Empirical Properties of Multilingual Phone-to-Word Transduction

Geoffrey Zweig [1] and Jon Nedel [2]
gzweig@microsoft.com, jnedel@gmail.com

September 2007

Technical Report
MSR-TR-2007-125

This paper explores the error-robustness of phone-to-word transduction across a variety of languages. It is motivated by recent literature advocating the decomposition of the decoding process into an initial phoneme recognition step followed by a subsequent word recovery step. We adopt this strategy, and use a phone-to-word transducer for word recovery. Our decoding process requires only a one-best phone string from the first stage and uses an error model on phones to recover from mistakes in the input. We implement a noisy channel model, and by controlling the error level, we are able to measure the sensitivity of different languages to degradation in the phonetic input stream. This analysis is carried further to measure the importance of each phone in each language individually. We study Arabic, Chinese, English, German and Spanish, and find that they behave similarly in this paradigm: in each case, a phone error produces about 1.4 word errors, and frequently incorrect phones matter slightly less than others. In the absence of phone errors, transduced word errors are still present, and we present an information-theoretic measure to explain the observed behavior.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1] Microsoft Research
[2] U.S. Department of Defense

1 Introduction

State-of-the-art speech recognition systems currently apply all the information sources at their disposal simultaneously in the decoding process. These sources consist of the pronunciation dictionary, the context model or decision tree, the language model, and the actual acoustic model or gaussians. This consolidation is most complete in decoders based of the Finite State Transducer paradigm [1, 2] where the dictionary, language model, and decision tree can be fully combined in advance of any decoding to produce a complete representation of the search space which can be stored on disk. This early combination is characteristic of dynamic decoders as well, where knowledge sources are applied as early as possible, for example with language model lookahead [3]. While this strategy is effective and unlikely to be replaced in the near future, from the research point-of-view it may be easier to implement and test new modeling techniques in a more decoupled framework.

Therefore, there has been a significant amount of work in recent years to support modularized recognizers for research purposes. In the FLaVoR architecture developed at Leuven University [4, 5], decoding is broken into a two step process, the first generating phone lattices and the second applying morpho-phonological and morpho-syntactic constraints to produce words. Similarly, in the *Automatic Speech Attribute Transcription* paradigm [6], it is proposed that the recognition process should proceed bottom up through multiple stages beginning with the detection of auditory cues such as articulatory features and proceeding through the application of independent modules to phone and word recognition. This sort of modularized architecture has a long and distinguished history [7, 8].

In an effort to better understand the properties of a modularized system, this paper studies the intrinsic difficulty of converting from phones to words. The first stage uses the phone set of [9] and associated acoustic models to recover a one-best phone sequence. The second stage uses a finite state transducer scheme to recover words from phones. In contrast with previous work on multi-stage decoding, our work relies solely on an error model in the transduction phase to formally model the mistakes that are made at the phone recognition level. The error model is an unconstrained model of IID insertions, substitutions and deletions, and more general than the *single error* model of [5]. The advantage of using this error model approach is that it allows us to directly implement a noisy channel model of speech communication, and to pose and answer a number of interesting questions that would be difficult to address otherwise. Specifically, we conduct a class of experiments that involves corrupting a reference phone sequence with a known error model, and then measuring our ability to recover words. This allows us to answer several questions that have not been well studied before:

1. How easy is it to recover words from a correct but unsegmented phone string, and how does this vary across languages?

2. As the phonetic input stream is corrupted with errors, how quickly is our ability to recover words degraded? Are there threshold effects where a

small number of phonetic errors can always be detected and recovered from? How does this vary across languages?

3. Are errors in some phones more important that errors in others, and how does this vary across languages?

4. How do the computational requirements of the phone-to-word transduction process vary as the phonetic input is progressively degraded?

