A GENERAL DISCRIMINATIVE TRAINING ALGORITHM FOR SPEECH RECOGNITION USING WEIGHTED FINITE-STATE TRANSDUCERS

Yong Zhao[⋆] *Andrej Ljolje*[†] *Diamantino Caseiro*[†] *Biing-Hwang (Fred) Juang*[⋆]

[⋆] Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
[†] AT&T Labs-Research, Florham Park, USA

ABSTRACT

In this paper, we present a general algorithmic framework based on WFSTs for implementing a variety of discriminative training methods, such as MMI, MCE, and MPE/MWE. In contrast to the ordinary word lattices, the transducer-based lattices are more amenable to representing and manipulating the underlying hypothesis space and have a finer granularity at the HMM-state level. The transducers are processed into a two-layer hierarchy: at a high level, it is analogous to a word lattice, and each word transition embodies an HMM-state subgraph for that word at a lower level. This hierarchy combined with the appropriate customization of the transducers leads to a flexible implementation for all of the training criteria being discussed. The effectiveness of the framework is verified on two speech recognition tasks: Resource Management, and AT&T SCANMail, an internal voicemail-to-text task.

Index Terms: discriminative training, speech recognition, weighted finite-state transducer.

1. INTRODUCTION

Over the past decade, discriminative training has achieved a significant performance improvement in various large vocabulary continuous speech recognition (LVCSR) systems. Representative discriminate training criteria include maximum mutual information (MMI) [1], minimum classification error (MCE) [2], and minimum phone/word error (MPE/MWE) [3].

Recent research [4], [5], [6] showed that many such discriminative training criteria can be formulated in a unified objective function form, where choices of the smoothing function, the gain function, and the hypothesis space resolve to different discriminative training criteria. The unified view of discriminative training highlights the computational connections between these training criteria. It thus facilitates the implementation of a general framework for discriminative training, where various optimization techniques can be fairly compared and their differences quantified.

In this paper, we present an algorithmic framework with sufficient generality for implementing various discriminative training methods using weighted finite-state transducers (WFSTs). With the work in [7], the use of WFSTs has become a fundamental tool in speech and language processing. In speech recognition, they allow an elegant integration of knowledge sources such as the context dependency, the pronunciation lexicon and the language model in a precompiled and very efficient search network. Here, the power of the WFSTs is leveraged to represent the lattice of the hypothesis space to achieve a general discriminative training framework.

Compared with the ordinary word lattices for discriminative training [4], [3], transducer-based lattices have the following advantages. First, many of the off-the-shelf operations and algorithms established for WFSTs [8] can be directly applied to represent and manipulate the lattices. The flexibility attached to the lattices will considerably boost a general and efficient implementation of discriminative training. Take the MCE training as an example, where the hypothesis space should exclude the reference sequence. This can be simply realized by taking the difference of the recognition lattice and the reference sequence.

Moreover, the input symbols of our transducer-based lattice are HMM states, acting as elementary units of the lattice. This makes it more efficient in representing the hypothesis space than the ordinary word lattices, because given the similar size of HMM states contained within the lattice, the HMM-state lattice will typically produce much more hypothesis sequences than the word lattices. Note that the richness of the competing hypotheses plays a crucial role in the discriminative training.

The use of transducer-based lattices for discriminative training has actually been reported in the literature [9], [10], [11]. However, most of these works focus on one or a few discriminative training criteria, and the transducer is used as an ordinary word (or phone) lattice. In this paper, to accommodate the calculation of various substring-level errors, the transducer-based lattices are processed into a two-layer hierarchy: at a high level, it is analogous to a word lattice, and each word transition embodies an HMM-state subgraph for that word at a lower level. A number of issues ensuring the general implementation of discriminative training are addressed, including disambiguating word tokens in the transducer-based lattices, collecting statistics for different substring errors, and synchronizing word labels with context-dependent HMM states.

The remainder of the paper is organized as follows. In Section 2, the unified discriminative training criterion is introduced. The general training framework based on WFSTs and several implementation issues are described in Section 3 and Section 4, respectively. Finally, we present the experimental results and conclusions in Section 5 and Section 6, respectively.

