An Iterative Approach To Decision Tree Training For Context Dependent Speech Synthesis

Xiayu Chen[1], *Yang Zhang*[2], *Mark Hasegawa-Johnson*[3]

Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign, IL

ahcxy2010@gmail.com, yzhan143@illinois.edu, jhasegaw@illinois.edu

Abstract

EDHMM with decision trees is a popular model for parametric speech synthesis. Traditional training procedure constructs the decision trees after observation probability densities have been optimized with the EM algorithm, assuming the state assignment probability does not change much during tree construction. This paper proposes an iterative algorithm that removes the assumption. In the new algorithm, the decision tree construction is incorporated into the EM iteration, with a safeguard procedure that ensures convergence. Evaluation on The Boston University Radio Speech corpus shows that the proposed algorithm can achieve a significantly better optimum in the training set than the original one, and that the advantage is well generalizable to the test set.

Index Terms: speech synthesis, speech clustering, EM algorithm, decision tree

1. Introduction

Explicit duration hidden Markov model (EDHMM) [1] is widely used in parametric speech synthesis and speech recognition systems [2, 3, 4]. Essentially EDHMM explicitly models the duration of each state in a Markov chain, as opposed to the inaccurate geometric duration inherent in a traditional HMM.

To account for context variations in speech parameters, context dependent modeling is widely applied. However there are usually not enough data to train a different model for every different context. To address this problem, decision tree is introduced [5, 6]. A decision tree is a binary tree whose leaf nodes are context-dependent clusters, with each observation directed to a leaf node from the root by answering the questions in non-terminal nodes all the way down. Decision trees are constructed essentially by maximizing the log likelihood of training data with some constraints preventing overfitting. Usually finding the globally optimal tree is intractable and in practice, people normally use a greedy algorithm [6]. The traditional training algorithm of EDHMM with decision tree relies on an important approximation assumption: state assignment probability is insensitive to the tree structure[6]. Therefore this algorithm trains all the HMM parameters without decision tree using EM algorithm, and calculates state assignment probability. Then, decision trees are constructed using a greedy algorithm, based on the state-assignment calculated.

We call this the Out-EM algorithm, as decision tree is built outside the EM iteration. We are interested in the impact of this approximation. Specifically, we would like to investigate if there's an efficient algorithm that removes this approximation, and evaluate what improvement can be achieved if this approximation is eliminated. This paper proposes an algorithm that incorporates tree construction in each EM iteration, which we call the In-EM algorithm. It turns out that, by adding a safeguard step, we can guarantee convergence of the new algorithm.

The following sections are arranged as follows: Section 2 states the model assumptions and notations; section 3 derives the Out-EM algorithm and discusses its convergence; section 4 displays experimental results comparing the two algorithms; conclusions are given in Section 5.

2. Model Assumptions and Notations

Before we derive the In-EM algorithm, it is necessary to briefly state the model assumptions and our notations. More detailed description of the model can be found in [2, 7].

2.1. EDHMM

As the common practice, we assume there is a hidden Markov chain $S_{1:T}$, where T is the total number of frames, $1:T$ denotes the subscript running from 1 to T, and each S_t is the phone state at frame t. We assume each phone consists of 3 states. In EDHMM, the duration of each state is explicitly modeled. For the i-th state, the state duration D_i is modeled as a Gaussian:

$$p_{D_i}(k) = \mathcal{N}(k; m_i, \sigma_i) \tag{1}$$

The transition probability from one state to another *different* state is given by the transition matrix A, and initial probability is given by the vector π:

$$\begin{aligned} A(i,j) &= \begin{cases} P(S_t = i | S_{t-1} = j, S_t \neq S_{t-1}) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \\ \pi(i) &= P(S_1 = i) \end{aligned} \tag{2}$$

The reason why self-transition is prohibited is that self-transition is already accounted for by the duration model.