This paper makes several contributions. First, we demonstrate that the proposed two-stage decoding process works across five CallHome [10] languages. Second, we show how our implementation of a noisy channel model can be used to answer the questions listed previously. We find that there is remarkable similarity across languages to degradation of the input phone string. In each case, 1 additional phone error causes approximately 1.4 word errors, and there are no threshold effects. While the slope of the phone-error vs. word-error curve is constant, the y-intercepts are different for different languages, and we present an information-theoretic measure to explain this phenomenon. At the individual phone level, we find that the improvement in word error rate caused by removing all the errors of a given phone is well-modeled by a power law. The change in word error rate is approximately equal to the change in phone error rate raised the the power 0.9. This also appears to be constant across the studied languages.

The remainder of this paper is organized as follows: in Section 2 we present the formulation and implementation of our method, along with the experimental methodology. Section 3 describes the CallHome dataset, and the phone recognizer that was used for the different languages. Section 4 examines the robustness of the transduction process to phonetic errors, and presents an explanation for the observed behavior. Section 5 addresses the question of whether some phones are more important than others. Section 6 offers concluding remarks.

2 Formulation

In the noisy-channel model we adopt, we assume that the sender begins with a sequence of words he or she intends to communicate, and attempts to speak a clean or reference phone sequence determined by the pronunciations of those words. A phone recognizer then processes the audio and produces an errorful version of the intended phones. The receiver gets this corrupted phone sequence and must decode the likeliest sequences of intended words. This can be more precisely stated if we let $\mathbf{w}_i$ denote the intended words, $\mathbf{p}_i$ denote the intended phone sequence, and $\mathbf{p}_c$ denote the corrupted phone sequence. The job of the

Intended words	I'm sorry we'll blame him
Intended phones	aI m S a r i: w i: l b l ei m H I m
Corrupted phones	aI m S a r i: w i: D l ei m H I m
Recovered words	I'm sorry we blame him

Table 1: Steps in the noisy channel model

decoder is then to determine

$$\begin{aligned}
\arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{p}_c) &= \arg\max_{\mathbf{w}} P(\mathbf{w})P(\mathbf{p}_c|\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} P(\mathbf{w}) \sum_{\mathbf{p}_i} P(\mathbf{p}_i, \mathbf{p}_c|\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} P(\mathbf{w}) \sum_{\mathbf{p}_i} P(\mathbf{p}_i|\mathbf{w})P(\mathbf{p_c}|\mathbf{p}_i, \mathbf{w}) \\
&\approx \arg\max_{\mathbf{w}} P(\mathbf{w}) \sum_{\mathbf{p}_i} P(\mathbf{p}_i|\mathbf{w})P(\mathbf{p_c}|\mathbf{p}_i) \\
&\approx \arg\max_{\mathbf{w},\mathbf{p_i}} P(\mathbf{w})P(\mathbf{p}_i|\mathbf{w})P(\mathbf{p_c}|\mathbf{p}_i)
\end{aligned}$$

The factors involved in the maximization each have simple interpretations: $P(\mathbf{w})$ is given by the language model; $P(\mathbf{p}_i|\mathbf{w})$ is given by the pronunciation model; and $P(\mathbf{p}_c|\mathbf{p}_i)$ is given by the phone-level error model. The maximization over words and phones can be done with Viterbi recursions. In all the experiments described subsequently, we use a first-order error model with insertion and deletion probabilities for every phone, and substitution probabilities for all pairs of phones. Table 1 illustrates an example of our noisy channel model.

There is a simple representation of this model in terms of finite state transducer operations. If we denote the intended word sequence by W, the pronunciation dictionary by P, the language model by L and the error model by E, then the received (corrupted) phone sequence R is given by $R = \mathit{sample}(W \circ P \circ E)$. The operation of decoding can be represented as $\mathit{bestpath}(R \circ E^{-1} \circ P^{-1} \circ L)$. As an implementation issue, there can be efficiency issues with a straightforward application of these operations. For example, the application of the inverse error model E^{-1} to $P^{-1} \circ L$ will increase its size by $O(N)$ where N is the number of phones. We have avoided such efficiency issues by storing only $P^{-1} \circ L$ as a transducer and building the application of the inverse error model directly into the dynamic programming recursions that find the likeliest word sequence.

