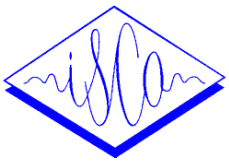

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

DIAGNOSTIC PERCEPTUAL EXPERIMENTS FOR TEXT-TO-SPEECH SYSTEM EVALUATION

Jan P. H. van Santen

Linguistics Research Department
AT&T Bell Laboratories
Murray Hill, NJ, 07974, U.S.A.

ABSTRACT

Perceptual methods are described for detecting problems in components of text-to-speech systems. One feature of our approach is the central role played by text-based computational procedures. Their primary use here is in the form of automatic methods for generating text materials with maximal coverage of the domain of a text-to-speech system. Other uses include a system for automatic phonemic scoring of orthographic responses in transcription tasks. A second important feature of our approach is usage of multiple perceptual paradigms. This is needed both to capture the inherent perceptual multidimensionality of synthetic speech and to handle the multiplicity of problems caused by different system components.

10.21437/ICSLP.1992-128

I. INTRODUCTION

Diagnostic text-to-speech system evaluation involves detecting weaknesses either in specific elements (e.g., a particular spelling-to-sound rule) or in broad components (e.g., the duration rule component), with as goal to improve a system [17, 15]. By contrast, comparative evaluation involves measuring multiple systems on a single decision axis, with as goal to discover which system is best [8, 12].

There are at least three factors that make diagnostic evaluation difficult. First, *perceptual multidimensionality*. Generated speech can be measured on many quasi-independent perceptual dimensions, including, to mention just a few, comprehensibility, segmental intelligibility, smoothness, and voice source quality. An experiment that concentrates on only one or two dimensions may miss important deficiencies. For example, intelligibility of single words provides no information on the intonation component. Second, *system complexity*. Current text-to-speech systems consist of multiple components; moreover, these components have a many–to–many relation with perceptual dimensions. For example, the perception as "pat" of what was intended to sound as "plat" may result from a spelling–to–sound rule error, an error in the duration for /l/, or (in concatenative synthesizers) an error in the "{pla}" unit; conversely, a bad concatenative unit can cause both intelligibility and smoothness problems. Third, the *coverage problem*. The domain of a text-to-speech system is usually the entire language, whereas in an experiment at most a few hundred sentences or words can be presented. Thus, experiments face a double generalization problem – from the listener sample to the listener population, and from the text sample to the language. The latter is particularly hard because of the extremely uneven frequency distribution of most types of linguistic "units" – whether words [3], feature vectors describing contexts relevant for segmental duration [13, 14], or acoustic units in concatenative synthesis [15]. How can we generate text that provides complete, or at least maximal, coverage of a particular type of "unit"?

This paper makes two key points. One is that the problems of perceptual multidimensionality and system complexity require that we have an arsenal of paradigms that each address different dimensions and components. Towards this end, some new, or improved, experimental paradigms are described. The second point is that text-based algorithms borrowed from computational linguistics have many useful applications in diagnostic evaluation. The most important application is to the coverage problem, which is addressed here with automatic procedures for generating text with maximal coverage of a selected type of linguistic "unit". Another application is for automatic phonemic scoring of orthographic name transcriptions.

II. TEXT GENERATION PROCEDURES

II.1 "Greedy" algorithms

The current version of the AT&T Bell Laboratories Text-to-Speech system (*TTS*) uses over 2,500 concatenative units [7, 15]. How does one generate text that covers each unit at least once using at most a few hundred *items* (sentences, names, numbers)?

It is assumed that by searching a sufficiently large text corpus we can construct a list of items that jointly use each unit at least once. Since typically such a list is quite long, we must now solve the problem of finding the shortest sublist that still covers all units. This problem is a special case of a standard combinatorial optimization problem (the "set-covering problem"), that is often solved with *greedy algorithms* ([5], p. 975).

In the present context, a greedy algorithm works as follows. As first item, we select the item with the largest number of units, not counting units that are repeated within the same item. After n items have been selected, we select as $n+1$–th item the one that has the largest number of units that have not yet been covered.

