Context-Sensitive Evaluation and Correction of Phone Recognition Output

Michael Levit, Hiyan Alshawi, Allen Gorin, Elmar Nöth

AT&T Labs-Research
180 Park Ave., Florham Park, New Jersey 07932, USA
{levit,hiyan,algor}@research.att.com, noeth@informatik.uni-erlangen.de

Abstract

In speech and language processing, information about the errors made by a learning system is commonly used to assess and improve its performance. Because of high computational complexity, the context of the errors is usually either ignored, or exploited in a simplistic form. The complexity becomes tractable, however, for phone recognition because of the small lexicon. For phone-based systems, an exhaustive modeling of local context is possible. Furthermore, recent research studies have shown phone recognition to be useful for several spoken language processing tasks. In this paper, we present a mechanism which learns patterns of context-sensitive errors from ASR-output aligned with the "true" phone transcriptions. We also show how this information, encoded as a context-sensitive weighted transducer, can provide a modest improvement to phone recognition accuracy even when no transcriptions are available for the domain of interest.

1. Introduction

Understanding the pattern of errors typical for a learning system provides insight into the learning process and can improve its results [3, 7, 8]. In [3] a *transformation-based error-driven learning* method was proposed that utilizes a confusion matrix to perform tagging correction by learning a set of correction rules. A similar approach was used in [7] to disambiguate the output of a word recognizer. However, this method did not make use of context and the size of ambiguity sets was restricted to two words. Even though many error correction mechanisms do exploit context [3, 5], they don't estimate all context-sensitive confusion probabilities, whose number for a lexicon of size N rises to N^4 even if only the immediate left and right contexts are considered. Instead they encode the context with a small number of features [5]. The ambiguity sets are also typically downsized drastically. Error correction with exhaustive local context modeling becomes manageable, however, when dealing with phone-based systems, where the lexicon size is $\sim$50 rather than thousands.

Recent work has demonstrated that phones are a serious alternative to words in certain spoken language processing tasks [10, 2]. In [2] phone recognition followed by topic classification has been shown to produce classification rates comparable to the traditional word-based approach; in our latest experiments we also obtained similar results by using approximate matching of *acoustic morphemes*, a technique first introduced in [6]. As a result, the reliability of a phone recognizer becomes important. In this paper we present a method for modeling the behavior of the phone recognizer under the influence of local string context and for using this information to improve phone recognition accuracy.

The application vehicle we chose for this work is telephony speech services, where speech recognition components must be adapted quickly to new or altered services, so it is desirable for the developer not to rely on the availability of transcriptions for training new tasks. In [2] a method for unsupervised training of a phone recognizer was introduced which iteratively recognizes the speech and re-estimates the language model. Here we focus on the situation where transcriptions for some other domain are available. We show how this additional data can be employed to improve phone error rate in the domain of interest. To achieve an improvement, we post-correct ASR-output, by passing it to a transducer which performs a number of context-sensitive phone corrections (substitutions, deletions, insertions).

The results of a cross-domain evaluation of this algorithm indicate a 4% relative reduction in phone error rate, whereby the effect is most prominent in the insertion rate which decreases by 14%.

2. Transductional model of ASR

Our goal is to develop a mechanism for describing ASR behavior and improving its performance on speech data. Given a spoken utterance S, the phone recognizer outputs a phone sequence S^{asr}. Suppose now that there is an oracle telling us what the ideal (undistorted) representation of S should be; we call this phone string S^{true}. In this case we can describe ASR behavior in terms of a transformation between S^{true} and S^{asr}:

$$\mathrm{ASR}(S^{\mathrm{true}}) = S^{\mathrm{asr}}. \tag{1}$$

On the other hand we might consider the inverse task of recovering the "true" phone string S^{true} based on the recognized string S^{asr}. In this case the transformation will be:

$$\mathrm{ASR}^{-1}(S^{\mathrm{asr}}) = S^{\mathrm{true}}. \tag{2}$$

Since the major part of the discussion in this section doesn't depend on which of these two ways is explored, we will use notation *input string* S^{I} for the argument of the transformation and *output string* S^{O} for its result, instead of S^{true} and S^{asr}.

Typically we distinguish among four basic types of phone transformations: identity, substitution, deletion and insertion. We call these transformations *phone mappings*.

