Automatic Exploration of Corpus-Specific Properties for Expressive Text-to-Speech: A Case Study in Emphasis.

Raul Fernandez and Bhuvana Ramabhadran

IBM TJ Watson Research Center
Yorktown Heights, NY 10598
{fernanra,bhuvana}@us.ibm.com

Abstract

In this paper we explore an approach to expressive text-to-speech synthesis in which pre-existing expression-specific corpora are complemented with automatically generated labels to augment the search space of units the engine can exploit to increase its expressiveness. We motivate this data-discovery approach as an alternative to an approach guided by data collection, in order to harness the full usefulness of the expressiveness already contained in a synthesis corpus. We illustrate the approach with a case study that uses *emphasis* as its intended expression, describe algorithms for the automatic discovery of such instances in the database and how to make use of them during synthesis, and, finally, evaluate the benefits of the proposal to demonstrate the feasibility of the approach.

1. Introduction

There has been recent interest in text-to-speech (TTS) research to address the need of speech synthesizers to not just sound natural and intelligible, but also to convey suitable expressions. Rather than being a decorative flourish, it can be argued that producing expressive synthetic speech is fundamental, not only to ensure that there is a match between the linguistic content of the text and the tone of voice in which it is delivered, but also to engage the user and maintain him motivated in the listening experience. This is particularly relevant as we move beyond simple short-prompts interactive scenarios (*e.g.*, a help desk application) toward more challenging, cognitively-taxing uses of text-to-speech technology (*e.g.*, a synthesized news podcast).

The IBM Expressive TTS System [1] is capable of generating speech in expressive styles suitable for conveying good news, conveying bad news, asking a question, or delivering emphasis. The system relies on augmenting its baseline speech corpus with smaller expression-specific corpora of speech, large enough to derive prosody models and to augment the search space with explicitly tagged expressive units. Although this approach works quite well, it is impeded by the fact that expanding the repertoire of expressions, or increasing the size of an existing corpus, is costly in terms of studio time and footprint size. As an alternative to indiscriminate data collection, we recently argued for an approach in which existing databases are exploited for the occurrence of (possibly more subtle) examples of expressions that are already contained in the database [2]. In this paper, we follow this philosophy and apply machine learning algorithms to our speech databases to automatically explore and learn new labels that can be used by the engine at run time to expand the range of its expressiveness. The purpose is not to discount additional data collection as a viable alternative, but rather to motivate exploring the overlap that there may already exist between the existing databases and a given category of interest, before proceeding with a data-collection approach. We will motivate and illustrate this approach with the case study of *emphasis*, an "expression" for which we already have a corpus of suitable recordings which can be used as a basis for training learning schemes.

The organization of this paper is as follows. In Section 2 we present an overview of the expressive component of the IBM TTS System. We discuss the idea of mining attributes from the dataset in Section 3, present and evaluate algorithms for the automatic labeling of emphasis in speech, and discuss how to make use of this output at run time. In Section 4 we evaluate the proposal and discuss results and finally conclude in Section 5.

2. Expressive System Overview

In this section we review only the architectural components of the TTS engine that are responsible for addressing the generation of expressive (acoustic and prosodic) targets and the expressive unit selection at synthesis time. For a more complete overview of the IBM TTS system, the reader is directed to, *e.g.*, [1].

The baseline corpus used to build the core concatenative database (henceforth referred to as the *neutral* corpus) consists of approximately 10 hours of audio recorded from a professional speaker delivered in a lively, friendly style. In addition to the neutral corpus, the system makes use of smaller, expression-specific corpora containing approximately 1 hour of audio. Some of the expressions we have considered are *good news*, *apologies*, *confusion* and *emphasis*. During the concatenative database build process, the synthesis units in the database (which, in the case of the IBM TTS system correspond to subphonemic speech segments aligned with a single state of a typically 3-state HMM) are labeled with a discrete-valued attribute vector containing, *e.g.*, linguistic, expressive and other kinds of information about the units. Fig. 1 contains an example of a 3-dimensional vector that illustrates the kind of attribute information the system could make use of. Each *attribute type* has a default *attribute value* from its value set associated with it (shown underlined in Fig. 1). Synthesis units that are not explicitly labeled are assumed by the engine to bear the default value for the unlabeled attribute in question. The attribute vector definition (*i.e.*, the list of attributes, the value set each is defined on as well as its default value) is fully customizable to the application and can be specified through an external configuration file.