2. UNIFIED VIEW OF DISCRIMINATIVE TRAINING

MMI [1], MCE [2], and MPE/MWE [3] represent three major forms of discriminative training for speech recognition. Recent research [4], [5] showed that many such discriminative training criteria can be formulated in a unified objective function, which is related to the weighted average of some predefined accuracy function over all hypothesis sequences. Suppose we have a set of R training sentences, where $\boldsymbol{O}^{(r)}$ is the acoustic observations of the r-th training utterance with the reference transcription $s_{\text{ref}}^{(r)}$. Then the unified discriminative training criterion $\mathcal{F}$ for optimizing the acoustic model parameters $\boldsymbol{\Lambda}$ can be expressed as [4]

$$\mathcal{F}(\boldsymbol{\Lambda}) = \sum_{r=1}^{R} f\Big(\log \frac{\sum_s P_\Lambda(\boldsymbol{O}^{(r)}, s)\mathcal{A}(s, s_{\text{ref}}^{(r)})}{\sum_{s\in S^{(r)}} P_\Lambda(\boldsymbol{O}^{(r)}, s)}\Big) \quad (1)$$

where $P_\Lambda(\boldsymbol{O}, s)$ denotes the joint probability of an observation sequence $\boldsymbol{O}$ and a hypothesis sequence s, $f(\cdot)$ denotes the smoothing function, and the gain function $\mathcal{A}(s, s_{\text{ref}})$ measures the accuracy of hypothesis s given its reference s_{ref}. S is the hypothesis space comprising a set of competing hypothesis sequences for the acoustic data $\boldsymbol{O}$. In the context of LVCSR, the hypothesis space is typically represented as a word lattice. As shown in [4], the particular choice of the smoothing function f, the hypothesis space S, and the gain function $\mathcal{A}(s, s_{\text{ref}})$ resolves to different discriminative training criteria, three of which are tabulated in Table 1 for comparison.

Table 1. Choice of parameters for different discriminative training criteria.

Criterion	$f(x)$	Hyp. space S	Gain
MMI	x	all	$\delta(s, s_{\text{ref}})$
MCE	$\frac{-1}{1+\exp(\varrho x)}$	all but s_{ref}	$\delta(s, s_{\text{ref}})$
MPE/MWE	$\exp(x)$	all	$\mathcal{A}(s, s_{\text{ref}})$

The unified view of discriminative objective functions highlights the computational connections between various discriminative training approaches. The key difference between these criteria comes from the definition of the gain function $\mathcal{A}(s, s_{\text{ref}})$. MMI and MCE uses the Kronecker delta gain function $\delta(s, s_{\text{ref}})$, which amounts to measuring the string-level accuracy, whereas the gain functions of MPE/MWE [3] exploit the substring-level accuracy, which appears more pertinent to the performance evaluation metric. In addition, MCE differs from the other criteria in that its hypothesis space should exclude the reference sequence s_{ref}.

One popular optimization method for discriminative training is the extended Baum-Welch (EBW) algorithm [3]. The parameter re-estimation relies on two lattice-based statistics: the posterior probability passing through arc e of the lattice S, $\gamma(e) = \frac{\sum_{s \in S: e \in s} P_\Lambda(\boldsymbol{O}, s)}{\sum_{s \in S} P_\Lambda(\boldsymbol{O}, s)}$, and the average gain over the hypotheses passing through arc e, $c(e) = \frac{\sum_{s \in S: e \in s} P_\Lambda(\boldsymbol{O}, s)\mathcal{A}(s, s_{\text{ref}})}{\sum_{s \in S: e \in s} P_\Lambda(\boldsymbol{O}, s)}$. The determination of $\gamma(e)$ is analogous to the state posterior probability in an HMM, and can be evaluated in a standard forward-backward procedure. In [3], $c(e)$ is calculated in a similar forward-backward fashion under an appropriate assumption of the gain function $\mathcal{A}(s, s_{\text{ref}})$. Section 3.4 will give a detailed derivation of $c(e)$ within the WFST framework.