Given this formulation, it is possible to explore a number of interesting questions. First, if one sets aside the issue of errors, how easy is it to recover words from phones? And how does this vary across languages? This can be answered simply by following the process with an "identity" error model that never inserts or deletes, and always replaces a phone by itself. Second, how sensitive is the decoding process to phone errors, and how does this vary across languages? This can be answered by constructing error models with higher or lower error rates, and then computing $\mathit{bestpath}(\mathit{sample}(W \circ P \circ E) \circ (E^{-1} \circ$

	# Training	# Devtest	Lexicon	OOVs
Egyptian	149k	33k	57k	1.6%
German	165	43	315	1.1
English	167	43	99	1.8
Mandarin	159	42	44	3.1
Spanish	145	38	45	2.6

Table 2: Language data statistics. All units except OOVs are words.

$P^{-1} \circ L$)). Finally, it is possible to explore the importance of single phones. Let E_p be the original error model E, except that errors involving phone p are adjusted to have zero probability. Then measuring the difference between using E and E_p in the round-trip process gives an indication of the importance of p.

We have explored the use of this methodology in five of the CallHome languages and using an acoustic model that uses a universal phone set. The database and acoustic model are described next.

3 Database and Acoustic Models

3.1 CallHome

In order to work with a data set with roughly equal resources across a variety of languages, we selected the CallHome database [10]. This database has speech, transcriptions, and lexica in Egyptian Arabic, Mandarin Chinese, English, German, Japanese, and Spanish. The audio data for each language consists of 120 telephone conversations of up to 30 minutes each (100 conversations for German); the conversations are between people in North America and their family members and friends overseas. Eighty of the conversations are marked as training data and 20 each for development and test, except for German which has development data only. Since the experiments did not involve parameter tuning and a test set is absent for German, all results are reported on the development set. Due to an 18% out-of-vocabulary rate for the Japanese lexicon, we did not use the Japanese language data. Statistics on the remaining five languages are presented in Table 2.

3.2 The UPR

To conduct our experiments, we need a phone-level error model for each language, reflecting realistic error patterns. To obtain these error models, we decoded the training data with acoustic models based on a universal phone recognizer (UPR) provided by the Department of Defense [9]. This recognizer uses 259 phones based on the International Phonetic Alphabet (IPA), and represents an effort similar to that pioneered with the GlobalPhone project and others [11, 12].

Language	Acoustic Training Data: Hours of Speech
Egyptian	17.9
German	15.2
English	88.0
Mandarin	76.2
Spanish	43.3

Table 3: Amount of UPR training data used for each language

The UPR system was built using the HTK Recognizer, version 3.3 and was trained iteratively, starting with data that was transcribed at the phone level, and later incorporating data that was transcribed at the word level. The universal acoustic models are trained using an iterative, bottom-up, two-level forced alignment and training procedure, described in detail in [9]. This procedure ensures that sounds represented by a given system are consistent across languages and that important phonetic distinctions in one language are annotated in all languages. For example, in English, aspiration of stop consonants is not contrastive and therefore is not usually labeled or explicitly modeled. However, in Hindi, aspiration of stop consonants is a feature that distinguishes different phonemes. A UPR system must be able to explicitly model aspiration of stop consonants.

When training the UPR, phone-level transcriptions were used to train the earliest models. This data was taken from the Phonetic Switchboard Corpus (SwbPhn) [13, 14] in English, and the OGI-MLTS Corpus [15] in English, German, Hindi, Japanese, Mandarin Chinese, and Spanish. Word-level transcriptions were later used to incorporate data from LDC data sets (e.g. CallHome and CallFriend) in a variety of languages. Table 3 presents the amount of training data used to train the system for each of the languages investigated in this paper.