Separate attribute-specific prosody models are also built at this stage. The current implementation of the engine allows for

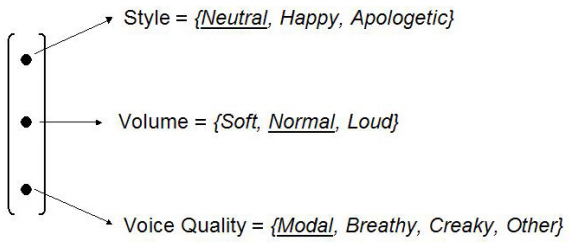

Figure 1: *Example of a 3-dimensional attribute vector, and the attribute value set for each component.*

the *style* attribute illustrated in Fig. 1 to be the attribute dimension that acts as a switch between different prosody models at run time to generate different prosody targets. The full dataset is therefore segregated into separate *style*-specific subsets, and the standard prosody models for pitch, duration (and, optionally, energy) are constructed for each subset [1] .

During synthesis, the input, assumed to be in the form of marked-up text, is processed by an XML parser. The resulting tags are used to assign attribute values to different words, and these values are propagated down to the subphonemic synthesis-unit level. The plain text is then processed by our standard rules-based front-end to produce phonetization and symbolic linguistic descriptions. Acoustic models, previously built while assembling the dataset, are used to generate a suitable list of context-dependent synthesis-unit candidates over which to evaluate the search, and the style-specific prosody models are invoked in order to generate prosody targets as a function of the corresponding style attribute value for each unit. Since prosody alone does not fully convey the desired style [3], we also include the smaller set of segments from each of the attributes in the search to allow the dynamic programming algorithm to evaluate trade-offs between matching different components of the target cost. That is, all segments from all attributes are considered in the search (as long as they fit the context-dependent constraints imposed by the acoustic models), and the attribute match is assessed through an additional component of the cost function, $C(\mathbf{t}, \mathbf{o})$, introduced to penalize using a speech segment labeled with attribute vector $\mathbf{o}$ when the target is labeled with attribute vector $\mathbf{t}$. Since all attributes are discrete-valued, this cost can be summarized by means of a square matrix. The elements of this matrix are usually tuned empirically.

3. Attribute Mining

The previous section highlighted an architecture that makes use of attribute annotations on the synthesis corpus to facilitate attribute-driven prosody target generation and dynamic programming search. We would now like to turn to the source of knowledge for these attribute labels. In one particular case, these attributes could be present in the data by experimental design and their specific occurrence be known *a priori*. Such is the case, for instance, of the approach we have followed in the past for collecting expression-specific corpora [1]: a professional reader is instructed to read text in a particular expressive style (*e.g.*, apologetic) in a recording studio and is closely supervised to make sure she delivers the intended expression. We have recently motivated going beyond this kind of *a priori* knowledge of the descriptors, and moving toward *discovering* them in the corpus to increase the range, and flexibility, of expressive concatenative text-to-speech [2]. A similar approach can be found in the work by Campbell and Marumoto [4], where prosodic and acoustic characteristics associated with different emotions are learned from emotion-specific corpora and then used to relabel segments in other databases. As a particular instance of this approach, consider the scenario where attributes can be expected to be in the dataset, and the occurrence of the different values for that attribute can be arrived at through a learning or rules-based mechanism. Imagine, as an example, labeling the speaking rate of every synthesis unit in the corpus as *slow, medium*, or *fast*. One could establish this discretization by some simple rules given knowledge of *e.g.*, text alignments, phone classes and speaker's average speaking rate. In the most general case of this approach, however, we may or may not have knowledge about whether the attribute is reflected in the dataset, the degree to which it is, and where it occurs. In this case, we wish to *mine* the corpus to discover these attribute values automatically.

