The IBM Expressive Speech Synthesis System

Wael Hamza, Raimo Bakis, Ellen M. Eide, Michael A. Picheny, and John F. Pitrelli

IBM T.J. Watson Research Center, Yorktown Heights, NY, 10598, USA
{hamzaw, bakis, eeide, picheny, pitrelli}@us.ibm.com

Abstract

This paper introduces the IBM Expressive Speech Synthesis system. We describe recent work in improving the quality of our baseline text-to-speech system as well as extending our capabilities to generate expressive synthetic speech. We present results showing improved base quality, especially for sentences drawn from a limited domain. We also demonstrate our ability to convey good news and bad news, produce contrastive emphasis, and ask a question appropriately. In order to facilitate access to the expressive capabilities, we use some of our proposed extensions to the Speech Synthesis Markup Language (SSML).

1. Introduction

As conversational speech interfaces proliferate, their usability becomes increasingly important. Users want to execute transactions efficiently, without being subjected to tedious verbosity, weird sounds, or an excessive cognitive load. Because current text-to-speech (TTS) systems don't adequately satisfy these needs, today's commercial applications tend to use synthesis sparingly, relying mainly on pre-recorded speech.

One complaint about TTS is its "cluelessness"—it may deliver good intonation and prosody on most short items, but in longer passages, or in short but meaning-laden phrases, the system's lack of understanding is often painfully obtrusive. To remedy this problem, the synthesizer

1) needs semantic and stylistic information, not just a sequence of phonemes and syntactic information, and

2) must be able to convey, through prosody and other paralinguistic devices, the additional meaning not explicit in the bare phoneme sequence.

A synthesizer with enough world knowledge can extract meaning from some texts and speak them appropriately. The application designer, nevertheless, like a stage director, must always be able to override the speaker's first attempt at interpretation. For this reason, we don't rely merely on the system's own analysis. Instead, we enable application designers to supply extra information and directions through manually- or automatically-generated markup in the synthesizer input stream.

In this paper we describe the IBM Expressive Speech Synthesis System, which embodies these ideas. Section 2 describes the markup language. Section 3 is an overview of our entire TTS system. Section 4 describes two methods for expressive synthesis, and their respective domains of applicability. Section 5 concludes and discusses future challenges.

2. Extended SSML

The new markup required for the current experiment was implemented in the framework of our previously proposed extensions [1] to the Speech Synthesis Markup Language (SSML) [2]. Specifically, for these experiments we added the new attribute "style" to the "prosody" element.

Styles, such as "conveying good news," affect not only pitch, timing and loudness, but also features not currently addressed by SSML, such as vocal-tract length and glottal waveform parameters. When the speaker smiles, for example, the effective vocal-tract length is smaller than when she pouts. Similarly, voice qualities such as breathiness and creakiness play an important role in expressive elocution, and are, therefore, also legitimate components of prosody [3].

Even if we had full control over all these low-level facets of prosody, it would still be a gargantuan task to build an expressive style like "good news" from them. The difficulty would be comparable to that of painting a photo-realistic portrait given canvas, palette, and brushes. The need for higher-level attributes in the markup language is evident.

In our experiments, therefore, we do not provide additional low-level attributes. Rather, our "style" attribute accepts high-level specifications such as "good-news," "bad-news" or "yes-no question". Thus, we handle marked-up text such as:

<prosody style="good-news">You have received a free upgrade to <emphasis> first class! </emphasis> </prosody>

or:

<prosody style="bad-news">Your flight will be delayed at least four hours because of the typhoon.</prosody>

For "emphasis," the current SSML standard already provides suitable markup; in this paper we address the challenge of implementing it in the synthesizer.

3. System Description

3.1 Building the Voice Database

We direct a professional speaker to record approximately 15 hours of speech in a friendly, energetic style, henceforth referred to as *neutral.* The same speaker reads additional scripts, *e.g.* "conveying good news," "conveying bad news," and "asking yes-no questions," each in the appropriate style. The process of building the voice database differs from our previous one [4] as follows. Each speech segment in the database is labeled by an attribute vector carrying linguistic and expressive information about that segment. For example, all speech segments from the "bad news" script are labeled to have a "style" element with value "bad news." Fig. 1 shows part of the attribute vector defined in our system.

