Phonological Parsing for Bi-directional Letter-to-Sound/Sound-to-Letter Generation[1]

Helen M. Meng, Stephanie Seneff and Victor W. Zue

Spoken Language Systems Group, Laboratory for Computer Science
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

ABSTRACT

In this paper, we describe a reversible letter-to-sound/sound-to-letter generation system based on an approach which combines a rule-based formalism with data-driven techniques. We adopt a probabilistic parsing strategy to provide a hierarchical lexical analysis of a word, including information such as morphology, stress, syllabification, phonemics and graphemics. Long-distance constraints are propagated by enforcing local constraints throughout the hierarchy. Our training and testing corpora are derived from the high-frequency portion of the Brown Corpus (10,000 words), augmented with markers indicating stress and word morphology. We evaluated our performance based on an unseen test set. The percentage of nonparsable words for letter-to-sound and sound-to-letter generation were 6% and 5% respectively. Of the remaining words our system achieved a word accuracy of 71.8% and a phoneme accuracy of 92.5% for letter-to-sound generation, and a word accuracy of 55.8% and letter accuracy of 89.4% for sound-to-letter generation. We also compared our hierarchical approach with an alternative, single-layer approach to demonstrate how the hierarchy provides a parsimonious description for English orthographic-phonological regularities, while simultaneously attaining competitive generation accuracy.

INTRODUCTION

This paper describes a trainable probabilistic system for reversible letter-to-sound/sound-to-letter generation. Sound-to-letter generation is a crucial aspect in the problem of automatic detection/incorporation of new words, which is in turn critical for the development of large vocabulary speech understanding systems. Moreover, letter-to-sound generation will continue to be important for speech output, especially in applications such as reading machines. To successfully achieve our goal, several important issues must be addressed. First, what should be the inventory of linguistic or lexical units for describing English orthographic-phonological regularities? Second, how should these units be incorporated into the representation of English orthography and phonology? Third, what algorithms can be used to synthesize and analyze the spelling and pronunciation of an English word in terms of these lexical units? These three issues will be addressed in detail in the following when we describe our approach and report on our system's performance for both letter-to-sound [1] and sound-to-letter generation [2]. The novel features of our approach include the reversibility of the combined parsing and generative processes, the ability to provide multiple output hypotheses, the capability of handling uncertainty in the input, as well as our treatment of non-parsable words.

PREVIOUS WORK

Letter-to-Sound Generation

One of the first approaches adopted for letter-to-sound generation is typified by MITalk [8]. It follows the theories of generative grammar and the transformational cycle as proposed by Chomsky and Halle [3]. A large set of ordered cyclical rules are applied in turn to the word in question until a final pronunciation emerges. While the process of establishing the appropriate rule set was tedious and time-consuming, the resulting system achieved a degree of accuracy that, to our knowledge, has not yet been matched by other more automatic techniques.

Because the generation of cyclical rules is a difficult and complicated task, several research groups have attempted to acquire letter-to-sound generation systems through automatic or semi-automatic data-driven techniques, based on neural nets or on an information theoretic approach. Typically, the goal is to provide as little *a priori* information as possible, ideally, only a set of pairings of letter sequences with corresponding (aligned or unaligned) phone sequences. Iterative training algorithms then produce a probability model that is applied to predict the most likely pronunciation. Probably the best known of these systems is NETtalk [4], which learns a pronunciation of the current letter by considering the six surrounding letters as input to the neural network. Lucassen and Mercer [5] acquired a set of rules automatically from a large lexicon of phonetically labelled data by growing decision trees using a criterion based on mutual information. Although direct comparisons of performance of different systems is difficult due to the lack of standardized phone sets, data sets, or scoring algo-

[1] This research was supported by ARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research, and a grant from Apple Computer Inc.

rithms, these systems have reported phone accuracies in the low 90's in terms of the percent of letters correctly pronounced.

Sound-to-Letter Generation

To our knowledge, there has been very little previous work reported in the literature addressing the problem of sound-to-letter generation. We are aware of only two prior research efforts in this area.

Lucas and Damper [6] developed a system for bidirectional text-phonetics translation using two neural networks to perform statistical string translation. This system does not require pre-aligned text-phonetic pairs for training, but instead tries to infer appropriate segmentations and alignments. In a phonetics-to-text translation task using two disjoint 2,000-word corpora for training and testing, they reported a 71.3% letter and a 22.8% word accuracy.