Suppose, we have reached position i in the string S^{I} and produced $j-1$ phones in string S^{O}. Now the following phone mappings are possible:

1. **identity:** take phone S^{I}_i as the next phone in S^{O} ($S^{\mathrm{O}}_j := S^{\mathrm{I}}_i$) and advance by one position in both strings;
2. **substitution:** same as identity, but instead of S^{I}_i emit some other phone;
3. **deletion:** advance in S^{I} without emitting anything to S^{O};
4. **insertion:** emit some phone to S^{O} and advance in this string, but not in S^{I}.

If we introduce the empty symbol ε, all four mappings can be written in the same way: $a \rightsquigarrow b$, where a (the *left side* of the mapping) and b (its *right side*) may not be ε at the same time.

One way to characterize ASR behavior is to estimate the probabilities of such mappings. Each phone mapping produces at most one phone in the output string S^{O}, leaving the previously generated phones unchanged, but in the general case they depend on the entire input string[1] S^{I}, which we call the *mapping context*, so that we can write the mapping in context as $a \rightsquigarrow b/S^{\mathrm{I}}$. If we ignore the influence of context completely, then the result of estimation will be a set of probabilities $P(a \rightsquigarrow b|a)$ for each pair of phones (a, b). This approximation assumes that the probability of mapping a into b remains constant no matter where in the string a occurs. Estimation of the well known confusion matrix relies on this kind of approximation. However, this type of description is not adequate, since recognition of a particular phone is significantly affected by its local neighborhood in the acoustic stream [9].

This leads us to a less rough approximation with four phones that have impact on the mapping probability: the input and output phones of the mapping and also the two symbols adjacent to the input phone in S^{I}. Such mappings we will denote $a \rightsquigarrow b/c\_d$, with c as the left context of a in S^{I} and d its right context.

Now, if there are N phones in our lexicon, the set of estimated probabilities will comprise $O(N^4)$ entries. In fact, if no insertions were possible, we would only have $O(N^3)$ distributions, with stochastic conditions:

$$\sum_x P(a \rightsquigarrow x/c\_d|cad) = 1.0,\ \forall c, a, d. \tag{3}$$

The presence of insertions makes the situation a little more complicated. Let the input string be $S^{\mathrm{I}} = a_0 a_1 \ldots a_{T-1} a_T$ and the current position in this string t. The next mapping will be either substitution or deletion of a_t in context $a_{t-1}\_a_{t+1}$ or insertion in context $a_t\_a_{t+1}$. Thus, at each point of time we not only have competing mappings in the same context, but also competing contexts:

$$\begin{aligned} \sum_x P(a_t \rightsquigarrow x/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \quad &+ \\ \sum_x P(\varepsilon \rightsquigarrow x/a_t\_a_{t+1}|a_{t-1}a_t a_{t+1}) \quad &= \quad 1.0. \end{aligned} \tag{4}$$

The problem with this formula is that we wish to avoid maintaining statistics for insertions conditioned on two phones in the left context (as opposed to only one in $P(\varepsilon \rightsquigarrow x/a_t\_a_{t+1}|a_t a_{t+1})$).

Let $\bar{P}(\mathrm{sd}(c)/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1})$ denote the probability of doing insertion in context $a_t\_a_{t+1}$ (i.e. not a substitution or deletion of a_t in context $a_{t-1}\_a_{t+1}$), given $a_{t-1}a_t a_{t+1}$, and $\bar{P}(\mathrm{ins}\,/a_t\_a_{t+1}|a_t a_{t+1})$ the probability of doing substitutions or deletions of a_t in context y_a_{t+1} with an arbitrary y (i.e. not an insertion in context $a_t\_a_{t+1}$), given $a_t a_{t+1}$. Then, we can split (4) in two:

$$\begin{aligned} \sum_x P(a_t \rightsquigarrow x/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \quad &+ \\ \bar{P}(\mathrm{sd}(a_t)/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \quad &= \quad 1.0; \end{aligned} \tag{5}$$

$$\sum_x P(\varepsilon \rightsquigarrow x/a_t\_a_{t+1}|a_t a_{t+1}) + \bar{P}(\mathrm{ins}\,/a_t\_a_{t+1}|a_t a_{t+1}) = 1.0. \tag{6}$$

We will see in the next section how the probabilities participating in these formulae can be estimated.