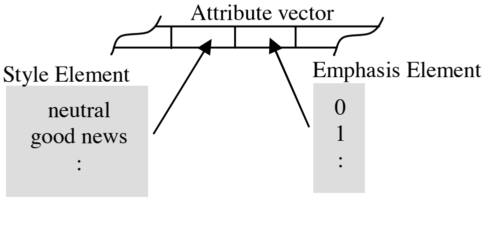

Figure 1: Part of an example attribute vector. Each attribute element takes values from its shaded list.

The attribute vector definition is customizable to the type of the application as well as the availability of linguistic and expressive information of the database segments. For convenience, database segments not labeled for a certain attribute are given the default value of this attribute.

3.2 Run-time Synthesis

During synthesis, the input, which is in the form of an extended-SSML document, is processed by an XML parser. Resulting text is used to form a sequence of targets as described previously [4], each of which contains information about the energy, pitch, and duration to be used in the search. In the current system, the extended-SSML tags are used to form an attribute vector per target, analogous to the one used in the voice-database-building process to label the speech segments.

In addition to the regular target cost function [4], an attribute cost function $C(t,o)$ is introduced to penalize the usage of a speech segment labeled with attribute vector o when the target is labeled by an attribute vector t. This cost function is realized as follows. A cost matrix C_i is defined for each element i in the attribute vector. The cost element $C_i[t_i,o_i]$ indicates the cost to select a speech segment labeled with the attribute o_i when a target attribute t_i is requested. The total attribute cost will be the summation of the individual elements' attribute costs. That is,

$$C(t,o) = \sum_{i=1}^{N} C_i[t_i,o_i]$$

where N is the size of the attribute vector. Table 1 shows an example of $C_i[t_i,o_i]$ for the expressive style element of the attribute vector. The attribute cost is key to the corpus-driven method for generating expressive speech described in Section 4.1.

		Target			
		neutral	**good news**	**bad news**	**...**
Segment	**neutral**	0.0	0.3	0.3	...
	good news	0.7	0.0	1.0	...
	bad news	0.7	$C_i[t_i,o_i]$	0.0	...
	...	...	...	...	0.0

Table 1: Example of an attribute cost matrix. Here, 0.7 is the cost of using a "good news" segment when the target label is "neutral."

The output speech is constructed from the search-generated segment sequence as follows. To avoid large pitch modification to the original signal, a piecewise linear connection of the observed end pitch of each selected segment is constructed. This contour is smoothed [4] to eliminate any rapid fluctuations which sound as if the speaker were shaking or distressed. Speech is then generated using a new algorithm, *contiguous bypass*, which aims to minimize distortion introduced by signal processing by bypassing this processing in cases in which the engine is using a sequence ("chunk") of segments which were contiguous in the original recordings. Using a signal processing algorithm similar to Frequency Domain Pitch Synchronous Overlap Add [5], segments not belonging to contiguous chunks are then modified in pitch to match the smoothed pitch contour and, optionally, in duration to match the target duration. Contiguous chunks are broken into two parts, "internal" segments and "boundary" segments. The internal segments are copied sample by sample to the output buffer, while boundary segments on either side provide a smooth transition between the sample-by-sample copy and our regular signal processing. This algorithm eliminates signal processing distortion in large contiguous segment chunks and provides a seamless transition to regular signal processing. The contiguous bypass algorithm is applied only to contiguous chunks that exceed a specified number of segments. This constraint prevents distortion resulting from switching too frequently between regular signal processing and sample-by-sample copying. The overall cost function is tuned to bias toward selecting long contiguous chunks; doing so makes the contiguous bypass algorithm more effective.

3.3 Results

In order to measure the improvements over our previous system, we conducted a listening test. The test material was presented to 12 male and 12 female native speakers of North American English. Stimuli consisted of 30 "neutral" sentences generated from each of three sources: our previous system, our new system, and natural speech from the corpus speaker. We divided the test sentences into two categories: in-domain and out-of-domain. In-domain sentences contain material such as weather reports and travel information, where similar but not identical sentences were found in our original script. Out-of-domain sentences are general and bear less resemblance to the content of the corpus.