The UPR acoustic models have diphone acoustic context, with 17 gaussians per state. The acoustic features were 39-dimensional, consisting of cepstra, deltas and double-deltas, and decoding was performed at the speaker-independent level. The UPR can also make use of n-gram phonotactic language models trained on transcripts of LDC data as well as data found on the Web. Language-specific phonotactic bigram language models were built for all the languages used in our experiments. Further details of the UPR phone set, acoustic model, and phonotactic language models can be found in [9].

The UPR can be run using either a truly universal model or using language-specific models. We used language-specific models to decode the CallHome training data and create the error models because language-specific models were available for all the CallHome languages, and they were more accurate than the universal models. The phone-error rates on the development data varied from 56.4% in English to 63.0% in German. Error rates on training and development data were essentially identical. To provide better statistics, our error models are based or errors observed in decoding the training data.

4 Robustness to Phonetic Errors

This section reports on the sensitivity of the transduction process to the overall error level in the input phone stream. The experiments all use a base error model that is obtained by decoding the CallHome training data with the UPR, aligning it to the reference phoneme string, and computing the various substitution, insertion and deletion probabilities. This is done separately for each language. To obtain error models at a variety of absolute error levels, we then scale this matrix down by moving probability mass from insertions, deletions and non-identity substitutions to identity substitutions. By corrupting the reference phones with the various error matrices and then measuring our ability to recover the correct words, we determine the sensitivity of the decoding process to input errors.

4.1 Accuracy and Speed

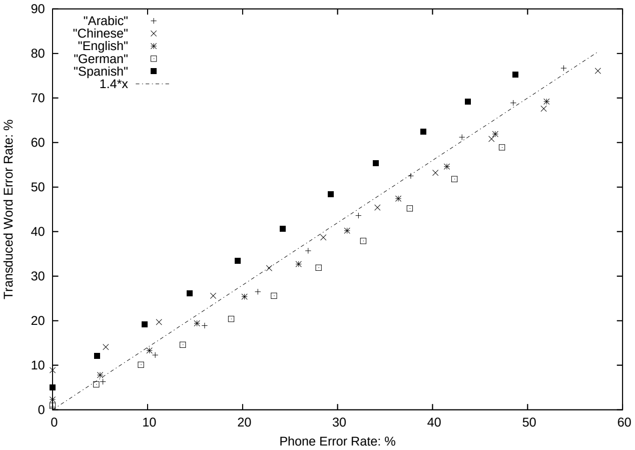

Figure 1: Output word error rate vs. input phone error rate

Figure 1 plots transduced word error rate (WER) as a function of the input phone error rate (PER). To a first approximation, the two are related by $WER = 1.4PER + \epsilon_{lang}$. The slope in all cases is approximately 1.4, and there is a language dependant y-intercept. These results show no evidence of redundancy – if redundancy were present, one would expect a threshold effect in which very low phone-error rates would have little or no impact on word error rate. Figure 2 plots system throughput as a function of the input phone error rate. The y-axis is logscale; whereas the word error rate scales linearly with phone error rate, the runtime increases exponentially. These experiments were run with a fixed beam such that there was little accuracy loss at high phone error rates.

