve [11]. The EER is the point where the two error types, false alarms and misses, occur with equal relative frequency, while the DCF, as specified by NIST, weights false alarms and misses in accordance with the demands of many real-world applications.

Token	**EER(%)**	**DCF**
phones	1.85	0.0937
phone bigrams	1.25	0.0685
phone trigrams	1.14	0.0604
words	1.44	0.0736
word bigrams	2.09	0.1130
GMM	0.90	0.0509

Table 1. *System performance for various tokens in terms of EER and min. DCF on NIST's 2001 Extended Data Task. Typical GMM performance is provided for comparison.*

The best performance is obtained using phone trigrams, an encouraging result given our hypothesis that longer tokens ought to be more useful for speaker detection. Word bigrams, however, give poorer performance than single words. To understand these scores, we need to consider the tradeoff between the increased power of longer tokens and data sparsity. For example, there are only 47 different phone tokens, but around 8600 different phone-trigrams, so that a test phone might have a few thousand choices for a target match while a test phone-trigram might have fewer than ten. The chances of finding a good match for a phone-trigram are increased by the additional context information inherent in the longer utterance, but often this advantage is outweighed by overwhelmingly more abundant data for single phone matches. Note that while finding a good match is useful evidence, not finding a good match could mean that the test and target are in fact different speakers or that we simply have not seen enough target instances to make any claims, one way or the other. Thus, using shorter tokens limits the sort of uncertainty that arises from data sparsity but sacrifices the greater confidence gained from matching longer tokens. In the case of word bigrams, even the most common tokens ("you know", "I think", etc.) only appear a few times in each conversation, even further exacerbating the data-sparsity issue.

3.2. ASR vs. truth

As mentioned earlier, the token labels and time-alignments are obtained from an ASR system which makes mistakes. Specifically, the ASR output, provided by SRI, is the product of a simplified 1-pass recognition system using only a bigram language model to compensate for the fact that the recognizer was trained on Switchboard-I data (our test data). This system achieved an average word error rate (WER) of about 30% on this material. As it turns out, current state-of-the-art recognizers now obtain WERs in the teens on Switchboard-I, so are now closing the gap between the "truth transcripts" and "ASR output" reported here.

Token	ASR	Truth
word bigrams	2.09	1.17
phone trigrams	1.14	1.03

Table 2. *System performance (EER) using ASR and truth transcripts.*

The truth transcripts significantly improve our results, especially when we use word-level tokens. This might be because the alignments are based on human transcriptions of the words rather than the phones. While more accurate word identities would provide more viable token matches, the phone identities given by the ASR might be close enough to the truth to allow for good matching. Nonetheless, this experiment demonstrates that we could significantly improve our results given better ASR, but more importantly, it might be to our benefit to find a way to exclude ASR from our system entirely, perhaps by using some more data-driven clustering algorithm to group similar frame sequences.

3.3. Scoring methods

All of the scores reported thus far are simple linear combinations of the basic-score and the hit-score methods discussed earlier. In addition to these, we also experimented with a third scoring method which focuses on negative evidence which the hit-score ignores. Intuitively, we might be persuaded that a speaker does not match a target if there is a case where even in the presence of many potential token matches, there is not a single good score. The formula for this *negative-score* (NS) closely resembles the hit-score but lets positive evidence drop out, emphasizing bad scores:

$$\text{NS} = \sum_{i \in \text{test tokens}} \frac{(\text{matched frames in } i)(\text{target instances})}{k^{M - \text{score}[i]}} \qquad (2)$$

Again, k is set at 2, while M represents a best guess at the maximum score. We compare these scoring methods in table 3.