Another related effort was conducted by Alleva and Lee [7], who used HMMs to model the *acoustics* of training sentences based on the orthographic transcriptions. Context-dependent quad-letter acoustic models were trained with 15,000 sentences, and used in conjunction with a 5-gram letter language model. Testing on a disjoint corpus of 30 embedded and end-point detected words (place and ship names) gave a 39.3% letter error rate and 21.1% word accuracy. However, this result is not directly comparable to our work because the phonemic/phonetic representation is bypassed.

A HIERARCHICAL LEXICAL REPRESENTATION

It has long been realized from research in speech synthesis that a variety of linguistic knowledge sources play an important role in determining English letter/sound correspondences [8]. For example, part-of-speech causes the noun and verb forms of "record" to be pronounced differently. A morphological boundary causes the letter sequence "sch" in "discharge" to be realized differently from that in "school" or "scheme". Stress changes the identity of vowels in a word, e.g. "define" vs. "definition". Also, syllabic constraints are expressible in terms of the sequential ordering of distinctive features - sonority sequencing in manner features, and phonotactic constraints in place and voicing features. Furthermore, there are graphemic constraints for letter to letter transitions. A novel feature of our system is that multiple layers of representation are incorporated to capture short and long distance constraints. These include *word class, morphs, syllables, manner classes, phonemes* and *graphemes.*

We created a framework which describes the spelling and pronunciation of English words using only a small inventory of labels associated with the aforementioned morphological and phonological units. These units are organized as a hierarchical tree structure, where the various levels of linguistic knowledge are collectively used to describe orthographic-phonological regularities. Figure 1 illustrates the description of the word "dedicated".

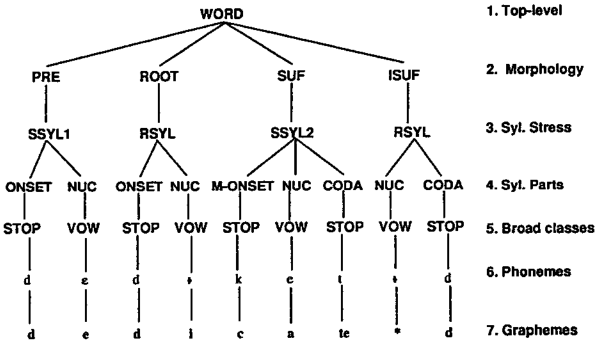

Figure 1: Parse tree for "dedicated" with different linguistic layers indicated numerically.

The higher levels encode longer distance constraints, while the lower levels carry more local constraints. By allowing the terminal nodes to be dual in nature (i.e., representing either phones or letters), we can create direct symmetry between the letter-to-sound and sound-to-letter generation tasks simply by swapping the input/output specification.

One should note in Figure 1 that [*] is a graphemic "place-holder" introduced to maintain consistency between the representations of the words "dedicate" and "dedicated", where an inflexional suffix [ISUF] has been attached to the latter word. Another noteworthy detail is the special [M-ONSET] category, which signifies that the letter 'c' should belong to the root "-dic-",[2] but has become a moved onset of the next syllable due to syllabification principles such as the Maximal Onset Principle and the Stress Resyllabification Principle.[3]

THE PARSING ALGORITHM

We are adopting a technique that represents a cross between explicit rule-driven strategies and strictly data-driven approaches. About 100 generalized context-free rules, such as those illustrated in Table 1 are written by hand, and training words are parsed using TINA [9], according to their marked linguistic specifications. The parse trees of format as show in Figure 1 are then used

[2] According to Webster's New World Dictionary, the root of "dedicated" is "-dic-", which is derived from the Latin word "dicare".