3. Estimation and encoding of phone mapping probabilities

3.1. Training phone mapping probabilities

In this section we show how to estimate the probabilities of phone mappings in context so that the probability of a transformation of the input corpus into the output corpus (both sequences of phone strings) is maximized. We use the EM algorithm to estimate the probabilities of phone mappings. During the expectation step we update counters γ of occurrences of phone mappings in contexts. The outline of the expectation step is presented below[2]:

```
FORALL $a, b, c, d :\ \neg(a = b = \varepsilon)$
    $\gamma(a \rightsquigarrow b/c\_d|cad) \ :=\ 0$
    IF    $a \neq \varepsilon$
    THEN  $\bar{\gamma}(\mathrm{sd}(a)/c\_d|cad) \ :=\ 0$
    ELSE  $\bar{\gamma}(\mathrm{ins}\,/c\_d|cd) \ :=\ 0$
FOR $t = 0 \ldots T,\ v = 0 \ldots V$
    $\gamma_s \ :=\ \alpha_{t-1,v-1} P(a_t \rightsquigarrow b_v/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1})\beta_{t,v}/\alpha_{T,V}$
    $\gamma_d \ :=\ \alpha_{t-1,v} P(a_t \rightsquigarrow \varepsilon/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1})\beta_{t,v}/\alpha_{T,V}$
    $\gamma_i \ :=\ \alpha_{t,v-1} P(\varepsilon \rightsquigarrow b_v/a_t\_a_{t+1}|a_t a_{t+1})\beta_{t,v}/\alpha_{T,V}$
    $\gamma(a_t \rightsquigarrow b_v/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \ \mathrel{+}=\ \gamma_s$
    $\gamma(a_t \rightsquigarrow \varepsilon/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \ \mathrel{+}=\ \gamma_d$
    $\gamma(\varepsilon \rightsquigarrow b_v/a_t\_a_{t+1}|a_t a_{t+1}) \ \mathrel{+}=\ \gamma_i$
    $\bar{\gamma}(\mathrm{sd}(a_t)/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \ \mathrel{+}=\ \gamma_i$
    $\bar{\gamma}(\mathrm{ins}\,/a_t\_a_{t+1}|a_t a_{t+1}) \ \mathrel{+}=\ \gamma_s + \gamma_d$
```

where forward and backward probabilities $\alpha_{t,v}$ and $\beta_{t,v}$ are obtained for each pair (t, v) successively. For instance, the iterative formula for αs is:

$$\begin{aligned} \alpha_{t,v} \quad \mathrel{+}= \quad & \alpha_{t-1,v-1} * P(a_t \rightsquigarrow b_v/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \\ + \quad & \alpha_{t-1,v} * P(a_t \rightsquigarrow \varepsilon/a_{t-1}\_a_{t+1}|a_{t-1}a_t a_{t+1}) \\ + \quad & \alpha_{t,v-1} * P(\varepsilon \rightsquigarrow b_v/a_t\_a_{t+1}|a_t a_{t+1}). \end{aligned} \tag{7}$$

During the maximization step, probabilities of all phone mappings are re-estimated:

$$\begin{aligned} P(a \rightsquigarrow b/c\_d|cad) \ &:=\ \frac{\gamma(a \rightsquigarrow b/c\_d|cad)}{\sum_x \gamma(a \rightsquigarrow x/c\_d|cad) + \bar{\gamma}(\mathrm{sd}(a)/c\_d|cad)}; \\ P(\varepsilon \rightsquigarrow b/c\_d|cd) \ &:=\ \frac{\gamma(\varepsilon \rightsquigarrow b/c\_d|cd)}{\sum_x \gamma(\varepsilon \rightsquigarrow x/c\_d|cd) + \bar{\gamma}(\mathrm{ins}\,/c\_d|cd)}. \end{aligned} \tag{8}$$