The joint probability of the state chain $S_{1:T}$ can be expressed by first dividing the chain into segments of identical states $\{S_{T_n:T_{n+1}-1} = K_n, n = 1, \cdots, N\}$, where T_n is the boundary between the n-th and $(n+1)$-th segment, K_n is the state for the n-th segment, and N is total number of such segments; and then expressing the joint probability in terms of $T_{1:N}$ and $K_{1:N}$:

$$\begin{aligned} &p_{T_{1:N},K_{1:N}}(t_{1:N}, k_{1:N}) \\ &= \pi_{k_1} p_{D_{k_1}}(T_1) \prod_{n=2}^{N} A(k_n, k_{n-1}) p_{D_{k_n}}(T_n - T_{n-1}) \end{aligned} \tag{3}$$

2.2. Decision Tree

For each state, there are several observation densities organized by a decision tree, which is a binary tree containing a context question in each of its non-terminal nodes and a mixture in each of its leaf nodes. The decision trees of different states are distinct. Conditional on S_t, the density C_t is determined by starting from the root node of the tree corresponding to the state S_t, answering every question encountered, and moving down to the left or right descendant node depending on the answer until a leaf node is reached. Expressing it in a probabilistic way, we have

$$p_{C_t|S_t}(i,j) = \begin{cases} 1 & \text{if the } i\text{-th mixture of state } j \\ & \text{models the context at frame } t \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

2.3. Observation

Observations include MFCC and F0, which are assumed to share the same hidden states, but have different decision trees. However, to keep our illustration simple and straightforward, we will just assume that our observation contains only MFCC. The principle of both cases is the same.

Denote the MFCC at frame t, as well as its 1st and 2nd order difference, as the vector X_t. Conditional on S_t and C_t, X_t is assumed to be multivariate Gaussian.

$$p_{X_t|S_t,C_t}(x|i,j) = \mathcal{N}(x;\mu_{ij},\Sigma_{ij}) \quad (5)$$

2.4. Parameters

Equations (3), (4) and (5) best summarize the model. The model parameters can be divided into two sets. The first set, denoted as Θ_1, contains parameters of EDHMM and emission probabilities, which are differentiable and can be analytically optimized.

$$\Theta_1 = \bigcup (A, \pi, \{\mu_{ij}\}, \{\Sigma_{ij}\}) \quad (6)$$

The second set, denoted as Θ_2, contains hyper parameters that define the structure and questions of the decision trees. Usually, obtaining a global optimization over Θ_2 is hard. A common alternative is to apply the greedy algorithm to find a sub-optimal solution [6]. This is the crux of the training algorithm, and where the In-EM and Out-EM algorithms diverge.

To further simplify notation, we will denote X as a supervector containing $X_{1:T}$, called observations; Z as a supervector containing $S_{1:T}$ and $C_{1:T}$, called hidden variables.

3. Training Algorithms

In this section, we will briefly state the traditional Out-EM algorithm and derive the proposed In-EM algorithm for the model introduced in section 2.

3.1. The Target Function

For parameter estimation, log likelihood is a common target function, but to avoid overfitting of decision tree, we apply the minimum descriptive length criterion [6]. The target function is

$$L(\Theta_1,\Theta_2) = -\log\left(p_X(x|\Theta_1,\Theta_2)\right) + D(\Theta_2) \quad (7)$$

where $D(\Theta_2)$ restricts model complexity, and can be written as

$$D(\Theta_2) = \frac{\alpha_j}{2}\log T + \log J \quad (8)$$

where α_j is the number of free parameters of the model j; T is the number of training frames; and J is the number of models.

Equation (7) is maximized using EM. In order to define notation, let

$$\begin{aligned} &Q(\Theta_1,\Theta_2,R) \\ &= -\int_z R(z)\log\frac{p_{X,Z}\left(x,z|\Theta_1,\Theta_2\right)}{R\left(z\right)}\,dz + D(\Theta_2) \end{aligned} \quad (9)$$

Then we have, by Jensen's inequality,

$$L(\Theta_1,\Theta_2) \le Q(\Theta_1,\Theta_2,R) \quad (10)$$

with equality when Q is minimized over R

$$R(z) = p_{Z|X}(z|x;\Theta_1,\Theta_2) \quad (11)$$

which can be evaluated using the forward-backward algorithm [4].

EM algorithm essentially optimizes Q with respect to R (the E-step) and Θ's (the M-step) alternatively and iteratively until convergence. Since equality of equation (10) holds after each E-step, L will also be optimized.