In an application of the algorithm to 67,440 items covering 2533 units, the greedy algorithm produced a list of 653 items with complete coverage of all units. (These items were used in the word pointing paradigm discussed below.) Random selection – where the list of 67,440 items was randomly permuted and then processed item for item in order of occurrence – required 64,516 items. In other words, the greedy algorithm reduced the number of items by almost two orders of magnitude.

The same procedure can be applied to many other "units". For example, for coverage of the conditions distinguished by the

duration component we take as units feature vectors that describe the context of each segment in a sentence. Other examples include: words (for maximizing the vocabulary of a set of sentences), phonemes (e.g., only 24 words are needed to cover 1429 consonant-vowel-consonant words in terms of onset consonants, medial vowels, and final consonants), diphones (to maximize coverage of segmental transitions; this is useful for testing parametric synthesizers), and various types of feature vectors – e.g., discourse features that predict pitch accent assignment [6].

The greedy algorithm can be amended in various ways. For example, one can maximize coverage of high frequency units by using the sum of the frequencies of the units in an item as criterion (*frequency-weighted* greedy algorithm). This is important when complete coverage is impossible, in which case one likes to cover the most frequent units first.

II.2 Tools from computational linguistics

Automatic text generation critically relies on the availability of large quantities of on-line text. Currently, it is not difficult to obtain corpora consisting of over ten million sentences and hundreds of millions of words.

Certain paradigms require automatic pronunciation generation, for which we used the pronunciation component of *TTS*; it is based on very large pronunciation dictionaries and, while it also contains rules, uses the dictionary for over 99 percent of input words [4]. Other tools borrowed from computational linguistics include Church's *parts* program [2], which computes parts-of-speech.

It should be emphasized that for automatic text generation errors made by these tools are usually not a serious problem, because the final result is always a relatively small list of items that can be inspected manually.

III. SOME EXPERIMENTAL PARADIGMS

III.1 Word Pointing Paradigm: testing concatenative units

Speech generated by concatenation can have a wide variety of problems, including problems in intelligibility, discontinuities, and missing or seriously distorted segments. Hence, a broad paradigm is needed where listeners can indicate any of a number of problems. Because of the large number of concatenative units in *TTS*, the greedy procedure discussed in Section II.1 plays a vital role. Since the acoustic units of *TTS* often span word boundaries, occur in unstressed syllables, or involve consonant clusters, the test materials cannot be confined to the ubiquitous consonant-vowel-consonant words presented in isolation; to the contrary, they must involve polysyllabic words embedded in multi-word sequences – as in the 653 sentences, names, and numbers discussed above.

In the paradigm, listeners see and hear (up to three times) a nonsense sentence, and, using simple keyboard commands, first highlight problematic words and then rate the seriousness of the problems on a 2-point scale.

Statistical analysis of two experiments using this procedure showed that listeners were able to perform the task reliably. First, the correlations between the rating responses of different listeners were highly significant (the null hypothesis that the between-listener correlations for the quality rating responses were zero was rejected at $p < 10^{-6}$, and almost all between-listener correlations were significant at $p < 0.01$). Second, listeners tended to agree which words in a sentence were problematic (the null hypothesis that the listeners randomly pointed at words was rejected at $p < 10^{-3}$, based on randomization tests).

After acoustic units occurring in words rated by the listeners as problematic had been replaced, a dramatic improvement in perceived quality was found. This was established in a paired comparison task in which, on the same items as in the pointing paradigm, the standard version of *TTS* was compared with the improved version.

In summary, the word pointing paradigm in conjunction with greedy algorithms enables one to inspect the quality of all concatenative units in an inventory, even in systems with as many as 2,500 units. Note, however, that concatenative problems at times involve *unit incompatibility*, where a unit causes problems only in conjunction with specific other units. Detection of unit incompatibility requires coverage of *unit pairs*, for which substantially larger bodies of text are needed.

III.2 The Minimal Pairs Intelligibility Test: An expanded version of the DRT

The Diagnostic Rhyme Test (*DRT* [16]) is a two-alternative forced choice intelligibility test that is frequently used in speech coder evaluation. Unfortunately, the DRT seriously underrepresents the domain of text-to-speech systems because it measures intelligibility only of consonants in word-initial position in isolated consonant–vowel (–consonant) words (forming *minimal pairs* that differ in only one *phonetic feature*, such as the voicing feature in "feel" vs. "veal"). For example, the DRT domain does not allow testing of concatenative units in *TTS* that are used only at word boundaries, in consonant clusters, or in unstressed syllables.