In previous work [2], we focused primarily on attributes that could be derived from the text itself (and from the symbolic description thereof produced by the front-end analysis). While this is a reasonable first step, it has the limitation that we ultimately wish to establish properties of the *spoken* synthesis units; using the text string as proxy for analysis can only provide an approximation given the multiple prosodic realizations that can exist for a given syntactic structure [5]. In the work presented here, we are following the approach of mining attributes from the corpus by focusing on automatic discovery of properties of the spoken units. We are illustrating this with a case study in *emphasis*. The motivation for focusing on this type of attribute is manifold: First, emphasis is one of the expressive labels for which we already have an existing smaller corpus of in-studio recordings with professional speakers explicitly instructed to produce it. This corpus can therefore be exploited as learning material for data-driven algorithms; that is, automatically discovering this attribute in the larger synthesis corpus can be bootstrapped to a part of the corpus where we have very high confidence the attribute is present. Secondly, emphasis is a fairly pervasive attribute of spoken language, and, although different speakers can vary in the manner and degree of the realization, we expected that, at least for some of the speakers with a more "lively" reading style, we would be able to find quite a few exemplars in the 10-hour baseline recordings. Finally, being able to properly produce emphasis is applicable to many text-to-speech scenarios where we would like to improve the expressiveness. This includes not just the canonical case of *contrast*, but also cases where you may want to highlight a rare word or increase the liveliness of a sentence by, *e.g.*, treating focus words differently, or speech-to-speech applications where user-intended focus or emphasis in the original language should be preserved and synthesized in the output target language.

3.1. Emphasis Classification

In this section we turn to the details of how to annotate the baseline corpus with emphasis labels. As mentioned above, since we have at our disposal smaller corpora containing emphasis annotations that we can use as training and development data, our approach will be to implement data-driven algorithms for automatically learning a mapping from a set of input predictor features to a binary emphasis label. What is understood in this work by emphasis is primarily a perceptual phenomenon. We are not adhering to any theoretical descriptions of how empha-

sis is accomplished. Rather, we are interested in modeling the characteristics of the speech obtained under the following conditions: A professional speaker is instructed to read a sentence in which some words are meant to be emphasized. When (usually) two judges present in the studio, plus the speaker herself, are satisfied with the outcome (*i.e.*, when the intended words, and only those, are perceived as emphatic), the recorded sentence is added to the emphasis corpus; otherwise, the speaker reads the sentence again. The script is carefully designed to ensure as much as possible that emphasis is requested for words where it would be natural to produce it. This avoids unnatural realizations that would be difficult for the speaker to produce, and which might hurt the quality at synthesis time.

Emphasis is treated here as a binary-valued word-level attribute (*i.e.*, a word is emphasized, or not), and the classification scheme implemented here is based on a set of features extracted at the word level. We realize that emphasis can be a continuous-valued attribute, or, since the architecture presented in the previous section relies on discrete attributes, that it would at least admit a multi-level discrete description rather than a binary one. However, for the purpose of the modeling done in this paper, and the instructions delivered to the speakers, it was treated as a binary variable.

The feature set is meant to capture some variations in pitch, energy and duration (the latter roughly modulated by broad phone classes) which are likely to be acoustic-prosodic correlates of emphasis. The full list of features is given below:

1. Average pitch in word, normalized by speaker's average pitch
2. Median pitch in word, normalized by speaker's average pitch
3. Standard deviation of pitch in word, normalized by speaker's pitch standard deviation
4. Pitch range over word, normalized by speaker's pitch standard deviation
5. Word duration, in seconds
6. Word duration in seconds, normalized by the number of phones in word
7. Word duration in seconds, normalized by the number of vowels in word
8. Ratio of vowel duration to overall duration in word
9. Previous value normalized by the vowels-to-phones ratio in word
10. Root-Mean-Squared energy value of word

Since one of the applications we envision for this kind of system is to be able to label corpora for speakers for whom we do not have any development data, we have avoided highly speaker-dependent features, such as absolute pitch-based features, from this list. However, for the case where the training and testing speaker were the same, we did consider these additional features, only to discover that they did not improve the performance. We have, therefore, omitted them from the final system and from the rest of the discussion.