[3] The Maximal Onset Principle states that the number of consonants in the onset position should be maximized when phonotactic and morphological constraints permit, and Stress Resyllabification refers to maximizing the number of consonants in stressed syllables.

word	→ [prefix] root [suffix]
root	→ stressed-syllable [reduced-syllable]
stressed-syllable	→ [onset] nucleus [coda]
nucleus	→ vowel
nasal	→ (/m/ /n/ /ŋ/)
/m/	→ ("m" "me" "mn" "mb" "mm" "mp")

Table 1: Example rules at each of the different layers.

to train the probabilities in a set of "layered bigrams" [10]. We have chosen a probabilistic parsing paradigm for four reasons: First, the probabilities serve to augment the known structural regularities that can be encoded in simple rules with other structural regularities which may be automatically discovered from a large body of training data. Secondly, since the more probable parse theories are distinguished from the less probable ones, search efforts can selectively concentrate on the high probability theories, which is an effective mechanism for perplexity reduction. Thirdly, probabilities are less rigid than rules, and adopting a probabilistic framework allows us to easily generate multiple parse theories. Fourthly, the flexibility of a probabilistic framework also enables us to automatically relax constraints to attain better coverage of the data.

Training Procedure

The layered bigrams formalism attaches probabilities to sibling-sibling transitions in context-free grammar rules. It has been shown to achieve a low perplexity at the linguistic level within the ATIS domain [10]. For our current sub-word application, we have modified the layered-bigrams in two ways: (1) parse trees are generated in a bottom-up fashion instead of top-down, and (2) the contextual information used in bottom-up prediction includes the complete history in the immediate left column.

Our experimental corpus consists of the 10,000 most frequent words appearing in the Brown Corpus [11], where each word entry contains a spelling and a *single* unaligned phoneme string. We used about 8,000 words for training, and a disjoint set of about 800 words for testing.

The set of training probabilities are estimated by tabulating counts using the training parse trees.[4] It includes bottom-up prediction probabilities for each category in the parse tree, and column advancement probabilities for extending a column to the next terminal. The same set of probabilities are used for both letter-to-sound and sound-to-letter generation.

Testing Procedure

In letter-to-sound generation, the system takes in a spelling as an input, generates a parse tree in a bottom-up left-to-right fashion, and derives a phonemic pronunciation from the complete parse. In sound-to-letter generation, the system accepts a string of phonemes as input, and generates letters. An inadmissible stack decoding search algorithm is adopted for its simplicity. If multiple hypotheses are desired, the algorithm can terminate after multiple complete hypotheses have been popped off the stack. These hypotheses are subsequently re-ranked according to their actual parse score. Though our search is inadmissible, we are able to obtain multiple hypotheses inexpensively with satisfactory performance.

EXPERIMENTAL RESULTS

Experiments on both letter-to-sound and sound-to-letter generation were conducted using 26 letters, one graphemic place-holder and 52 phonemes (including several unstressed vowels and pseudo diphthongs such as /o r/). Each entry in the test corpus contains a spelling corresponding to a *single* pronunciation. The generation procedures use evaluation criteria that directly mirror one another. Word accuracy is the percentage of *parsable* words for which the top-ranking theory generates a spelling/pronunciation that matches the lexical entry exactly. Non-parsable words are those for which no spelling/pronunciation output is produced. "Top N" word accuracy refers to the percentage of parsable words for which the correctly generated spelling/pronunciation appears in the top N complete theories. Letter/Phoneme accuracies include insertion, substitution and deletion error rates, and are obtained using the program provided by NIST for evaluating speech recognition systems.

Results on Letter-to-Sound Generation

In letter-to-sound generation, about 6% of the test set was nonparsable. This set consists of compound words, proper names, and words that failed due to sparse data problems. Results for the parsable portion of the test set are shown in Table 2. The 69.3% word accuracy corresponds to a phoneme accuracy of 91.7%, where an insertion rate of 1.2% has been taken into account.

Thus far there are no standardized evaluation methods for text-to-speech systems, and therefore comparison among different systems remains difficult. Errors in the generated stress pattern and/or phoneme *insertion* errors are often neglected. Evaluation criteria that have been used include word accuracy, accuracy per phoneme and accuracy per letter (in measuring the accuracy per letter, silent letters are regarded as mapping to a [NULL] phone). We believe that accuracy per letter would generally be higher than accuracy per phoneme, because there are generally more letters than phonemes per word, and the letters mapping to the generic category [NULL] would usually be correct. To verify our claim, we computed the two measurements based on our training set, using the alignment provided by the training parse trees. Our re-

[4] See [1] for a more detailed description of this process.