3.2. Specificity versus robustness

Given the number of phones in our dictionary N, we have to estimate N^3 different context-dependent probability distributions (or $O(N^4)$ context-dependent probabilities). Even with a moderate number of phones used (in our experiment $N = 43$), the amount of data practically available is not sufficient to estimate all context-dependent probabilities reliably. To alleviate this problem, we interpolate among contexts with different degrees of specificity. Consider phone mappings: $a \rightsquigarrow x/c\_d$; for each x its probability can be computed as a linear combination:

$$\begin{aligned} P(a \rightsquigarrow x/c\_d) = w_f P(a \rightsquigarrow x/c\_d) + w_l P(a \rightsquigarrow x/c\_*) \quad &+ \\ w_r P(a \rightsquigarrow x/*\_d) + w_n P(a \rightsquigarrow x/*\_*) + w_z \frac{1}{N}&, \end{aligned} \tag{9}$$

with $w_f + w_l + w_r + w_n + w_z = 1.0$ and wildcard “*” standing for any symbol. In this case, before interpolation can be done, four optimization processes must be performed in parallel. In our experiments we used the following weights: $w_f = 0.5,\ w_l = w_r = 0.2,\ w_n = 0.09,\ w_z = 0.01$. Also, a subsequent probability normalization is required to make sure the stochastic conditions are not violated. Additionally, we do not allow the probabilities of any mappings to fall below a certain small threshold.

[1] An obvious extension is to make the mappings dependable on at least some part of S^{O} as well.

[2] We assume $a_t = \varepsilon\ \forall t < 0, t > T$.

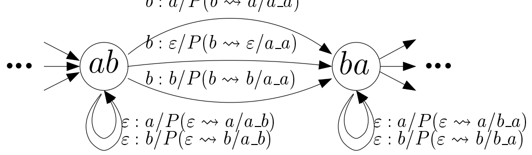
$b : a/P(b \rightsquigarrow a/a\_a)$
$b : \varepsilon/P(b \rightsquigarrow \varepsilon/a\_a)$
$b : b/P(b \rightsquigarrow b/a\_a)$
...
ab
ba
...
$\varepsilon : a/P(\varepsilon \rightsquigarrow a/a\_b)$
$\varepsilon : b/P(\varepsilon \rightsquigarrow b/a\_b)$
$\varepsilon : a/P(\varepsilon \rightsquigarrow a/b\_a)$
$\varepsilon : b/P(\varepsilon \rightsquigarrow b/b\_a)$

Figure 1: *Simple coding of phone mapping probabilities as a weighted FSM; fragment from an FSM for a two symbol alphabet.*

3.3. Encoding the probabilities as a transducer

After running the EM algorithm, the estimated probabilities of the phone mappings in context are encoded as a transducer. Each state of this transducer is marked by a pair of phones that can follow each other in the corpus. Arcs connecting two states ab and bc sharing one middle phone b perform substitutions of b in the context a_c, and loops from the state ab perform insertions in context a_b (see Figure 1). In fact, in our experiments we use a somewhat more complex encoding which allows an insertion in context to be more probable than its absence.

Depending on whether the transducer models ASR or ASR^{-1} we will call it the *Distortion* or the *Correction* Transducer.

4. Improving Phone Accuracy

In this section we show how the Correction Transducer obtained in the way described above can be used to improve the phone recognition accuracy. Suppose that we have two corpora from different domains and that for one corpus transcriptions are available.

Since the task of manually producing phone transcriptions is hardly feasible for large amounts of speech data, we resort to word transcriptions created by human labelers and transduce them to the phone level by taking the most probable dictionary pronunciation for each word[3]. This is certainly different from employing the ideal phone transcriptions but two factors justify our choice: even human labelers have difficulty reaching a consensus when asked to produce the phone transcription of a speech signal [4]; in our experiments the results of utterance classification on phone transcriptions produced as just noted significantly outperformed those obtained with ASR.

The training scheme along with the testing strategy (or application at run-time) are presented in Fig. 2.

During the first step, the phone recognizer is trained on the audio data from the training corpus (domain of interest). Then, we recognize the training corpus from another domain, for which manual transcriptions are available. Using the pairs recognized-utterance/transcribed-utterance for the second domain, we train the Correction Transducer $\mathcal{F}$ representing transformation ASR^{-1}. At run-time we recognize the incoming utterance with the same recognizer obtaining phone string S, and then apply $\mathcal{F}$ as follows:

$$\hat{\mathcal{S}} = \mathcal{S} \circ \mathcal{F}, \qquad (10)$$

$\mathcal{S}$ and $\hat{\mathcal{S}}$ being linear FSM representations of S and its corrected version $\hat{S}$.