We explored a variety of classification schemes on this task and found that K-Nearest-Neighbor and Support Vector Machines were the two top performers, in that respective order, over other classifiers like Decision Trees or Naive Bayes. Evaluations were done in all cases using 10-fold cross-validation. The fact that a simple K-Nearest Neighbor (with $K \approx 10$) consistently performs at the top is possibly due to the fact that, given the good amount of data we have, its performance is starting to approximate that of the Bayesian posterior for this feature set. Nonetheless, to explore the possibility of benefiting from classifier combination, we stacked the outputs of these 2 top performers into a combination scheme using a Naive Bayes classifier. The first stage of the training, therefore, maps the input feature space listed above into two (intermediate) estimates, $P_{KNN}(\omega|\mathbf{x})$ and $P_{SVM}(\omega|\mathbf{x})$ of the class posterior probability whereas the second (output combination) stage takes these estimates as input features and maps them to one final class posterior $P_{NB}\Big(\omega\Big|P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\Big)$. A word with feature vector $\mathbf{x}$ is assigned to the class ω which satisfies the Bayes decision rule:

$$\hat{\omega} = \arg\max_{\omega} P(\omega) P_{NB}\Big(P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\Big|\omega\Big) \quad (1)$$

where

$$P_{NB}\Big(P_{KNN}(\omega|\mathbf{x}), P_{SVM}(\omega|\mathbf{x})\Big|\omega\Big) = \frac{P(\omega)}{Z} \times P_{KNN}(\mathbf{x}|\omega) \times P_{SVM}(\mathbf{x}|\omega) \quad (2)$$

and Z is a normalizing constant to ensure the posterior sums up to 1. Analysis of the error distribution of the two intermediate classifiers reveals that there is considerable overlap between their outputs. This lack of complementarity, therefore, limits the usefulness of a classifier combination scheme and does not satisfy the independence assumption on which the success of the Naive Bayes classifier depends. Combining classifiers only offered a modest 2% absolute improvement. However, since the classification is done off-line, we have accepted the extra computational cost in exchange for the minor improvement. The results reported below are all based on the final output of the combining classifier.

We trained and tested two separate speaker-dependent systems, one for a male speaker and one for a female speaker, with 15,204 and 13,278 word tokens respectively. The empirical prior distribution for emphasized words for each set was 22%. Although this number may seem low for a corpus that was expressly designed to collect emphasis, it is challenging to maintain the naturalness and flow of the sentences during the data collection process, as explained above, when emphasized words appear much more frequently than this. Performance was assessed on the training set by means of 10-fold cross-validation. The confusion matrices showing the performance for the two speakers are shown in Tables 1 and 2. The systems achieve an overall recognition rate of 91.17% (male speaker) and 89.86% (female speaker).

		Labeled	
		Emph	Non-Emph
True	Emph	2710	602
	Non-Emph	741	11151

Table 1: Confusion matrix showing the emphasis classification results for the male speaker. Overall recognition rate is 91.17%. Prior class probabilities are $[0.22, 0.78]$ for emphasis and non-emphasis respectively.

A class-dependent analysis of the systems showing different performance measures, Recall (Rec), Precision (Prec), False Positive Rate (FP) and F-Measure (F-Meas), is also shown in

		Labeled	
		Emph	Non-Emph
True	Emph	2295	578
	Non-Emph	768	9637

Table 2: Confusion matrix showing the emphasis classification results for the female speaker. Overall recognition rate is 89.86%. Prior class probabilities are $[0.22, 0.78]$ for emphasis and non-emphasis respectively.