3.2. The In-EM Algorithm

In-EM is an iterative algorithm. Denote the value of Θ_1, Θ_2 and R after the k-th iteration as $\theta_1^{(k)}$, $\theta_2^{(k)}$ and $r^{(k)}$ respectively. Also, let $R^*(\theta_1,\theta_2)$ be the optimized R when $\Theta_1 = \theta_1$ and $\Theta_2 = \theta_2$. Similarly we have $\Theta_1^*(\theta_2, r)$ and $\Theta_2^*(\theta_1, r)$. To make the principle clear, we'll keep our derivation abstract.

3.2.1. Main EM Procedure

In the E-step,

$$r^{(k+1)} = R^*(\theta_1^{(k)},\theta_2^{(k)}) = p_{Z|X}(\cdot|x;\theta_1^{(k)},\theta_2^{(k)}) \quad (12)$$

which can be calculated by the forward-backward algorithm.

In the M-step, first write the optimized Θ_1 as a function of Θ_2, so that $Q\left(\Theta_1^*\left(\Theta_2, r^{(k+1)}\right), \Theta_2, r^{(k+1)}\right)$ is reduced to a function of Θ_2. This is tractable because $\Theta_1^*\left(\Theta_2, r^{(k+1)}\right)$ can often be written in closed form. In our model, for example:

$$\begin{aligned} \mu_{ij}^* &= \frac{\sum_{t:j \text{ models the context of } x_t}\gamma_t(i)x_t}{\sum_t \gamma_t(i)} \\ \Sigma_{ij}^* &= \frac{\sum_{t:j \text{ models the context of } x_t}\gamma_t(i)(x_t-\mu_{ij}^*)(x_t-\mu_{ij}^*)^T}{\sum_t \gamma_t(i)} \end{aligned} \quad (13)$$

where

$$\gamma_t(i) = p_{S_t|X}(i|x;\theta_1^{(k)},\theta_2^{(k)}) \quad (14)$$

Then minimize $Q\left(\Theta_1^*\left(\Theta_2, r^{(k+1)}\right), \Theta_2, r^{(k+1)}\right)$ with respect to Θ_2, namely

$$\begin{aligned} \theta_2^{(k+1)} &= \underset{\Theta_2}{\operatorname{argmin}} Q\left(\Theta_1^*\left(\Theta_2, r^{(k+1)}\right), \Theta_2, r^{(k+1)}\right) \\ \theta_1^{(k+1)} &= \Theta_1^*\left(\theta_2^{(k+1)}, r^{(k+1)}\right) \end{aligned} \quad (15)$$

As discussed earlier, finding the exact optimum over Θ_2 is difficult, and a sub-optimum is found using a greedy algorithm.

3.2.2. Convergence

The fact that Θ_2 cannot be globally optimized leads to the problem of convergence. Because only when global optimum is found can the relation

$$\begin{aligned} Q\left(\theta_1^{(k+1)}, \theta_2^{(k+1)}, r_1^{(k+1)}\right) &\leq Q\left(\theta_1^{(k)}, \theta_2^{(k)}, r_1^{(k+1)}\right) \\ &\leq Q\left(\theta_1^{(k)}, \theta_2^{(k)}, r_1^{(k)}\right) \end{aligned} \tag{16}$$

be guaranteed, which leads to convergence. Otherwise, the first inequality does not necessarily hold, although the second still does.

To fix the problem, we add a safeguard step. Everytime an EM iteration is done, equation (16) is checked. If it does not hold, abandon the updated values and let

$$\begin{aligned} \theta_2^{(k+1)} &= \theta_2^{(k)} \\ \theta_1^{(k+1)} &= \Theta_1^*\left(\theta_2^{(k+1)}, r^{(k+1)}\right) \end{aligned} \tag{17}$$

Since the optimization over Θ_1 and R is global, (17) guarantees (16). However, it should be noted that In-EM algorithm can only guarantee a local optimum.

3.3. The Out-EM Algorithm

For comparison, Out-EM is briefly stated in our notation:

1) Fix Θ_2 as some simple form $\bar{\theta}_2$, e.g. only depth-1 trees;

2) Optimize $Q(\Theta_1, \bar{\theta}_2, R)$ over R and Θ_1 using EM algorithm:

$$\begin{aligned} r^* &= R^*(\theta_1', \bar{\theta}_2) \\ \theta_1' &= \Theta_1^*(\bar{\theta}_2, r^*) \end{aligned} \tag{18}$$

3) Optimize $Q(\Theta_1^*(\Theta_2, r^*), \Theta_2, r^*)$ over Θ_2 using the greedy algorithm.