3. DISCRIMINATIVE TRAINING WITH WFSTS

In this section, we present a WFST-based framework with sufficient generality for statistics collection and parameter optimization of different discriminative training criteria. The AT&T FSM library [8] is used for the representation and manipulation of WFSTs.

3.1. Weighted Finite-State Transducers

A WFST T over a semiring $\mathbb{K}$ can be defined as a 8-tuple $T = (\Sigma, \Delta, Q, I, F, E, \lambda, \rho)$, where Σ and Δ are the respective input and output alphabets, Q is a finite set of states, $I \subseteq Q$ and $F \subseteq Q$ are the respective sets of initial states and final states, $E \subseteq Q \times \Sigma \times \Delta \times \mathbb{K} \times Q$ is a finite set of arcs, and $\lambda : I \mapsto \mathbb{K}$ and $\rho : F \mapsto \mathbb{K}$ are the initial and final state weight assignments, respectively. Some algorithms require a potential function $\tau : Q \mapsto \mathbb{R}$ associated with each state.

Given an arc $e \in E$, we denote its input label by $i[e]$, its source state $p[e]$, its destination state $n[e]$, its weight $w[e]$, and its output label $o[e]$. A path $s = e_1^k$ is a sequence of consecutive arcs such that $n[e_{i-1}] = p[e_i], i = 2, \ldots, k$. We denote by $S(q, q')$ the set of paths from q to q', and $S(q, e, q')$ the set of paths from q to q' passing through arc e. The weight of a path, or a finite set of paths, can be defined through the $\otimes$ and $\oplus$ semiring operations. Since in this paper we are only concerned with the probability semiring in which weights represent probabilities, we express the weight operations directly using the product and sum rules of probability for the notational simplicity. As such, the weight of the path set S is given by $w[S] = \sum_{s \in S} \prod_{e \in s} w[e]$.

3.2. Representing Lattices with WFSTs

In speech recognition, WFSTs allow an elegant integration of knowledge sources such as the context dependency, the pronunciation lexicon, and the language model, in a precompiled and very efficient search network.

It is also natural to represent the lattice, generated via a recognition pass through the search network, with a transducer. Unlike the ordinary word lattices, the transducer-based lattice has a finer granularity at the HMM-state level. Specifically, for each arc, the input label is an HMM state, the output label is a word or a null symbol (denoted by ϵ), and the weight is the product of the acoustic and language model probabilities $w[e] = P_\Lambda(\boldsymbol{o}_e, e)$. The temporal boundaries of the arcs are indicated by assigning time instances to the state potentials τ.

3.3. Disambiguating Word Tokens in WFST-Based Lattices

To allow such lattices for a general discriminative training scheme, the main problem left is how to reproduce the alignment information between words and their constituent arcs, such that the substring-level error for, say, MPE/MWE can be calculated efficiently. Note that this alignment is not readily available in such a transducer, as ambiguities arise with ϵ-output arcs. First, they may be associated with different word identities from different paths; second, even in one single path, an ϵ-output arc may be a part of the previous output word or the next output word.

The second issue can be eliminated by deliberately generating the recognition lattices in two passes. First, we produce word-level lattices using an approximate lattice generation algorithm [12]. In a second pass, these word lattices are used as the "grammar", composed by the pronunciation lexicon L and the context-dependency transducer C, to generate the HMM-state lattices. Because the lexicon L always places the word output labels at the beginning arcs of pronunciations [7], and the composed search network in the second pass is not optimized, the resulting HMM-state lattice will retain the same property as the lexicon L, that is, word tokens always begin with (and end before) a non-ϵ-output arc. Moreover, if a state initiates a word token with a non-ϵ-output outgoing arc, its remaining outgoing arcs will also output non-ϵ labels. Hence, word initial states (also being word final states) can be considered as properties of the lattice, regardless of specific words.