All listeners heard all stimuli, with order randomized, and were asked to rate the overall quality of the speech they heard on a 1-to-7 scale. We measured our performance on the in-domain and out-of-domain sentences separately, as shown in Table 2. Note that while we made good progress on out-of-domain sentences, closing 12% of the gap between our system and natural speech, we made more striking improvement on sentences which were similar to the sentences in our corpus, closing 46% of the gap between our system and natural speech. That result is consistent with our expectations; the techniques described in this section aim to increase the length of contiguous passages retrieved from the database and to limit the processing, thereby increasing the naturalness, of those passages. In-domain sentences provide the opportunity to obtain longer contiguous passages than out-of-domain sentences, and therefore benefit more from the new algorithms.

System	Mean opinion score on 1 to 7 scale	
	Out-of-Domain	In-Domain
Baseline System	3.40	3.81
Current System	3.71	4.92
Natural Speech	5.94	6.24
Improvement	12%	46%

Table 2: Listening test results

4. Expressive Speech Synthesis

We are currently exploring two complementary approaches to add expressiveness to concatenative synthesis. The *corpus-driven* approach is to collect corpora for each desired expressive style, and train synthesis models, *e.g.* for f_0 and duration prediction, separately for each corpus [6], [7]. Then these models may be switched in as required. This approach has the advantage of not relying on any linguistic framework to categorize prosodic events. Further, it provides speech for each expressive style, which may yield more natural synthesis with less processing than speech drawn from another expressive style would.

However, this approach also poses significant obstacles. Multiplying corpus-collection costs by a number of styles greatly increases development costs, and this approach does not readily lend itself to overlapping styles, *e.g.* apologetic questioning. So we also pursue a *prosodic-phonology* approach, in which we statistically model acoustic parameters such as f_0 and phone duration on a single, large corpus, in terms of a set of prosodic labels taken from a prosodic phonology, such as ToBI, as well as in terms of other features such as phonetic context. Then we implement expressive styles as a type of dictionary, relating styles to sequences of these prosodic labels using rules learned from a small corpus. While this method addresses the problems of the corpus-driven approach, it introduces its own obstacles, namely, the well-known difficulties in optimizing manually-determined rules, the substantial task of prosodically labeling a corpus, and the relative immaturity of prosodic phonology.

These approaches are not mutually exclusive; we envision that using style-specific f_0 and duration models which incorporate prosodic-phonology features could outperform either approach alone.

Some expressive styles seem more amenable to the corpus-driven approach than the prosodic-phonology approach, and *vice versa*. For example, “conveying good news" might be better realized by a corpus-driven approach, due to its complex, systemic effect on the speech signal, while “emphasis" seems more amenable to the prosodic-phonology approach because of its simpler, localized manifestation. Accordingly, the corpus-driven approach is pursued here to synthesize good news, bad news, and questions, while we use prosodic phonology for contrastive emphasis.

4.1 Corpus-Driven Expressive Synthesis

As mentioned in Section 3, corpora in three expressive styles, conveying good news, conveying bad news, and asking a question, are recorded from the same speaker that recorded the general corpus. From the expressive corpora, we build separate f_0 and duration models for each expressive style, using the same statistical approach used to build the general f_0 and duration models mentioned in Section 3. We use the attribute-cost matrix described in Section 3 to penalize using segments labeled with certain expressive styles when other expressive styles are requested. We manually tune the values in this cost matrix.

In order to synthesize a given style, we use the prosodic models built from the database in the given expressive style. In addition to building prosody models from each style, we include the small set of segments from each of the styles in the search, motivated by the fact that prosody alone does not fully convey the desired style [6]. All segments from all styles are considered in the search, weighted by their attribute costs. Should we increase the size of the expressive databases, we would expect the cost of substituting one style for another would need to be increased. However, in the current system, the expressive databases are small, and the quality of the synthesis is improved by allowing neutral segments, as well as segments from other styles, in the search. In doing so, we trade-off the degree to which the desired style is conveyed by the spectral qualities of the segments chosen to comprise the synthetic utterance against the smoothness and overall quality of the synthesis.