Table 3. For a class ω, these performance measures are defined as follows:

$$\text{Recall} = \frac{\text{Num. correctly labeled } \omega}{\text{Total number of actual } \omega}$$

$$\text{Precision} = \frac{\text{Num. correctly labeled } \omega}{\text{Total number of predicted } \omega}$$

$$\text{False Positive Rate} = \frac{\text{Num. incorrectly labeled } \omega}{\text{Total number of not } \omega}$$

$$\text{F-Measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

We want, in general, to obtain both a high recall and a high pre-

Speaker	Class	Rec	Prec	FP	F-Meas.
Male	Emph	0.82	0.78	0.06	0.80
	Non-Emph	0.94	0.95	0.18	0.94
Female	Emph	0.80	0.75	0.07	0.77
	Non-Emph	0.93	0.94	0.20	0.94

Table 3: Performance figures (recall, precision, false-positives and F-measure) derived from the confusion matrices in Tables 1 and 2.

cision measure (a combined fact that a high F-Measure ought to reflect) while minimizing the number of false positives. As we can see from Table 3, the best results for the Emphasis class are obtained for the male speaker with an F-Measure of 0.8 and a False Positive Rate of 0.06. The results for the female speaker are only marginally different. These numbers suggest that the feature set and classification scheme described here are doing a reasonable job at modeling the emphasis class, while still allowing some room for improvement in future iterations of this work. Further work could explore, for instance, how spectrally-derived features, such as energy in different spectral bands [6], contribute to the realization and perception of emphasis and can aid in its automatic classification.

3.2. Building an Expressive System with Automatic Labels

We can apply the systems proposed in the previous section to the task of discovering examples of emphasis that may occur throughout the rest of the unlabeled baseline database. When we do this, we discover that approximately 8% to 10% of the words in this corpus receive the emphasis label. An empirical subjective analysis of the output of this labeling suggests that the results are better for the case of the male speaker, who speaks in a style that shows more demarcated alternation between emphasized and unemphasized words. The words that are automatically labeled as being emphasized are then given an attribute value that can be used at run time by the framework described in Section 2 to bias the search toward choosing segments with emphasis. Our approach is to keep a distinction between those units that belong to the small emphasis corpus from

		Target			
		neutral	collEmph	labEmph	
Segment	neutral	0.0	0.3	0.2	...
	collEmph	0.5	0.0	0.0	...
	labEmph	0.5	0.1	0.0	...
	...	...	...	...	0.0

Table 4: Attribute cost matrix to combine automatically- and hand-labeled expressions. Automatically-labeled segments are weighted differently than hand-labeled ones to reflect inaccuracies in labeling scheme.

which the classifiers were built, and the units that are automatically labeled. The former are closely scrutinized in the studio during the recording session, and therefore carry a high confidence on the label. The confidence on the latter set is clearly limited and constrained by the performance of the automatic classifiers, which are always prone to have a margin of error. Since the main objective of this work is to increase the inventory of units with a particular attribute from which the search can choose at run time, the prosody models are not rebuilt in light of the additionally labeled data now available. Rather, we retain the models originally developed with the original emphasis data only. However, it remains to be explored whether this additional (noisier) data can offer improved prosody targets.

Fig. 2 summarizes the procedure used to build an expressive system with emphasis: after the steps described in the previous section to build the emphasis classifiers, the baseline corpus is analyzed and augmented with emphasis annotations, and both the annotated baseline and emphasis corpora are combined to produce one final system. Provided we keep the labeled emphasis as a distinct attribute value, the reliability of the annotation scheme can be addressed at run time by choosing weights for the attribute cost matrix that reflect this uncertainty. This is illustrated by the sample cost matrix shown in Table 4. Here *collEmph* is used to describe the emphasis annotations attached to the emphasis-specific corpus collected in the studio whereas *labEmph* describes the annotations produced by the automatic labeling scheme. This is in theory a square matrix although, in practice, the *labEmph* label is unlikely to be requested as an explicit target: A user would mark up the text with a tag that translates to a *collEmph* request directly, not to a request for a target with labeled emphasis. *labEmph* acts as an additional annotation that is tied in some sense to *collEmph* by the system developer behind the user interface layer. However, the engine architecture allows us to directly make this kind of request if, for instance, we wish to test how well these labels alone produce the percept of emphasis. In this example, the 0.1 value in the matrix reflects the fact that, whenever a *collEmph* target is sent to the search, we penalize retrieving a *labEmph* unit slightly more than retrieving a *collEmph* unit (by definition a perfect match, and therefore 0 cost). If we wish to make the two labels equivalent, we can do so by making their two respective row entries identical.