Token	Basic	HS	NS
phones	2.25	1.98	38.9
phone trigrams	2.01	1.30	37.0

Table 3. *Performance (EER) of basic-score, hit-score, and negative-score for phones and phone trigrams.*

As it turns out, negative evidence, at least as we have formulated it here, is almost meaningless. This may be one reason why the hit-score outperforms the basic-score, which factors in all evidence, positive and negative. More specifically, we can assume that not all tokens have the same speaker-discriminating power, so that the hit-score benefits from extracting only crucial matches that are compromised in the basic-score's across-the-board averaging approach. Perhaps this is why the disparity between the hit-score and the basic-score is greater for phone trigrams than for phones – the phones are all fairly useful tokens but the phone trigrams tend to vary in their speaker-discriminating power, whether because of their intrinsic value or because of sparsity constraints, and the hit-score is better at picking out the useful information.

3.4. Frame alignment methods

All the results reported so far have used DTW to align the frames for token comparisons. How do the other alignment strategies discussed earlier perform? What if we allow unconstrained matching – i.e. within the bounds specified by the token start and end times, we let each test frame match any target frame?

Token	DTW	SW	FF	UNC
phones	1.85	2.60	2.56	2.49
phone trigrams	1.14	1.14	1.17	1.44

Table 4. *Performance (EER) with various frame alignment methods (SW = sliding window, FF = first frames aligned, UNC = unconstrained frame matching)*

When we use individual phones as tokens, DTW is clearly preferred, giving significantly better performance than the other frame alignment methods. However, we see a surprising result when we use phone trigrams. We expected that DTW would be more important for longer tokens than for shorter ones, in effect correcting for the scarcity of long well-matched test-target tokens. And yet, our DTW algorithm shows no clear advantage over the non-time-warping methods in this case. At the moment, we are not sure why this is the case, but we suspect that the problem might lie in our implementation of DTW, which currently limits the options for doubling and skipping frames. It is also possible that given such low error rates, we have hit a performance barrier which can only be broken once we have dealt with other issues such as channel mismatches. On the positive side, we are encouraged by the remarkable performance of the non-time-warping methods, as well as by the fact that the sequential methods provide a significant advantage over the unconstrained approach for the longer tokens.

3.5. System combinations

Another important test of a new speaker detection system is how well it combines with a standard GMM baseline. We designed our system to capitalize on sequence information that the GMM neglects with the expectation that the two methods would perform well together. While our phone-trigram system is slightly behind the GMM, a simple 50-50 linear combination of the two systems yields a huge improvement. Most notably, the DCF drops by more than a factor of 3, from 0.0509 (GMM) to 0.0157 (combined).

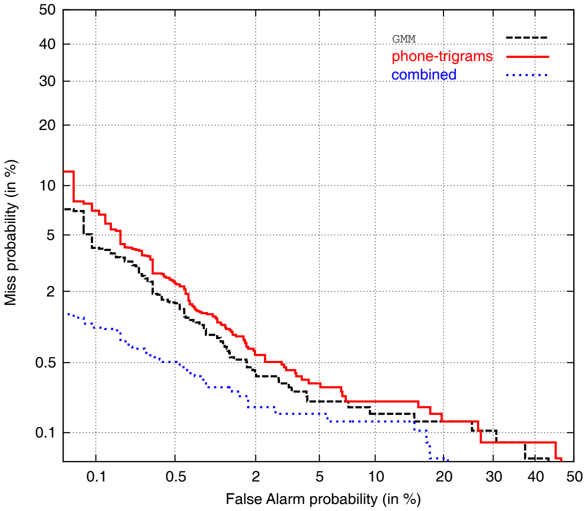

Fig. 1. *DET plots for the phone-trigram system, the GMM system, and a linear combination of the two.*

4. FUTURE WORK

This system is still at an early stage of development and much work remains to be done to realize its full potential. Our future work can be split into short-term experiments and long-term projects. Short-term efforts include:

- Expanding the background and ZNORM sets. Both normalizations have proven extremely helpful (phone trigram raw score = 9.5% EER; after background normalization = 2.8% EER; after ZNORM = 1.1% EER) but the sets were kept small to minimize computation.
- Trying other distance metrics. We ought to try performing Linear Discriminant Analysis (LDA) on the features before using the Euclidian metric or choosing a different distance measurement.
- Improving the DTW algorithm. Our DTW algorithm supports variable penalties for skipping and doubling frames, but currently, we simply assign a constant penalty for all frame skips. Since we know that the errorful ASR alignments reduce performance considerably, we ought to consider reducing or eliminating the cost of frame skipping at the edges of tokens, for example.
- Changing the front-end. Given that our scores are closely tied to the raw features, we stand to benefit from a feature mapping algorithm [12]. While CMS roughly corrects for some channel variation, feature mapping more carefully places utterances into a "handset-independent" space.

The long-term goals for this project are more exploratory. Though at present we simply compare a fixed set of test and target tokens using DTW instead of a more standard GMM or HMM approach, we should exploit the freedom of our example-based method to do dynamic token selection, perhaps in a style comparable to variable-length unit selection employed by some Text-To-Speech systems. We could use the longest test strings for which there exist sufficient target instances, and back off to shorter test-target matching if necessary. We could also imagine an even more general, more data-driven approach. If we search dynamically for long good test-target matches at the frame level rather than at the token level, we can avoid using ASR entirely, which has the potential to be both faster and more accurate.

5. CONCLUSIONS

As more and more data becomes available for extended data tasks, we would like to test our intuition that finding long good acoustic matches between test and training data is the key to speaker detection. Our initial experiments have yielded promising results and we look forward to expanding and improving our system.

6. ACKNOWLEDGEMENTS

The authors would like to thank everyone in the speaker-id group at ICSI and at SRI for providing technical assistance and stimulating conversation. In particular, George Doddington championed the hit-scoring idea, Kofi Boakye generated front-end features, and SRI provided their GMM system scores and ASR output.

7. REFERENCES

[1] G. Doddington, "Speaker Recognition Based on Idiolectal Difference between Speakers," in *Proc. Eurospeech-2001*, pp. 2521–2524.

[2] L. Ferrer *et al.*, "Modeling Duration Patterns for Speaker Recognition," in *Proc. Eurospeech-2003*, pp. 2017–2020.

[3] D. Reynolds *et al.*, "The SuperSID Project: Exploiting High-Level Information for High-Accuracy Speaker Recognition," in *Proc. ICASSP-2003*, pp. 784–787.

[4] A.L. Higgins, L.G. Bahler, and J.E. Porter, "Voice Identification Using Nearest-Neighbor Distance Measure," in *Proc. ICASSP-1993*, vol. II, pp. 375–378.

[5] A. Corrada-Emmanuel, M. Newman, B. Peskin, L. Gillick, and R. Roth, "Progress in Speaker Recognition at Dragon Systems," in *Proc. ICSLP-1998*, vol. 4, pp. 1355–1358.

[6] M. De Wachter, K. Demuynck, D. Van Compernolle, and P. Wambacq, "Data Driven Example Based Continuous Speech Recognition," in *Proc. Eurospeech-2003*.

[7] S. Axelrod and B. Maison, "Combination of Hidden Markov Models with Dynamic Time Warping for Speech Recognition," in *Proc. ICASSP-2004*, pp. 173–176.

[8] H. Aronowitz *et al.*, "Text Independent Speaker Recognition Using Speaker Dependent Word Spotting," in *Proc. ICSLP-2004*.

[9] A. Stolcke *et al.*, "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, College Park, MD, 2000.

[10] "NIST 2001 Speaker Recognition website," http://nist.gov/speech/tests/spk/2001.

[11] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," in *Proc. Eurospeech-1997*, vol. 4, pp. 1895–1898.

[12] D. Reynolds, "Channel Robust Speaker Verification via Feature Mapping," in *Proc. ICASSP-2003*, vol. II, pp. 53–56.