5. Experiments

For our experiments we used data from two telephony domains:

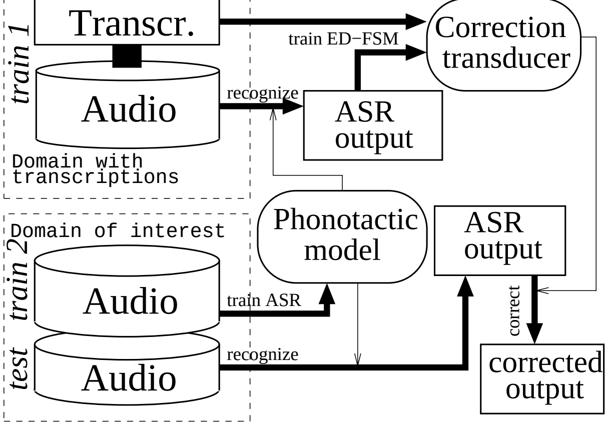

Figure 2: *Training ASR and Correction Transducer and their role in phone recognition.*

1. **Domain of interest:** a corpus of prescription related requests to a pharmaceutical company made over the phone: P-train (∼15K), P-test (5K). The latter provided with transcriptions to measure the phone accuracy (see below).
2. **Transcription Domain:** HMIHY, a collection of utterances made by callers to the AT&T Customer Service number. Two subsets were distinguished: H-train (∼25K utterances) and H-test (∼3K); transcriptions were available for both;

We used P-train to train a 5-gram phonotactic model for phone recognition as described in [2]. After that, H-train utterances were used to train the Correction/Distortion Transducers.

An informative intrinsic criterion of the goodness of phone mapping probability estimators is the changes of the final forward probability $\alpha_{T,V}$ over iterations. Figure 3 shows how the average length-normalized probability of transforming ASR-output into transcriptions changes for the training (H-train) and test (H-test) corpora. We see that, when context is taken into account, we are able to model the ASR-behavior much more precisely than when phone mappings are considered without regard to context[4].

5.1. Diagnostic tool

The Distortion Transducer estimated as described in Section 3.1 can be used to illustrate the behavior of the phone recognizer and exemplify typical mistakes it makes.

In Table 1 we present some of the most probable non-identity phone mappings accounted for by this transducer (in ARPABET symbols) :

phone mapping	probability	example
$ax \to ey/uw\_hh$	0.88	speak to a human
$ax \to ae/ng\_d$	0.76	mailing address
$t \to d/r\_ih$	0.59	started
$k \to \varepsilon/s\_t$	0.86	asked
$ih \to \varepsilon/r\_d$	0.78	hundred

Table 1: *Some of the most probable (and frequent) phone transformations from transcriptions to ASR-output.*

[3] In the reported experiments we use a TTS system to produce phone transcriptions from word transcriptions.

[4] The test corpus is more challenging for ASR because it contains only initial dialog utterances; these are much longer on average than the training corpus utterances which include many instances of short utterances like “Yes” and “No”.

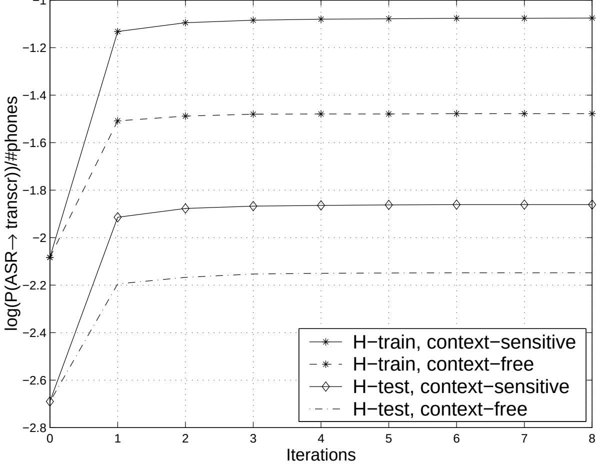

Figure 3: *Average length-normalized log-probability of the transformation Transcription→ASR; cases of context-sensitive and context-independent optimization.*

We see that the ASR tends to confuse phonetically similar phones like t and d, ax and ae, and omit several other phones. However this misrecognition is not persistent throughout different contexts, but is often restricted to the cases of coarticulation phenomena like undershoot, and is not present in the phonetically simple cases. For instance, in context iy_aa (as in "AT&T-card") k is recognized 100 percent of the time. For future studies, it may also be interesting to compare the results obtained to recognition mistakes made by humans [1].