4. Evaluation and Discussion

In order to test the usefulness of the proposed automatic mining approach, we designed a listening test where subjects were presented with pairs comprising one neutral sentence and one sentence containing emphatic words, and asked to make choices about the emphasis-carrying sentence. Three type of sentence

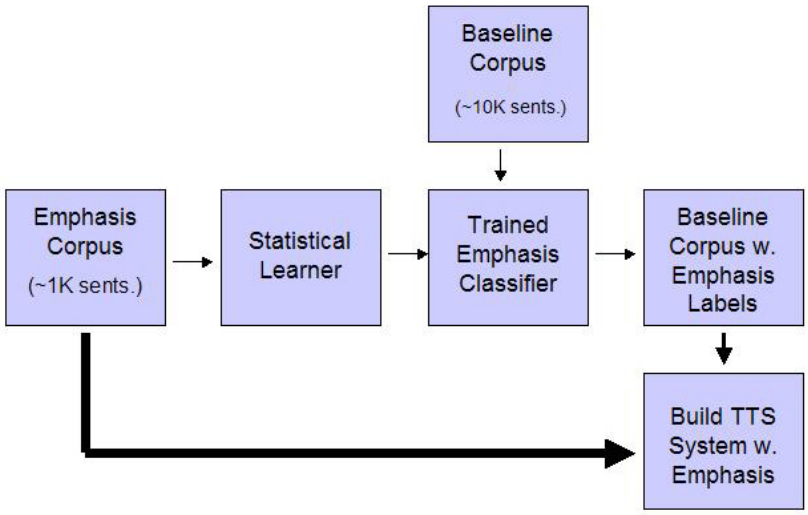

Figure 2: *Building an expressive TTS system with collected and mined expression. The relative width of the two lines feeding into the final system is meant to illustrate the fact that we can assign different weights to these different data subsets based on our confidence of the labels.*

stimuli were used to make up the pairs:

- Type A: a sentence where no word was marked for emphasis was synthesized using the unannotated baseline corpus.
- Type B: a sentence where one or more words were marked for emphasis was synthesized using the unannotated baseline corpus, plus the collected emphasis corpus.
- Type C: a sentence where one or more words were marked for emphasis was synthesized using the baseline corpus automatically annotated with emphasis labels, plus the collected emphasis corpus.

The texts of 12 distinct sentences, each containing one or more words marked-up for emphasis, were synthesized as described above to produce 3 sets of stimuli (when synthesizing Type A sentences, the marked-up emphasis was ignored). Both the Type B and Type C versions of each sentence marked-up the same word or words for emphasis. We tried to mark-up words where emphasis might fall naturally but avoided contrastive-emphasis constructions since in those cases the text alone is often a predictor of where the emphasis should be realized. After generating these basic stimuli, 12 pairs were produced for 2 testing conditions, as follows:

- Condition 1: a pair consisting of 1 sentence of Type A (neutral) and 1 sentence of Type B (emphatic), both with the same text
- Condition 2: a pair consisting of 1 sentence of Type A (neutral) and 1 sentence of Type C (emphatic), both with the same text

A total of 24 pairs, 12 from each condition, were combined to produce one final set of listening samples. All run-time parameters were set to be the same for all conditions. A playlist was created by randomly interleaving the pairs from each condition, and by randomizing the order within each pair. Additionally, a second playlist was assembled by reversing the order within each pair from the first list. Thirty-one listeners took part in the test; 16 listened to the samples in the original order and 15 in the reverse order.

When synthesizing emphasis, we usually resort to making use of very brief pauses (usually in the order of 5 to 10 msecs.) around the emphasized words. This alone often suffices to create the impression of emphasis although the acoustic and prosodic realization of the units that follow are often at odds with this impression if no further emphasis units are used. Since the focus of this work has been on this last component of the emphasis realization (*i.e.*, using suitably labeled units to produce emphasis), we have left out the pauses around the emphatic words since we felt this effect might confound or overwhelm the effect we are trying to study. The implication is also that the stimuli become much harder to evaluate in this case since the listener might benefit, or expect, a salient break index around emphasized words [7].