Accuracy		top choice correct	top 5 correct	top 10 correct
train	word	77.3%	93.7%	95.7%
	phoneme	94.2%	–	–
test	word	69.3%	86.2%	87.9%
	phoneme	91.7%	–	–

Table 2: Letter-to-Sound-Generation Experiments: Word and Phoneme Accuracy for Training and Testing data

Accuracy		top choice correct	top 5 correct	top 10 correct
train	word	58.8%	85.0%	89.3%
	letter	90.6%	–	–
test	word	51.9%	77.0%	81.1%
	letter	88.6%	–	–

Table 3: Sound-to-Letter Generation Experiments: Word and Letter Accuracy for Training and Testing data

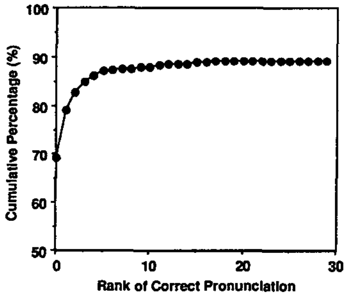

Figure 2: Letter-to-Sound: Percent correct whole-word theories as a function of N-best depth for the test set

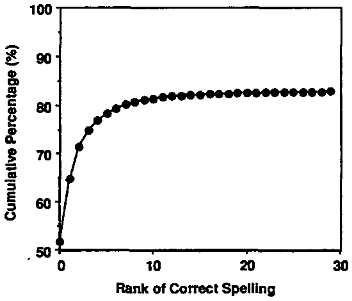

Figure 3: Sound-to-Letter: Percent correct whole-word theories as a function of N-best depth for the test set

sult shows that a *per letter* measurement would lead to a 10% reduction in error rate.

Figure 2 is a plot of cumulative percent correct of whole word theories as a function of the N-best depth for the test set. Although 30 complete theories were generated for each word, no correct theories occur beyond N =18 after resorting, with an asymptotic value of just over 89%.

Results on Sound-to-Letter Generation

In sound-to-letter generation, about 4% of the test set was nonparsable. Results for the parsable words are shown in Table 3; top-choice word accuracy for sound-to-letter is about 52%. This corresponds to a letter accuracy of 88.6%, with an insertion error rate of 2.5% taken into account. This performance compares favorably with those reported in previous work.

Figure 3 is a plot of the cumulative percent correct (in sound-to-letter generation) of whole word theories as a function of N-best depth of the test set. The asymptote of the graph shows that the first 30 complete theories generated by the parser contain a correct theory for about 83% of the test words. Within this pool, resorting using the actual parse score has put the correct theory within the top 10 choices for about 81% of the cases, while the remaining 2% have their correct theories ranked between N = 10 and N = 30. Resorting seems to be less effective in the sound-to-letter case, presumably because many more "promising" theories can be generated than for letter-to-sound. A possible reason for this is the ambiguity in phoneme-to-letter mapping, and another reason is that geminant letters are often mapped to the same (consonantal) phoneme. For example, the generated spellings from the pronunciation of "connector" i.e., the phoneme string (k ɪ n ɛ k t ɚ), include: "conecter", "conector", "connecter", "connector", "conectar", "conectyr", "conectur", "connectyr", "connectur", "conectter", "connectter" and "cannecter". Many of these hypotheses can be rejected with the availability of a large lexicon of legitimate English spellings.

Error Analyses

Both of the cumulative plots shown above reach an asymptotic value well below 100%. The words that belong to the portion of the test set lying above the asymptote appear intractable – a correct pronunciation/spelling did not emerge as one of the 30 complete theories. Detailed analysis of these words shows that they fall into approximately 4 categories. (1) Generated pronunciations that have subtle deviations from the reference strings. (2) Unusual pronunciations due to influences from foreign languages. (3) Generated pronunciations which agree with the regularity of English letter-phoneme mappings, but were nevertheless incorrect. (4) Errors attributable to sparse data problems. Some examples are shown in Table 4. It is interesting to note that there is much overlap between the set of problematic words in letter-to-sound and sound-to-letter generation. This implies that