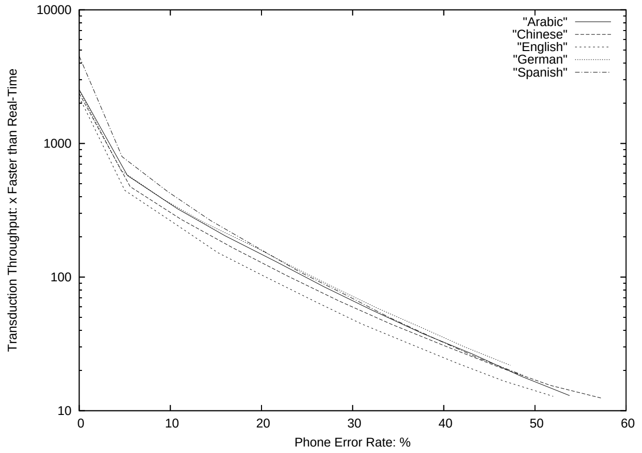

Figure 2: Runtime vs. phone error rate

4.2 The Pronunciation Gap: Explaining the y-intercept

The transduced word error rate achieved in the absence of any phone errors is not zero, and differs by over a factor of ten for the different languages: from 0.6% for Arabic to 8.8% for Mandarin (our Mandarin results are reported in terms of character error rate - CER - as is conventional). The fact that the error rates in this case are not zero is attributable to two main factors: there may be multiple ambiguous ways of segmenting a phone sequence into words, and even for a given segmentation, homonyms can make the word-labeling ambiguous. The low error rate for Arabic is explainable since short vowels are present in the Arabic transcripts and the phonetic sequence is therefore essentially identical to the graphemic sequence of the words. The high error rate in Mandarin is also explicable by the fact that the UPR phones used are tone-insensitive and thus cause a large number of homonyms in the lexicon.

To present a more formal explanation of the observed differences in the y-intercept, we examine the mutual information between word sequences and their derived phone sequences. This is a function of both the lexicon and the language model. It is clearly a function of the lexicon since if all words had the same pronunciation, there would be no information. It is a function of the language model because the mutual information will decrease if the language model has a propensity to generate ngrams of words whose phone sequences admit alternate word segmentations, or if it tends to generate homonymns.

To define the mutual information between phones as words, let $\mathbf{r_s}$ be the phone sequence for utterance s in the database. Let $\mathbf{l_s}$ be the word sequence.

	Phone-to-word WER	Phonetic Gap: bits
Egyptian	0.6%	-0.0020
German	0.9	-0.029
English	2.3	-0.080
Spanish	5.0	-0.18
Mandarin	8.9 (CER)	-0.44

Table 4: The phonetic gap

Let R and L be phone-sequence and word-sequence variables respectively that take specific values such as $\mathbf{r_s}$ and $\mathbf{l_s}$. Then

$$\begin{aligned} M(L;R) &= \sum_{L,R} P(L,R)\log\frac{P(L,R)}{P(L)P(R)} \\ &\approx \sum_{s\in observed\_data\_segments} \log\frac{P(\mathbf{r}_s,\mathbf{l}_s)}{P(\mathbf{r}_s)P(\mathbf{l}_s)} \\ &= \sum_s \log\frac{P(\mathbf{r}_s|\mathbf{l}_s)P(\mathbf{l}_s)}{P(\mathbf{r}_s)P(\mathbf{l}_s)} \\ &= \sum_s \log\frac{P(\mathbf{r}_s|\mathbf{l}_s)}{\sum_{\mathbf{w}} P(\mathbf{r}_s|\mathbf{w})P(\mathbf{w})} \end{aligned}$$

$P(\mathbf{w})$ is given by the language model. $P(\mathbf{r_s}|\mathbf{w})$ is the probability of an observed phone string given a word string. It is given by the sum over all the alignments of $\mathbf{r_s}$ to the phones in $\mathbf{w}$ of the probability of the substitutions, insertions and deletions in the alignment, and can be computed using dynamic programming.

The quantity $M(L;R)$ is a measure of how much information the phones provide about the words. If we let H be the entropy of the language, then $M(L;R) - H$ provides a measure of the excess information that is available when inferring words from phones. This can be thought of as a "phonetic gap" between the information present in the phones and that necessary to disambiguate the words.