It is not trivial to handle the alignment ambiguity due to multiple paths. First, we note that it suffices to distinguish the word tokens that disagree in the values required for calculating the substring-level errors, as discussed later in Section 3.4. These values, called context labels l, can include word identities, temporal boundaries, and even phone sequences. We propose a token disambiguation algorithm such that the arcs in the processed transducer have deterministic context labels. Every arc e is tagged with its context label l as (e, l). During a graph forward pass, the context labels are initialized to the word initial arcs, and passed on by the following states and arcs towards the word final arcs. If the context labels of the arcs reaching the same state q conflict, q will be replicated to account for different contexts. Each copy (q, l) of the state q will have the same

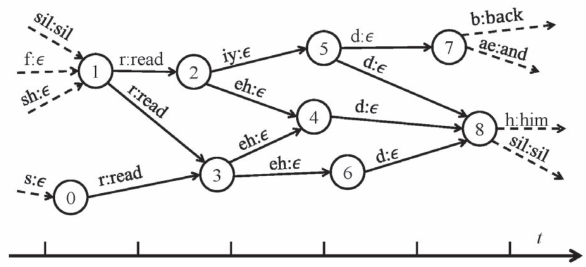

Fig. 1. Example of the subgraph of a word "read" in the converted lattice, where the context label is word identity. For convenience, the transducer shows a mapping from phones to words, rather than a mapping from HMM states to words actually used. Each arc is labeled with "input:output", with weights omitted. The arcs composing the subgraph are marked with solid lines.

outgoing arcs as q, but a subset of the incoming arcs of q that are tagged with context l. Word final states are replication-free, as they do not need to carry forward the context labels. Hence, the disambiguation procedure would not lead to a combinatorial explosion.

As a consequence, the converted lattice presents a two-layer hierarchy. It is analogous to a word lattice at a high level, and each word transition embodies an HMM-state subgraph for that word. The subgraphs, representing the words with distinct context information, do not intertwine with each other. A simple example of the subgraph is illustrated in Fig. 1.

3.4. Statistics Collection with WFSTs

One difficulty that arises in the model re-estimation of the MPE/MWE training is to determine $c(e)$, the average gain of all the paths passing through arc e. As shown in [3], $c(e)$ can be computed efficiently using the forward-backward algorithm, provided the gain function $v[s] = \mathcal{A}(s, s_{\text{ref}})$ of any path s can be decomposed into a sum of independent terms associated with its constituent arcs, $v[s] = \sum_{e \in s} v[e]$. We have

$$c(e) = \frac{\sum_{s \in S(I,e,F)} w[s]v[s]}{\sum_{s \in S(I,e,F)} w[s]} = \bar{\alpha}(p[e]) + v[e] + \bar{\beta}(n[e]) \quad (2)$$

where $\bar{\alpha}(q)$ and $\bar{\beta}(q)$ represent the average gain of partial paths leading to and leaving from state q, respectively, expressed as

$$\bar{\alpha}(q) = \frac{1}{\alpha(q)} \sum_{s \in S(I,q)} w[s]v[s]; \quad \bar{\beta}(q) = \frac{1}{\beta(q)} \sum_{s \in S(q,F)} w[s]v[s]$$

in which $\alpha(q) = \sum_{s \in S(I,q)} w[s]$ and $\beta(q) = \sum_{s \in S(q,F)} w[s]$ denote the total weights of the partial paths leading to and leaving from state q, respectively. The quantities $\alpha(q)$, $\beta(q)$, $\bar{\alpha}(q)$, and $\bar{\beta}(q)$ can be evaluated recursively in the forward-backward passes [3].

The final consideration is that the above formulae assume that the gain function is decomposed to the HMM-state level, which does not readily fit the substring-level (phone or word) gain function defined by MPE/MWE. Take MWE as an example, where the accuracy of a hypothesis word z is defined as [3]

$$v[z] = \max_{z' \in s_{\text{ref}}} \begin{cases} -1 + 2\text{overlap}(z, z') & \text{if } o[z] = o[z'] \\ -1 + \text{overlap}(z, z') & \text{otherwise} \end{cases} \quad (3)$$

where overlap(z, z') is defined as the time overlap between z and a reference word z' normalized by the length of z'.