Category	correct spelling	generated spelling	generated pronunciation	correct pronunciation
(1) Subtle	acquiring	equiring	ɪkwɑ^{y}rɪŋ	ɪkwɑ^{y}ɝɪŋ
	balance	balence	*correct*	bælɪns
	launch	lawnch	*correct*	lɔnč
	pronounced	pronounst	prɪnɑ^{w}nst	pro^{w}nɑ^{w}nst
(2) Unusual	champagne	shampain	čæmpɪgni^{y}	šæmpe^{y}n
	debris	dibree	di^{y}brɪs	dɪbri^{y}
(3) Regular	basis	*correct*	bæsɪs	be^{y}sɪs
	elite	aleat	ɪlɑ^{y}t	ɪli^{y}t
	violence	viallence	*correct*	vɑ^{y}ɪlɪns
	viscosity	viscossity	vɪsko^{w}sɪti^{y}	vɪskɑsɪti^{y}
(4) Sparse	braque	brack	brækwi^{y}	bræk

Table 4: Some examples of generation errors

improvements made in one generative direction should carry over to the opposite direction as well.

EVALUATING THE HIERARCHY

We believe that the higher level linguistic knowlege incorporated in the hierarchy is important for our generation tasks. Consequently, we would like to empirically assess: (1) the relative contribution of the different linguistic layers towards generation accuracy, and (2) the relative merits of the overall design of the hierarchical lexical representation. Our studies [13] are based on letter-to-sound generation *only*, although we expect that the implications of our study should carry over to sound-to-letter generation.

Investigations on the Hierarchy

The implementation of our parser is flexible, in that it can train and test on a variable number of layers in the hierarchy. This enables us to explore the relative contribution of each linguistic level in the generation task. We conducted a series of experiments whereby an increasing amount of linguistic knowledge (in terms of the number of layers in the hierarchy) is omitted from the training parse trees. For each reduced configuration, the system is re-trained and re-tested on the same training and testing corpora as described earlier. For each experiment we compute the *top-choice word accuracy* and *perplexity*, which reflect the amount of constraint provided by the hierarchical representation. We also measure the *coverage* to show the extent to which the parser can generalize to account for previously unseen structures, and count the *number of system parameters* in order to observe the computational load, as well as the parsimony of the hierarchical framework in capturing English orthographic-phonological regularities. We found that for every layer omitted from the representation, linguistic constraints are lost, manifested as a lower generation accuracy, higher perplexity and greater coverage. Fewer layers also require fewer training parameters.

The significant exception was the case of omitting the layer of broad classes (layer 5), which seems to introduce *additional* constraints, thus giving the highest generation performance. The word accuracy based on the parsable portion of the test set was 71.8%,[5] which corresponds to a phoneme accuracy of 92.5%. This improvement[6] can be understood by realizing that broad classes can be predicted from phonemes with certainty, and the inclusion of the broad class layer probably led to excessive smoothing across the individual phonemes within each broad class.[7] Again, about 6% of the test set was nonparsable. When a robust parsing scheme is used to recover the nonparsable words, 100% coverage was achieved, but performance degrades to 69.2% word and 91.3% phoneme accuracy.

Comparison with a Single-Layer Approach

We also compared our current hierarchical framework with an alternative approach which uses a single-layer representation. Here, a word is represented mainly by its spelling and an aligned phonemic transcription, using the [NULL] phoneme for silent letters. The alignment is based on the training parse trees from the hierarchical approach. For example, "bright" is transcribed as /b r ɑ^{y} NULL NULL t/. The word is then fragmented exhaustively to obtain letter sequences (word fragments) shorter than a set maximum length. During training, bigram probabilities and phonemic transcription probabilities are computed for each letter sequence. Therefore this approach

[5] When normalized on the entire test set, the word accuracy becomes 67.5%.

[6] We also found improvement on sound-to-letter generation – 55.8% word accuracy on the parsable test words, corresponding to 89.4% letter accuracy, and 5% of the words were nonparsable.

[7] However, broad classes may still serve a role as a "fast match" layer in recognition experiments, where their predictions could no longer be certain, due to recognition errors.

captures some graphemic constraints within the word fragment, but higher level linguistic knowledge is not explicitly incorporated. Letter-to-sound generation is accomplished by finding the "best" concatenation of letter sequences which constitutes the spelling of the test word.

To facilitate comparison with the hierarchical approach, we use the same training and test sets to run letter-to-sound generation experiments with the single-layer approach. Several different value settings were used for the maximum word fragment length. We expect generation accuracy to improve as the maximum word fragment length increases, because longer letter sequences can capture more context. However, this should be accompanied by an increase in the number of system parameters due to the combinatorics of the letter sequences. Furthermore, there are no nonparsable test words in the single-layer approach, because it can always "backoff" to mapping a single letter to its most probable phoneme.