We constructed the Minimal Pairs Intelligibility Test (*MPIT*), consisting of a list of 256 pairs of meaningless *sentences* that form minimal pairs in the sense of differing in exactly one segment. For example, on a given trial a listener would hear the sentence:

"The uniform towels snitch a sniffer"

and, without repetitions, has to choose between visually presented alternatives ("uniform" vs. "uniformed"). A different listener would hear the companion sentence, where "uniform" is replaced by "uniformed". The complete list is given in [15].

Of critical importance is that this list covers many "domain categories", as defined in terms of phonetic features, segment location, word location, and other factors:

1.
   - Consonant substitutions ("copper" vs. "chopper").
   - Vowel substitutions ("tutor" vs. "teeter").
   - Consonant insertions/deletions ("attitudes" vs. "altitudes").
2.
   - One-feature substitutions ("ringers" vs. "riggers")
   - Two-feature substitutions ("burnish" vs. "furnish")
3.
   - Word-initial ("gaskets" vs. "baskets"; "isolation" vs. "oscillation"; "riskiest" vs. "friskiest").
   - Word-internal ("musty" vs. "musky"; "chipper" vs. "cheaper'; "defected" vs. "deflected").
   - Word-final ("familiar" vs. "familial"; "advance" vs. "advanced"; omitted for vowels).

4. • Stressed syllables ("revolve" vs. "resolve"; "founder" vs. "flounder"; "perimeter" vs. "parameter").
   • Unstressed syllables ("maternity" vs. "paternity"; "contact" vs. "contract"; "insure" vs. "ensure").
5. • Intervocalic consonants ("losers" vs. "louvers"; "likability" vs. "liability").
   • Non-intervocalic consonants ("clutter" vs. "flutter"; "declaim" vs. "disclaim").
6. Adjectives vs. plural nouns vs. plural verbs vs. singular nouns.
7. Sentence-medial vs. –internal vs. –final words.

These minimal pairs were found with text corpora tagged for parts-of-speech, pronunciation algorithms, and special-purpose algorithms for finding minimal pairs for the various domain categories.

The same phonetic features were used as in the DRT. The choice of a feature system is not critical, especially for testing concatenative synthesizers that are not organized on a featural basis. For example, the /*p*/ in /*pat*/ may sound like /b/ not because a system has a general problem with the voicing feature, but simply because the "{pa}" unit is bad. There is no entity or component in the system that corresponds to the general voicing feature, so that this feature is not diagnostic. In the MPIT, the role of features is to help produce a set of phoneme pairs that are sufficiently confusable to produce measurable error rates. From a diagnostic perspective, the other factors defining the domain categories (e.g., within-word location, being in clusters) are at least as important as phonetic features. For example, certain systematic changes have occurred in *TTS* in terms of how consonant clusters are handled.

In an application of the MPIT to various text-to-speech systems and natural speech, it was found that no errors were made for word-initial stressed intervocalic consonants. In other words, *precisely the sublist that corresponds to the DRT domain* produced an absolute ceiling effect. Substantial errors were made on consonants in clusters and on unstressed vowels, indicating that the DRT overestimates speech intelligibility. Still more interestingly, performance of different systems in one domain category was generally not predictive of performance in other categories. For example, the 1991 version of *TTS* was worse than the 1987 version on consonants (1-feature substitutions), yet better averaged over all categories.

The results show that useful minimal pairs lists can be automatically generated with text based procedures. The same procedures can be applied to other feature systems or to different domain categories (e.g., accented vs. de-accented words).

III.3 Automatically scored orthographic name transcription task

Minimal pairs methods have two drawbacks. First, they are inefficient in that each presentation provides information about the intelligibility of only one segment and hence, in the case of concatenative synthesizers, about the quality of only one acoustic unit. Tasks in which the listener is asked to *transcribe* the entire utterance are more efficient because – ignoring the possibility that listeners use redundancy – each segment in the utterance is tested. A second drawback is the closed response format, making unusual errors (e.g., /p/ pronounced as /z/) difficult to catch. Also in this respect transcription tasks are preferable.