We performed three separate listening tests to measure our ability to generate good news, bad news, and yes-no questions respectively using this corpus-driven approach. Each test was presented to 16 male and 16 female native speakers of North American English. Each listener heard a pair of audio files for each sentence; one member of the pair was the sentence spoken in the default/neutral style, and the other member of the pair was the sentence spoken in the expressive style being tested. The order of the members was randomized across sentences. Each sentence was text that could be appropriately spoken in the expressive style being tested, *i.e.* sentences such as "you have the winning number" went into the "good news" test, while sentences such as "I dented your new car" went into the "bad news" test.

For each pair of sentences, each listener was asked which one sounded more like it was spoken with the desired style. For example, for the yes-no questions, each listener was asked which of the stimuli, the question generated from the default system or from the system expressing questions, sounded more like a question.

Shown in Table 3 are the results of the tests. In the right-hand column is the percentage of responses for which the expressive stimulus was identified as sounding more like the desired style than did the baseline, neutral stimulus.

style	Percent correct
Bad news	70.2
Good news	80.3
Yes-no questions	84.6

Table 3: Corpus-driven listening test results

4.2 Prosodic-Phonology Expressive Synthesis

This approach is based on the theory that prosody is encoded as linguistic units which link elements of meaning to signal acoustics in a way analogous to how a segmental phonology links meaning to acoustics. For example, pluralness is typically modeled not by a direct acoustic modeling, but rather through rules which place /s/ or /z/ segments at the end of many words and have exceptions for anomalous pairs like “ox”/“oxen”. Similarly, we seek to correlate styles such as “contrastive emphasis” with patterns of linguistic units of prosody, and then develop statistical acoustic-prosodic models which in turn relate those units to signal parameters, much as statistical acoustic-phonetic models represent the acoustics of *e.g.* /s/ independent of whether the /s/ represents pluralness. In this framework, addition of a new style merely requires additions to the rules relating styles to prosodic labels, not collection of a new speech corpus. This is analogous to the addition of new words merely requiring new dictionary entries rather than new recordings of specific words.

We choose American English Tones and Break Indices (AmE-ToBI) [8], [9] as the prosodic inventory, as it appears to represent a reasonable consensus of researchers in English prosody, and it exhibits reasonable consistency among transcribers [10], [11]. AmE-ToBI analyzes intonation in terms of a hierarchy of intonational phrases with edge tones such as L%, H%, L- and H-, and pitch accents such as H*, L*, and L*+H. AmE-ToBI also represents the degree of disjuncture between adjacent words with labels like 4 for a full intonational phrase break, and 1 for most phrase-internal word boundaries.

For our experiments, about 1/3 of the corpus was hand-labeled using AmE-ToBI. The corpus was used to train the f_0 and duration models, with a “missing" value used for AmE-ToBI features in the unlabeled data. AmE-ToBI features for the given syllable and four neighboring syllables are included when developing the f_0 and duration prediction models. We model the AmE-ToBI parameter HiF0 by training the models to predict f_0 / HiF0, and then we multiply the predicted value at run-time by a HiF0 determined by rule, thereby improving modeling compared to our previous study [12].

To determine the patterns relating styles to AmE-ToBI patterns, we collected a very small corpus consisting of a few declarative sentences spoken by 20 professional speakers. We analyzed this corpus, finding that a contrastively-emphasized word consistently has at least intermediate prosodic phrase boundaries on each side of the word, accompanied by break indices of at least 3. The word is marked with pitch accent H* and phrase accent L-. Finally, the pitch range of this one-word intermediate phrase is conspicuously high; the high pitch in the phrase, marked by HiF0, averages 28 percentile points higher in the speaker's pitch range than the intermediate phrase containing the word produced neutrally. We put rules in our system accordingly.

We generated 48 questions/answer pairs, differing in one word by which the answer addresses the question. We then synthesized two versions of the answer, one in which that word is (appropriately) marked for contrastive emphasis, and another in which another word is instead (inappropriately) marked for contrastive emphasis. An example is:

Q: Mice have thirty-two muscles in each ear?
A1: CATS have thirty-two muscles in each ear.
A2: Cats have THIRTY-two muscles in each ear.