It can be immediately noted that the solution is not even a local optimum, because r^* is an 'old' optimum at $\theta_1', \bar{\theta}_2$. An important assumption for this algorithm to work, is that R^* is insensitive to Θ_2. It is worth noticing that there is still chance that Out-EM finds better estimates than In-EM, if R^* is sufficiently insensitive and In-EM is trapped in a really bad local optimum. We'll evaluate this chance in the next section.

4. Experiment and analysis

4.1. Experiment configuration

In this section, a set of experiments comparing In-EM and Out-EM algorithms are described and analyzed. These experiments are conducted on the Boston Radio Speech Corpus [8] with phone and F0 labeling. To facilitate preliminary experiments, silent phone (including pause, silence and closure of plosives) and phones with too few data are discarded. The dataset thus includes 37 phone types with up to 33511 phone tokens.

The experiments are divided into two parts. The first part, where all data are incorporated in the training set, will demonstrate that In-EM is able to better optimize the target function, and evaluate approximation error of Out-EM. The second part, where frames are split into equally-sized training set and test set, will demonstrate that the superiority of In-EM is generalizable to data outside the training set. These two parts will be discussed in detail in subsequent subsections respectively.

4.2. Training set analysis

As is mentioned, In-EM algorithm is easily trapped in local optima. For this reason, In-EM algorithm is run for 20 times with different random initializations. We expect to see that the In-EM algorithm has better performance than Out-EM statistically.

To study the performance on MFCC and F0 separately, trees of F0 are controlled when trees of MFCC are trained with the two distinct algorithms, and vice versa. Table 1 shows the percentage of instances where In-EM achieves better target function than Out-EM for MFCC (with trees of F0 controlled) and F0 (with trees of MFCC controlled). To see the relationship of tree scale and performance, the number of leaf nodes for a decision tree is constructed in Out-EM algorithm is also presented. The phones listed in the table are selected randomly.

Table 1 shows that in most cases, the In-EM algorithm can better minimize the target function than Out-EM. Furthermore, it can also be observed that this advantage of In-EM algorithm is more evident when the decision tree is large. For instance, phone /oy/, the worst case being displayed for MFCC, has the fewest number of nodes; while phones where In-EM for MFCC has 100% outperformance have on average 10 nodes for each state. This is because when the data size is too small, the trees constructed by both algorithms are simple, with limited difference in their structure. Conversely for a properly sized tree, not only the question selected makes a difference, but the position of each question within the tree also matters, which leads to In-EM consistently outperforming Out-EM. It is also presented that the advantage of In-EM for F0 is more distinct with voiced phones. This can similarly be explained by the scale of decision tree. For unvoiced phones, data are insufficient. Some decision trees even contain only one node. In this case, In-EM is less likely to find a better structured decision tree.

Table 1: *Percentage of experiments (P) where In-EM description length is better than Out-EM for* MFCC *and* F0. *U stands for unvoiced, V for voiced phone, LN is the number of leaf nodes of decision trees of the 3 states of each phone*

Phone	U/V	P_{MFCC}	LN of MFCC	P_{F0}	LN of F0
/p/	U	90%	(8,3,5)	80%	(1,1,4)
/k/	U	100%	(8,13,10)	30%	(1,1,11)
/s/	U	95%	(6,8,14)	55%	(12,1,12)
/l/	V	100%	(10,6,11)	100%	(12,4,8)
/ng/	V	95%	(2,2,5)	100%	(2,2,9)
/ih/	V	100%	(10,13,13)	65%	(13,3,20)
/ao/	V	100%	(12,7,5)	100%	(5,5,4)
/oy/	V	70%	(2,1,2)	15%	(1,1,2)
Total	-	90.6%	-	70.0%	-

Examples of decision trees trained by both algorithms are presented in Fig. 1. As shown, the questions selected by In-EM and Out-EM tend to be quite similar, but their ordering within the tree may differ considerably.