In general, $M(L;R)$ is difficult to compute since it involves summing over all possible word sequences in the denominator. To simplify the computation, we have approximated the sum over all data segments by a sum over the words in the lexicon weighted by their unigram frequency. Essentially this uses a notional data set consisting of the words in the lexicon. In this notional data set, each data segment is just a single word, so the denominator computation for each segment involves only $O(V)$ computations where V is the vocabulary size. Under this assumption, $M(L;R)$ is bounded from above by the entropy of the unigram language model so the pronunciation gap will always be negative.

Table 4 shows the phonetic gap along with the round trip word word error rates. It can be seen that there is a good correlation between the gap and the observed word error rate.

5 Sensitivity to Individual Phones

By using our noisy channel model, we have been able to study the sensitivity of word error rate to individual phones in two ways. The first uses a corruption process similar to that described previously. Insertions, deletions and substitutions are made in an IID fashion according to the empirically derived error model, with one exception: all errors involving a particular phone are excluded. The corruption process is run separately for each phone, and the resulting strings are then transduced to words. The transduced word error rate is then computed, and we compute the decrease in error rate over the baseline where no errors are excluded. Simply looking at the decrease in word error rate gives an indication of how important a phone is. However, one might expect that more frequent phones or those involved in numerous errors would be more important by this measure. To normalize against such frequency effects, we also count the number of phone errors that have been excluded from the input. This allows us to create a scatterplot plotting the number of word errors corrected after transduction against the number of phone errors corrected on the input side. This is shown in Figure 3 for each phone in each of the languages studied.

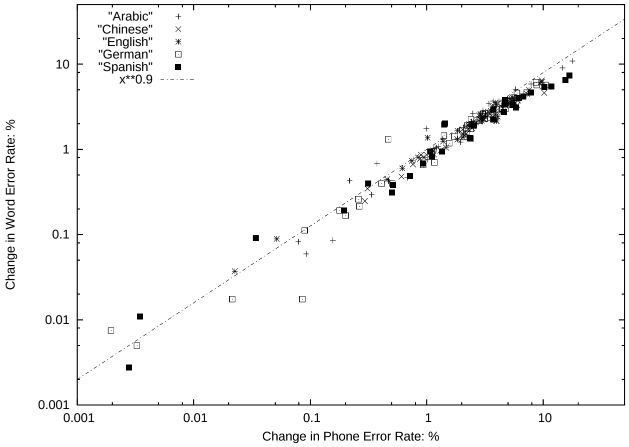

Figure 3: Sensitivity to individual phones with simulated errors

The second method of computing sensitivity to individual phones avoids the artificial corruption process. This is done by aligning the phone-level UPR output to the reference phone string. Then, for a particular phone, we fix all the errors involving the phone (insertions, deletions, or non-identity substitutions either to or from the phone). The remaining steps are identical to the first method, and we obtain another scatterplot. This is shown in Figure 4. The fact that the two scatterplots are similar increases our confidence in the reliability of the artificial IID corruption process.

Figures 3 and 4 are consistent with the observation seen in Section 4 that word error rate is linearly dependent on phone error rate. To a first approxi-

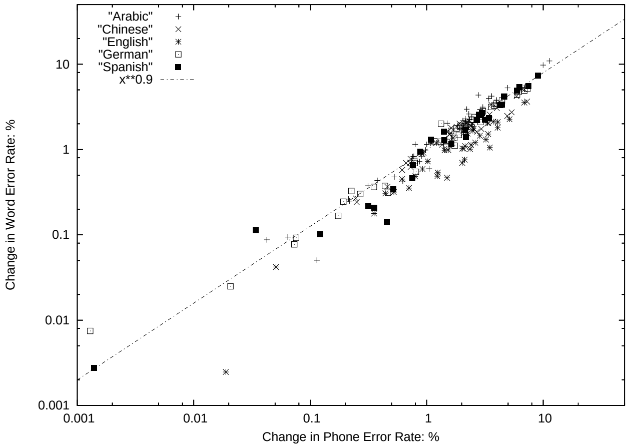

Figure 4: Sensitivity to individual phones using the raw data

mation, this holds true at the level of individual phones as well. A closer look, however, shows that this is not quite the case: it is on a log-log scale that the relationship appears linear, with a slope of about 0.9. This implies that there is a slight tendency such that phones which are frequently involved in errors are relatively downweighted.