Listeners read the question, then heard the two answers and were asked which was spoken more appropriately. In this case, "cats" is the word which should be emphasized, and so the answer indicating perception of the appropriate contrastive emphasis would be answer 1. Sentence order within stimulus, and stimulus order, were balanced across 32 listeners, who each heard the 48 stimuli in a single session after three practice stimuli. Overall, listeners identified the correctly-contrastively-emphasized sentence 82.6% of the time.

5. Conclusion

We introduce expressiveness to our concatenative synthesis system through a variety of techniques each of which can be invoked under suitable circumstances. We also introduce several algorithms which improve the general quality of our concatenative synthesis. When the corpus bears some resemblance to the text to be synthesized, we bias the segment search to favor selecting long contiguous chunks, and then we suppress signal processing within those chunks, preserving the speaker's natural expressiveness and making a substantial improvement in perceived quality. When possible to collect a corpus in a desired expressive style, synthesizing even unrelated material in the same style benefits from f_0 and duration models trained on such a corpus. When there is no concatenative corpus associated with an expressive style or a specific type of content, we can employ a prosodic-phonology approach to provide expressive synthesis using the general corpus. In the future, we intend to pursue refinements and combinations of these techniques to realize further improvements in expressive speech synthesis.

6. Acknowledgements

The authors would like to thank Andy Aaron and Tim Mathes at IBM Research for directing the speaker and managing the recording of our corpora, Lesley Carmichael of the University of Washington for her work on ToBI-labeling our data, and Allen Delmar of IBM Research for running our listening tests.

7. References

[1] Eide, E., *et al.*, "Multilayered Extensions to the Speech Synthesis Markup Language for Describing Expressiveness," *Proc. Eurospeech 2003*, Geneva, Switzerland.

[2] http://www.w3.org/TR/speech-synthesis the Speech Synthesis Markup Language, Version 1.0.

[3] Ní Chasaide, A. and C. Gobl, "Voice Quality and f_0 in Prosody: Towards a Holistic Account," *Proc. Speech Prosody 2004*, Nara, Japan, March 23-26, 2004.

[4] Eide, E., *et al.*, "Recent Improvements to the IBM Trainable Speech Synthesis System." *Proc. ICASSP 2003*, Hong Kong, Vol. 1, pp. 708-711.

[5] Moulines, E. and F. Charpentier, "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones," *Speech Communication*, Vol. 9, No. 5-6, 1990, pp. 453–467.

[6] Bulut, M., S. Narayanan, and A. Syrdal. "Expressive Speech Synthesis Using a Concatenative Synthesizer," *Proc. ICSLP 2002*, Denver, CO, USA.

[7] Eide, E., "Preservation, Identification, and Use of Emotion in a Text-to-speech System," *IEEE Workshop on Speech Synthesis*, September, 2002. Santa Monica, CA, USA.

[8] Silverman, K., *et al.*, "TOBI: A Standard for Labeling English Prosody," *Proc. ICSLP*, Banff, Alberta, Canada, October 13-16, 1992, Vol. 2, pp. 867-870.

[9] http://www.ling.ohio-state.edu/~tobi the ToBI official website.

[10] Pitrelli, J. F., M. E. Beckman, and J. Hirschberg, "Evaluation of Prosodic Transcription Labeling Reliability in the ToBI Framework," *Proceedings of ICSLP*, Yokohama, Japan, Oct. 1994, v. 1, pp. 123-126.

[11] Syrdal, A. K., and J. McGory, "Inter-Transcriber Reliability of ToBI Prosodic Labeling," Proceedings of ICSLP, Beijing, China, October 2000, v. 3, pp. 235-238.

[12] Pitrelli, J. F. and E. M. Eide, "Expressive Speech Synthesis using American English ToBI: Questions and Contrastive Emphasis," *Proc. IEEE ASRU 2003*, St. Thomas, U.S. Virgin Islands, December 1-4, 2003.