To directly calculate the average word accuracy using the HMM-state lattices, we can re-decompose the word-level accuracies $v[z]$ into HMM-state levels $v[e]$. First, we perform the token disambiguation, as shown in Section 3.2, so that the accuracy of word tokens can be deterministically calculated. The context label is defined as the pair of the word identity and the word start time. With the tokens disambiguated and the context labels passed onto the word final arcs, we see that all required information to calculate the word accuracy (3) are contained in the word final arc e_f. They are the word label $o[z[e_f]]$, the word start time $\tau(p[\text{initial\_arc}[z[e_f]]])$, and the word end time $\tau(n[e_f])$, where $z[e]$ denotes a word token containing the arc e. Thus, to calculate the average word accuracy, equivalently the accuracy of an HMM-state arc e can be defined as

$$v[e] = \begin{cases} v[z[e]] & \text{if } e \text{ is the final arc of } z[e] \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

4. IMPLEMENTATION ISSUES

4.1. Substring Error

So far, we have discussed the issue of the statistics collection using the word-level error for MWE as an example. To account for MPE, whose gain function is defined similar to the one for MWE (3) except for using phone tokens, we only need to preprocess the lattice by changing its word output labels to phone labels. Then the transducer-based lattice at its high level becomes a phone lattice, and the foregoing procedure will work the same way for MPE. In fact, as the labels of the transducers are generic, the discriminative training algorithm can be easily configured to minimize the error rate of context-dependent (CD) phones, HMM states, and other substrings.

4.2. Synchronizing Word Labels with HMM States

There is a potential problem that may degrade the performance of the MWE training due to the use of the context-dependency transducer C. As in [7], the transducer C for a triphone context introduces a single-phone shift between a triphone and its context-independent counterpart to avoid the matching delay in the composed search network. This produces a recognition lattice where word output labels are marked on arcs one phone before they truly begin, and thus leading to a bias in calculating the word accuracy for MWE. Let us refer to the first HMM-state arc of a phones as a prime arc. Then to overcome the bias, we can push forward the output labels on the prime arcs to their following prime arcs. However, conflictions may arise if a prime arc takes in multiple preceding prime arcs that do not agree in their output labels. A procedure similar to the token disambiguation described in Section 3.2 is used to eliminate the conflictions. The context label of an arc is now the output label of its preceding prime arcs. Then the context labels can be deterministically passed towards and rewrite the output labels of the following prime arcs.

4.3. MCE Training

For MCE, the lattice of the hypothesis space should exclude the reference sequence. A direct implementation is by taking the difference between the recognition lattice and the reference sequence. Another approach is to keep using the recognition lattice as the denominator lattice, but subtract the contribution of the reference sequence from the denominator statistics in the statistics collection stage [5]. Both approaches are supported in the framework.

5. EXPERIMENTS AND RESULTS

The unified discriminative training algorithm has been evaluated on two tasks. The first is a speech recognition task on DARPA resource management (RM) database with medium-size vocabulary, and the second is a large-scale telephone speech recognition using the AT&T SCANMail [13] database.

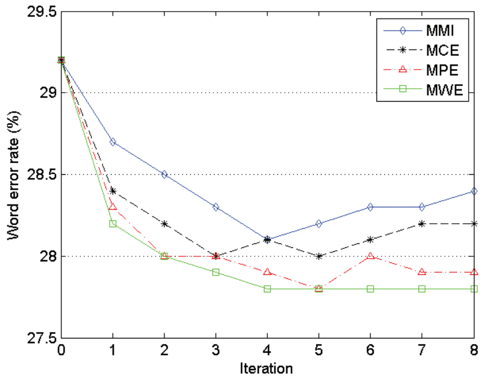

Fig. 2. WER (%) as a function of the iterations for different discriminative training criteria on the SCANMail database.

5.1. Experiments on Resource Management (RM) Database

Initial experiments were carried out on the 991-word DARPA RM database. The training data are the NIST/RM SI-109 training set consisting of 3,990 utterances from 109 native American speakers, and the test data consist of 1,200 utterances. Each feature frame is characterized by a 39-dimensional feature vector, including the 12-dimensional MFCCs plus log energy, and their first and second derivatives. The state-clustered cross-word triphone acoustic models are built with the AT&T FSM toolkit for speech recognition. This produces a total of 1,802 distinct states and 10,808 Gaussian components, with an average of 6 Gaussians per state. The ML baseline using the RM word-pair grammar yields word error rate (WER) of 4.2%.