However, unless one uses phonetically trained listeners, a practical problem arises: scoring orthographic responses for phonemic accuracy. Below, a scoring system is described and applied to transcriptions of names.

Why names? The relative unpredictability of names has two advantages over ordinary words. First, it reduces listeners' reliance on redundancy and may thus provide a more accurate assessment of segmental intelligibility. Second, it allows a larger variety of materials in terms of the number of distinct di- or triphones covered. Note that the uncertain orthography of names makes phonemic, as opposed to orthographic, scoring all the more critical. For example, one does not want to count it as an error when a listener gives response "Szymanski" to text-to-speech input "Szymansky".

The input to the scoring system consists of the phoneme string as intended by the text-to-speech system being tested (the *target string*) and an orthographic response of a listener to the uttered phoneme string. How do we, in the face of the phonemic ambiguity of orthography, infer from the orthographic response whether the listener heard the segments correctly?

The first step consists of generating a *set of phonemic interpretations* of the orthographic response. We adapted an algorithm originally developed for predicting allophonic variations from phoneme strings [9]. The algorithm computes the n most likely pronunciations for a given orthographic input, where n is under user control. For the purpose of name transcription, it was trained on a manually transcribed corpus of 50,000 names.

The second step consists of selecting one of these phonemic interpretations as the interpretation intended by the listener (the *intended response*), and measuring its discrepancy from the target string. The selection is based on the principle of *giving the listener maximal benefit of the doubt* by selecting as intended response the phonemic interpretation closest to the target string. To illustrate, suppose that the input to the text-to-speech system is "Michael Riley", resulting in target string /maʲkəl raʲlɪ/. When a listener gives "Nichal Riley" as orthographic response, the transcription system generates phonemic interpretations such as:

1. /naʲkəl raʲleʲ/
2. /naʲkəl raʲlɪ/
3. /nɪčəl raʲleʲ/
4. ...

The principle dictates that we should select as intended response the best-matching interpretation, here /naʲkəl raʲlɪ/. As string match measure we use the *cost* as defined by the standard Levinshtein-distance string alignment algorithm [10]. This algorithm requires assigning weights to error categories, which here are based on a 3-point confusability index (low, medium, high) derived from Voiers' [16] feature system.

In the final output, the system shows for each segment in the target string the associated segment in the intended response and the cost. This allows one to make the same detailed analysis of error patterns as in the MPIT.

When applied to various text-to-speech systems and natural speech, it was found that medium- and low-confusability substitutions were rare for vowels (less than 6 percent error rate) and even rarer for consonants (less than 2 percent error rate). High-confusability substitutions were quite frequent for 0-stressed vowels for text-to-speech systems, but relatively rare for natural speech (one-fourth as likely), which may indicate poor treatment of reduced vowels. Of further diagnostic importance was that,

comparing the 1991 and 1987 versions of *TTS*, increases in error rate were found for high-confusability consonants in unstressed syllable initial consonant clusters and word-final consonant clusters. These results parallel those obtained with the MPIT.

In terms of overall error rates, the 1991 version of *TTS* was found to produce significantly fewer errors than the 1987 version of *TTS*, natural speech (whether or not re-synthesized with a scheme developed by Talkin and Rowley [11]) fewer than 1991 *TTS*, and natural speech (not re-synthesized) fewer than re-synthesized natural speech. The latter is diagnostically important, because the same re-synthesis scheme is used in *TTS*.

How valid is the scoring system? The predictable effect of confusability level on error rates provides a degree of validation of the procedure, as does the similarity of the results to those obtained with the MPIT and the pattern of overall error levels of the different speech versions. In addition, spot checking indicated that on less than 1.5 percent of the responses the orthographic response was phonetically correct, but did not generate the correct phonemic interpretation due to unorthodox spelling. Finally, an exhaustive search indicated that the system never generated a phonemic interpretation that matched the target string but was derived from an obviously wrong orthographic response.

In summary, detailed data on segmental intelligibility were obtained with far fewer listeners than in the MPIT, resulting from the fact that each presentation tests on average 12 segments (which is the average number of phonemes per name) compared to only one segment. It would appear that whenever the computational complexity of the scoring system is not an obstacle, the transcription task is a useful alternative to the MPIT.