Typical convergence curves are presented in Fig. 2. The convergence patterns of In and Out EM are very different. Since parameters for decision tree structure are discrete, In-EM description length generally has a small but evident decrease once every few iterations. Those decreases indicate that the structure of the decision tree changes. In contrast, the curve of Out-EM training decreases gradually in the first phase and then

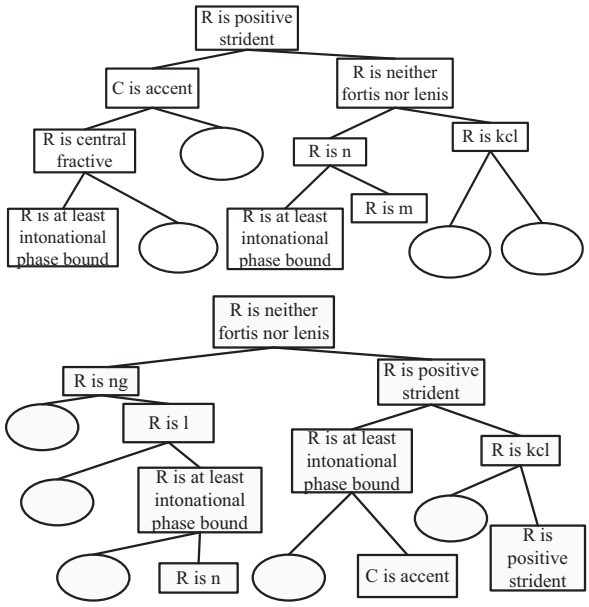


Figure 1: *Example decision trees. Top panel:* MFCC *tree for phone* $/ae/$ *state 1 trained Out-EM. Bottom panel:* MFCC *tree for phone* $/ae/$ *state 1 trained In-EM. R stands for right, L stands for left and C stands for current. Expression-like questions are on prosodic break level. Nodes with depth greater than 3 are omitted.*

undergoes a large decrease, because that is when decision trees are constructed. After that, the value of DL for Out-EM training stays stable. In Fig. 2, it is obvious that In-EM outperforms Out-EM in most cases. Poor equilibrium of In-EM is usually accompanied by slow convergence.

4.3. Test set analysis

In this subsection, the data of each phone are divided into two sets with almost equal numbers of frames. The first set is for training, while the second is for testing. In this experiment, training is performed on the training set, and after that, the likelihood of test data are calculated as the performance metric.

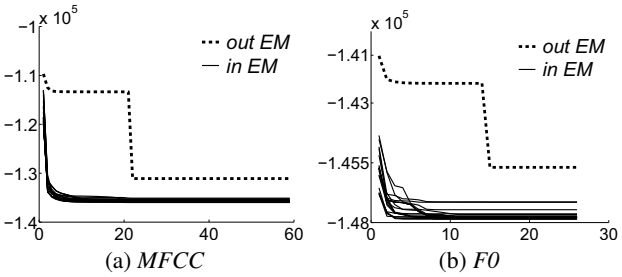
To formally compare the performance on test data, we perform a hypothesis test where the null hypothesis H_0 is that the joint likelihood of the model trained with In-EM outperforms that of Out-EM in 50% of all trials,

$$H_0 : P\left(P_X\left(x|\Theta_{\text{In-EM}}\right) > P_X\left(x|\Theta_{\text{Out-EM}}\right)\right) = 0.5 \quad (19)$$

where x is the observed feature vector, i.e., MFCC, of segments in the testing data, $P_X\left(x|\Theta_{\text{In-EM}}\right)$ and $P_X\left(x|\Theta_{\text{Out-EM}}\right)$ are posterior probability of x calculated with parameters of Out-EM training $\Theta_{\text{Out-EM}}$ and In-EM training $\Theta_{\text{In-EM}}$. If H_0 is true, then the probability that the likelihood of the model trained by In-EM is greater than that by Out-EM in at least N out of T segments follows the binomial tail distribution

$$P(N, T|H_0) = \sum_{n \geq N} \binom{T}{N} \left(\frac{1}{2}\right)^T \quad (20)$$

We perform the hypothesis test under 90% confidence level,



(a) *MFCC* (b) *F0*

Figure 2: *Curve of value of Q-function changing with times of iteration, of training set of phone* $/ao/$*, for* MFCC *and* F0 *training. Solid lines for 20 randomly initialized In-EM training , dotted line for Out-EM training. The curve begins after the first iteration.*

which means if $P(N, T|H_0) < 0.1$, then the null hypothesis is rejected.