5.2. Correcting ASR-output

Similar statistics can be collected when training the Correction Transducer to reflect the inverse transformation ASR^{-1}. Examples of probable context-sensitive corrections are: $th \rightarrow \varepsilon/s\_p$ (mostly noise removal) and $sh \rightarrow s/eh\_ch$ (as in "question"). In the latter case it can be argued that the correction replaces one legitimate pronunciation by another, simply standardizing it. This normalization effect however is also of use; for example, it simplifies the downstream classification task.

To assess the effect of correction we computed the phone accuracy of P-test recognized with ASR trained on P-train before and after the composition with the Correction Transducer $\mathcal{F}$ trained on H-train.

criterion	before	correction	correction
	correction	w. context	w/o context
substitutions	13.8%	13.2%	13.8%
deletions	8.7%	9.0%	8.7%
insertions	6.3%	**5.4%**	6.3%
phone error rate	28.8%	**27.6%**	28.7%

Table 2: *Phone error rates before and after context-sensitive and context-independent corrections.*

From Table 2 we see that context-sensitive correction by composition has a positive impact on the substitution and insertion rates. The overall phone error rate drops by 1.2 percent points (a 4% relative improvement), whereas the impact of context-independent correction is neglectable. However, this correction mechanism fails to compensate for deletions made by the ASR. This may be explained by the fact that our phone recognizer behaves in a "cautious" manner, skipping large chunks of the audio signal (sometimes as large as the entire utterance) unless a reliable recognition is possible. Another explanation is that the probabilities of deletions and substitutions are conditioned on three phones, whereas the insertion probabilities only on two (see Eq. (5), (6)), which makes the estimation less specific.

Finally, to answer the question whether we would have been able to use the available transcriptions more directly, we trained the ASR phonotactic model on the same HMIHY-transcriptions that were used to train the Correction Transducer and recognized the P-test data with it. This resulted in a phone error rate of 29.8% which is significantly higher than the one achieved using the strategy outlined in Figure 2.

6. Conclusions

We presented a method for context-sensitive evaluation of the performance of a phone recognizer. Unlike simple confusion matrices, our method allows us to assess recognition performance not only in terms of phone confusion pairs but also in terms of the immediate left and right context of the phones. We showed how this information can be used in an error correction task to improve phone recognition accuracy, a task for which the traditional context-independent confusion matrix provides no improvement in accuracy. We observed that the type of correction employed is especially effective for the purposes of insertion rate reduction.

7. References

[1] Allen, J.B. "How Do Humans Process and Recognize Speech?", IEEE Trans. on Speech and Audio Processing, 2(4):567–577, 1994.

[2] Alshawi, H. "Effective Utterance Classification with Unsupervised Phonotactic Models"; to appear in HLT-NAACL, Edmonton, Canada, May 2003.

[3] Brill, E. "A Corpus-Based Approach to Language Learning", Ph.D. Dissertation, Department of Computer Science, University of Pennsylvania, 1993.

[4] Cucchiarini, C. "Phonetic Transcription: a Methodological and Empirical Study" , Ph.D. thesis, University of Nijmegen, 1993

[5] Golding, A. R., Roth, D. "A Winnow-Based Approach to Context-Sensitive Spelling Correction"; Machine Learning, Number 1-3, Vol. 34, pp 107-130, 1999.

[6] Levit, M., Nöth, E. and Gorin, A. L."Using EM-trained String Edit Distance for Approximate Matching of Acoustic Morphemes"; ICSLP-2002, Denver, 2002, Colorado, pp. 1157-1160.

[7] Mangu, L and Padmanabhan, M. "Error Corrective Mechanisms For Speech Recognition"; ICASP-2001, Salt Lake City, Uta, 2001.

[8] Morris, A.C., Misra, H. "Confusion Matrix Based Posterior Probabilities Corection", Submitted to ICASSP 2003

[9] Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M. and Makhoul, J. "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", in Proc. ICASSP-1985, Tampa, Florida, 1985, pp 1205-1208.

[10] Young, S.J., Foote, J.T., Jones, G.J.F., Sparck Jones, K. and Brown, M.G. "Acoustic Indexing for Multimedia Retrieval and Browsing", in Proc.ICASSP-97, April 1997, Vol. 1, pp. 199-202