Usage of the scoring system is not confined to names. Other candidates for automatic scoring include intelligibility tests for meaningless sentences, as have been described, e.g., by Benoit [1].

IV. CONCLUSIONS

The three paradigms are part of a growing array that is currently routinely used to diagnose problems and to measure progress in specific components of *TTS*. Other paradigms include a rating paradigm in which listeners select from a list of nine which problem is most descriptive of their misgivings; a forced choice paradigm with certainty rating; and various "tuning" experiments where listeners determine optimal settings of certain parameters.

These experiments are helping *TTS* development at several levels. At the lowest level, they detect problems in specific acoustic units, duration rules, or dictionary entries. At the highest level, they help set priorities. For example, if experiments show that the dictionary component rarely produces complaints, attention can be focused on other components.

The main point of this paper is that the special problems posed by diagnostic evaluation (perceptual multidimensionality, system complexity, and coverage) can be addressed with new methods based on tools borrowed from computational linguistics and with a multiple-paradigm approach to evaluation.

References

[1] C. Benoit. An intelligibility test using semantically unpredictable sentences: Towards the quantification of linguistic complexity. *Speech Communication*, 9:293–304, (1990).

[2] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, (1988). Association for Computational Linguistics.

[3] K.W. Church and W.A. Gale. A comparison of the enhanced good-Turing and deleted estimation methods for estimating probabilities of english bigrams. *Computer Speech and Language*, 5:19–54, 1991.

[4] C.H. Coker, K.W. Church, and M.Y. Liberman. Morphology and rhyming: two powerful alternatives to letter-to-sound rules for speech synthesis. In *Proceedings of the Second European Conference on Speech Communication and Technology*, pages 83–86, Autrans, France, (1990). ESCA.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachussetts, (1990).

[6] J. Hirschberg. Using discourse context to guide pitch accent decisions in synthetic speech. In G. Bailly, C. Benoit, and T.R. Sawallis, editors, *Talking Machines: Theories, Models, and Designs*, pages 367–376. Elsevier, 1992.

[7] J. P. Olive. A new algorithm for a concatenative speech synthesis system using an augmented acoustic inventory of speech sounds. In *Workshop on speech synthesis*, pages 25–30, Autrans France, (1990). ESCA.

[8] D.B. Pisoni, H.C. Nusbaum, and B.G. Greene. Perception of synthetic speech generated by rule. *Proceedings of the IEEE*, 73:1665–1676, (1985).

[9] M.D. Riley. A statistical model for generating pronunciation networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 1, Toronto, Canada, (1991). ICASSP91.

[10] D. Sankoff and J.B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley, London, (1983).

[11] D. Talkin and J. Rowley. Pitch-synchronous analysis and synthesis for tts systems. In *Workshop on speech synthesis*, pages 55–59, Autrans France, (1990). ESCA.

[12] R. van Bezooijen and L.C.W. Pols. Evaluating text-to-speech systems: Some methodological aspects. *Speech Communication*, 9:263–270, 1990.

[13] J. P. H. van Santen. Deriving text-to-speech durations from natural speech. In G. Bailly, C. Benoit, and T.R. Sawallis, editors, *Talking Machines: Theories, Models, and Designs*, pages 275–285. Elsevier, (1992).

[14] J. P. H. van Santen. Description of contextual factors affecting vowel duration. *Journal of the Acoustical Society of America*, 91(4 [Pt. 2]):2443, (1992).

[15] J. P. H. van Santen. Perceptual experiments for diagnostic testing of text-to-speech systems. *Computer Speech and Language*, 6, (1992 [in press]).

[16] W. D. Voiers, A. D. Sharpley, and C. J. Hehmsoth. Research on diagnostic evaluation of speech intelligibility. Research Report AFCRL-72-0694, Air Force Cambridge Research Laboratories, Bedford, Massachussetts, (1972).

[17] C. E. Wright, M. J. Altom, and J. P. Olive. Diagnostic evaluation of a synthesizer's acoustic inventory. *Journal of the Acoustical Society of America, Suppl. 1*, 79:s25, (1986).