Similar to the previous subsection, the In-EM procedure for each phone is also carried out 20 times with random initialization. However, only the decision tree with smallest description length is selected to compare to the decision tree trained with Out-EM for the same phone. The number of segments whose joint likelihood of In-EM are better than that of Out-EM is counted and summed to calculate $P(N, T|H_0)$. Table 2 shows the result of some phones and the statistical result of all phones. The phones being displayed were selected uniformly at random from the set of all phone labels.

Table 2: *Percentage of experiments whose In-EM test likelihood is better than Out-EM for* MFCC *and* F0, P *stands for* $P(N, T|H_0)$

phon.	T	$\frac{N_{\text{MFCC}}}{T}$	P_{MFCC}	$\frac{N_{\text{F0}}}{T}$	P_{F0}
$/k/$	534	57.39%	3×10^{-4}	61.05%	$< 10^{-10}$
$/s/$	1456	76.58%	$< 10^{-10}$	59.55%	$< 10^{-10}$
$/th/$	143	69.23%	$< 10^{-10}$	53.85%	0.1786
$/m/$	608	63.65%	$< 10^{-10}$	66.61%	$< 10^{-10}$
$/eh/$	664	51.20%	0.2681	64.31%	$< 10^{-10}$
$/oy/$	23	26.09%	0.9947	78.26%	0.005
$/er/$	129	64.34%	5×10^{-4}	77.52%	$< 10^{-10}$
Total	18072	60.69%	$< 10^{-10}$	60.14%	$< 10^{-10}$

The statistical result of all phones for both the MFCC and F0 training show that $P(N|T, H_0) \ll 0.1$, thus the null hypothesis is rejected. We may say that In-EM trains a significantly better model.

5. Conclusion

In this paper, a new procedure of decision tree training in HMM-based speech synthesis is proposed, and is proven to converge. Though trees trained In or Out EM are disparately shaped, the questions involved for a given state of one phone are relatively stable. By the differently shaped decision trees, the In-EM decision tree can converge to a smaller description length of the training data in most cases. The data likelihoods computed by In-EM also generalize to test data significantly better than those of Out-EM. Therefore the proposed procedure is proved to be a better procedure for state clustering.

6. References

[1] J. D. Ferguson, "Variable duration models for speech" in *Proc. Symp. Application of Hidden Markov Models to Text and Speech*, 1980, pp. 143-179.

[2] Heiga Zen, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura, "A hidden semi-markov model-based speech synthesis system," *IEICE Transactions. on Information and Systems E Series D*, vol. 90, no. 5, pp. 825, 2007.

[3] Mari Ostendorf, Vassilios V. Digalakis, and Owen A Kimball,"From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 5, pp. 360-378, 1996.

[4] Ken Chen, Mark Hasegawa-johnson, Aaron Cohen, Sarah Borys, Sung-Suk Kim, Jennifer Cole, and Jeung-Yoon Choi "Prosody dependent speech recognition on radio news corpus of American English,"*Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 13, no. 6, pp. 232-245, 2005.

[5] Steve J. Young, Julian J. Odell, and Philip C. Woodland, "Treebased state tying for high accuracy acoustic modelling," in *Proceedings of the Workshop on Human Language Technology*, Stroudsburg, PA, USA, 1994, HLT 94, pp. 307-312, Association for Computational Linguistics.

[6] Julian J. Odell, "The use of context in large vocabulary speech recognition," Ph.D. dissertation, Queens' College, Univ. of Cambridge, Cambridge, UK, 1995.

[7] Takayoshi Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems," Ph.D. dissertation, Dept. Elec. Comp. Eng., Nagoya Inst. of Technology, Nagoya, Japan, 2002.

[8] Mari Ostendorf, Patti J. Price, Stefanie Shattuck-Hufnagel, "The Boston University radio news corpus," *Linguistic Data Consortium*, pp. 1-19, 1995.