Table 2. Comparison of discriminative training criteria on RM database.

	MLE	MMI	MCE	MPE	MWE
WER (%)	4.2	3.9	3.8	3.4	3.5
Rel. red. (%)	—	7.1	9.5	19.1	16.7

The recognition performance of four discriminative training criteria, MMI, MCE, MPE, and MWE, are given in Table 2. All of the criteria achieve significant improvements over the ML baseline. MPE yields the lowest WER (3.4%) among all the criteria, with 19.1% relative improvement over the ML baseline.

5.2. Experiments on SCANMail Database

The proposed framework has been evaluated on a large-scale telephone speech recognition task using the AT&T SCANMail [13] database. The task contains the voicemail messages received by 140 AT&T employees. The training and test sets contain 200 hours and 2 hours of speech, respectively. For each audio frame, 21 cepstral coefficients and energy are extracted from each audio frame, and mean normalized. Then 11 consecutive frames are projected onto a 60-dimensional feature space by Heteroscedastic Discriminant Analysis (HDA). The state-clustered cross-word triphone acoustic models are built. This produces a total of 6.9k distinct states and 161k Gaussian components, with an average of 24 Gaussians per state. The system uses a 31k word vocabulary, and a trigram language model trained on 700k words. The WER of the ML baseline is 29.2%.

Fig. 2 depicts the changes of WER over iterations for the four discriminative training criteria, where their best recognition performance are detailed in Table 3. MPE and MWE yield the best recognition performance, 4.8% relative error rate reduction compared with the ML baseline. Also, MWE behaves more stable than MPE in the course of the re-estimation procedure.

Table 3. Comparison of discriminative training criteria on the SCANMail database.

	MLE	MMI	MCE	MPE	MWE
WER (%)	29.2	28.1	28.0	27.8	27.8
Rel. red. (%)	—	3.8	4.1	4.8	4.8

6. CONCLUSION

We have described a general framework for implementing various discriminative training methods, such as MMI, MCE, and MPE/MWE. The lattice of the hypothesis space was represented with WFSTs, and the power of WFSTs was leveraged, leading to a general implementation of discriminative training. Experiments on the RM and AT&T SCANMail recognition tasks showed that all of the criteria yielded significant improvements over the ML baseline. Finally, it should be pointed out that the study mainly focuses on the effectiveness of using WFSTs for unified discriminative training, rather than a formal performance comparison. To compare and conclude these different training criteria in a rigorous manner necessitates a series of experiments with different settings and on various recognition tasks, which will be one of our future tasks.

7. REFERENCES

[1] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, 1986.

[2] B. H. Juang, W. Chou, and C. H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[3] D. Povey, *Discriminative training for large vocabulary speech recognition*, Ph.D. thesis, Univ. of Cambridge, 2004.

[4] W. Macherey, L. Haferkamp, R. Schlüter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Interspeech*, 2005.

[5] R. Schlüter, W. Macherey, B. Müller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Commun.*, vol. 34, pp. 287–310, 2001.

[6] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 14–36, Sept. 2008.

[7] M. Mohri, F. C. N. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," *Handbook on speech processing and speech communication, Part E: Speech recognition*, 2008.

[8] M. Mohri, F. C. N. Pereira, and M. Riley, *AT&T Finite-State Machine Library*, http://www2.research.att.com/˜fsmtools/fsm/.

[9] E. McDermott, T. J. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large-vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, 2007.

[10] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. Interspeech*, 2005.

[11] G. Heigold, R. Schluter, and H. Ney, "Modified MPE/MMI in a transducer-based framework," in *Proc. ICASSP*, 2009.

[12] A. Ljolje, F. Pereira, and M. Riley, "Efficient general lattice generation and rescoring," in *Proc. Eurospeech*, 1999.

[13] S. Whittaker, J. Hirschberg, B. Amento, L. Stark, M. Bacchiani, P. Isenhour, L. Stead, G. Zamchick, and A. Rosenberg, "SCANMail: a voicemail interface that makes speech browsable, readable and searchable," in *Proc. SIGCHI*, 2002, ACM Press.