The hierarchical approach (without the broad class layer) achieved the same performance as the highest performing single-layer approach, which allowed a maximum fragment length of 6.[8] The mean fragment length of the segmentations used in the test set by the single-layer approach was 3.7, while the mean grapheme length used by the hierarchical approach was only 1.2. The hierarchical approach is capable of reversible generation using about 32,000 parameters, while the single-layer approach requires 693,300 parameters (a 20-fold increase) for uni-directional letter-to-sound generation. In order to achieve reversibility, the number of parameters would have to be doubled.

DISCUSSION

Our current work demonstrates the utility of a hierarchical framework, which is relatively rich in linguistic knowledge, for bi-directional letter-to-sound/sound-to-letter generation. The use of layered bigrams in our hierarchy is extendable to encompass natural language constraints [10], prosody, discourse and perhaps even dialogue modeling constraints on top, as well as phonetics and acoustics at the bottom. As such this paradigm should be particularly useful for applications in speech synthesis, recognition and understanding.

The versatility of our framework can lead to a variety of applications. These range from a lexical representation for large-vocabulary recognition, which provides semantic information, syntactic information, and a clustering mechanism for fast match [12], to a low-perplexity language model for character recogition tasks, where our system gives a test set perplexity of 8.0 as constrasted with 11.3 from a standard letter bigram. In the near future, we plan to report on our robust parsing mechanism for extending coverage, and to experiment with alternative search strategies in the layered bigrams framework.

[8] We did not investigate the cases where maximum word fragment lengths are set beyond 6, due to computational limitations, and the vast number of training parameters required.

REFERENCES

[1] S. Hunnicutt, H. Meng, S. Seneff and V. Zue, "Reversible Letter-to-Sound Sound-to-Letter Generation Based on Parsing Word Morphology," pp. 763-766, *Proceedings, European Conference on Speech Communication and Technology*, September, 1993.

[2] H. Meng, S. Seneff and V. Zue, "Phonological Parsing for Reversible Letter-to-Sound/Sound-to-Letter Generation," *Proc. ICASSP-94*, April, 1994.

[3] N. Chomsky and M. Halle, *The Sound Pattern of English*, Harper & Row, 1968.

[4] T. J. Sejnowski and C. R. Rosenberg, "NETtalk: Parallel Networks that Learn to Pronounce English Text," *Complex Systems*, 1, 1987.

[5] J. M. Lucassen and R. L. Mercer, "An Information Theoretic Approach to the Automatic Determination of Phonemic Baseforms," pp. 42.5.1-42.5.4, *Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing*, 1984.

[6] S. M. Lucas and R. I. Damper, "Syntactic Neural Networks for Bi-directional Text-Phonetics Translation," pp. 127-141, in *Talking Machines, theories, models and designs*, edited by G. Bailly and C. Benoit, North-Holland publishers.

[7] F. Alleva and K. F. Lee, "Automatic New Word Acquisition: Spelling from Acoustics," pp. 266-270, *Proceedings of the Darpa Speech and Natural Language Workshop*, October, 1989.

[8] J. Allen, S. Hunnicutt and D. Klatt, *From Text to Speech: The MITalk System*, Cambridge University Press, 1987.

[9] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications", *Computational Linguistics*, Vol. 18, No. 1, pp. 61-86, March 1992.

[10] S. Seneff, H. Meng, and V. Zue, "Language Modelling for Recognition and Understanding Using Layered Bigrams," pp. 317-320, *Proceedings, Second International Conference on Spoken Language Processing*, October, 1992.

[11] H. Kucera and W. N. Francis, *Computational Analysis of Present-Day American English*, Brown University Press, 1967.

[12] D. W. Shipman and V. W. Zue, "Properties of large lexicons: Implications for advanced isolated word recognition systems, *Proceedings, ICASSP-82*.

[13] H. Meng, S. Seneff and V. Zue, "The Use of Higher Level Linguistic Knowledge for Letter-to-Sound Generation", to appear in *Proceedings, International Symposium on Speech, Image Processing and Neural Networks*, April, 1994.