6 Discussion

This paper has examined the robustness of phone-to-word transduction in a variety of languages and over a range of error rates. The methodology directly implements a noisy channel model in which an intended word string is converted to a corrupted phone string which is then decoded to recover words. We find that even at low error rates, the introduction of a phone error on average creates about 1.4 word errors. This is seen to be constant across the five languages studied, and across a wide range of absolute error levels. At the level of individual phones, the sensitivity to errors is almost linear as well, but seems to be optimized in the sense that frequently misleading phones have slightly less impact per error than their more reliable counterparts.

These observations are subject to several caveats. First, they are specific to CallHome - the particular amounts of language model training text, the specific lexica, and the conversational style of the speech. Further experimentation must be done to see how well the observations hold up in other domains. Second, the observations are specific to the transduction process used and might not reflect what would be seen from human listening experiments in which the transmitted phone sequence was corrupted, or what would be seen with some other decoding methodology (e.g. using semantic or syntactic information). Finally, the errors were introduced in an IID manner, and under the assumption that the error pattern is the same at low absolute rates as at high rates. Despite these caveats, the findings are reasonable: the lack of redundancy indicates a system optimized for fast transmission, and no single phone seems to be unduly responsible for errors.

Acknowledgements

We thank Pat Schone for assistance with the UPR and Hynek Hermansky, Alex Acero, Li Deng and Patrick Nguyen for comments and references.

References

[1] M. Mohri, F. Pereira, and M. Riley, "Weighted finite state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.

[2] G. Saon, G. Zweig, and D. Povey, "Anatomy of an extremely fast LVCSR decoder," in *Interspeech*, 2005.

[3] S. Ortmanns, H. Ney, and A. Eiden, "Language-model look-ahead for large vocabulary speech recognition," in *ICSLP*, 1996.

[4] K. Demuynck, T. Laureys, D. Van Compernolle, and H. Van Hamme, "FLaVoR: a flexible architecture for LVCSR," in *Eurospeech*, 2003.

[5] K. Demuynck, D. Van Compernolle, and H. Van Hamme, "Robust phone lattice decoding," in *Interspeech*, 2006.

[6] C.H. Lee, M. Clements, S. Dusan, Eric Fosler-Lussier, K. Johnson, B.H. Juang, and L. Rabiner, "An overview on automatic speech attribute transcription (ASAT)," in *Interspeech*, 2007.

[7] J. Wolf and W. Woods, "The HWIM speech understanding system," in *ICASSP*, 1977.

[8] L. Erman, F. Hayes-Roth, V. Lesser, and R. Reddy, "The hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty," *Computing Surveys*, vol. 12, no. 2, 1980.

[9] B. Walker, B. Lackey, J. Muller, and P. Schone, "Language-reconfigurable universal phone recognition," in *Eurospeech*, 2003.

[10] "Linguistic data consortium," http://www.ldc.upenn.edu/.

[11] T. Schultz and A. Waibel, "Language independent and language adaptive acoustic modeling for speech recognition," *Speech Communication*, 2001.

[12] C. Corredor-Ardoy, L. Lamel, M. Adda-Decker, and J.L. Gauvain, "Multilingual phone recognition of spontaneous telephone speech," in *ICASSP*, 1998.

[13] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *ICASSP*, 1992, pp. 517–520.

[14] S. Greenberg, "The switchboard transcription project," Tech. Rep., Johns Hopkins Workshop on Innovative Techniques for LVCSR, 1996.

[15] A. Cole, Y. Muthusamy, and B. Oshikal, "The OGI multi-language telephone speech corpus," in *ICSLP*, 1992.