During the test, listeners were given the chance to listen to each pair, repeatedly if they wished, and were told that each sentence in the pair *may* contain one or more words bearing emphasis. Their task was to select which sentence of the pair they thought best conveyed emphasis. The overall results from all 31 subjects are summarized, according to condition, in Table 5.

Condition	Neutral	Emphatic
1	229 (61.6%)	143 (38.4%)
2	181 (48.7%)	191 (51.3%)

Table 5: Results of listening test for conditions 1 and 2. Each cell contains the number of times (and percentage) that a particular type of sentence (neutral or emphatic) was preferred within each condition

This is a difficult listening identification task for some of the reasons we have already highlighted. Additionally, the listener's attention is not directed toward specific words in the pair so that he can contrast those words. Moreover, the word(s) that may be candidates for perceived emphasis in the first sentence may not be the same word(s) that the listener is considering as candidates in the second sentence (in which case he may have to resolve based on, *e.g.*, the relative degree of emphasis). In spite of this, it is surprising that, when evaluating Condition 1, subjects indicated 61.6% of the time that the neutral baseline sentence was the emphasis-bearing stimulus and only chose the emphatic sentence 38.4% of the time. Our hypothesis for why this is so is the following: in the emphatic samples of Condition 1 (sentences of Type B), we are highly biasing the search toward choosing emphasis units from a smaller inventory of units (the small studio recording), and trying to aggressively recruit units from this limited inventory for synthesis creates artifacts (*e.g.* clicks or warbles) that might interfere with the perception of emphasis.

However, when we compare across conditions, we see that there is a large improvement from 38.4% to 51.3% (statistically significant at the $p < 0.001$ level) [1] in the identification of intended emphasis when automatically labeled units are allowed to play an explicit role in the synthesis of emphatic words (sentences of Type C). Since our ultimate goal is to improve how accurately emphasis is conveyed, in practice we would adopt the hybrid approach described, where we use a combination of break-index and unit selection. This would allow us to bias less aggressively toward choosing the "right" units while exploiting the perceptual salience of carefully placed pauses. However, the experiment we have carried out here demonstrates the advantage of exploring the synthesis corpus, by means of automatic expression-recognition algorithms, to extract examples of expressive units that can be found scattered throughout a large database, and which can be harnessed at synthesis time to increase the expressiveness of TTS.

5. Conclusions

In this paper we have tried to make a case for applying machine learning and datamining techniques to concatenative-unit speech synthesis corpora in order to enhance the expressiveness of TTS. Although recording a large database for every desired expressive style can be very effective, it can also be costly in terms of recording time, voice talent fees, and system footprint size. The premise of this work lies in recognizing that there is often noticeable expressive variability to be found within large databases which can be exploited as an alternative to enlarging existing expression-specific corpora. The approach is of course limited by the extent to which the expression can be at all found in a baseline, mostly neutral, database: some speakers may exhibit less expressive variability, especially when they may have been coached to speak with a consistent style for the purpose of speech synthesis. Or the limitation may be one of degree: a given expression may be found, but in a much mitigated form.

To apply these ideas we focused on the identification and realization of perceptual emphasis, something which we expected to find with some likelihood in a large database of approximately 10 hours. We described a system that can be automatically trained to identify with reasonably high performance the occurrence of emphasized words throughout the database, and then demonstrated that augmenting the corpus with these automatically discovered labels significantly enhances the perception of intended emphasis.

[1] Statistical significance is assessed in a Bayesian fasion by treating the "identification rate" as a random variable $x \in [0, 1]$ with a parametric distribution $p(x|k, n) \propto x^k(1 - x)^{n-k}$, where k is the number of times a stimulus is identified in a population of size n. Significance is then evaluated as the p value which satisfies the inequality $p(x_1 < x_2|k_1, n_1, k_2, n_2)$ for the two events examined. See Appendix D in [8] for mathematical details.

6. Acknowledgements

The authors wish to thank Andy Aaron for useful discussions and feedback leading to the design of the perceptual tests, Andy and Larry Sansone for their help in running these tests, and Michael Picheny for useful discussions and motivation for this research. The authors also wish acknowledge and thank the support of the TC-STAR project (Technology and Corpora for Speech to Speech Translation; <http://www.tc-star.org/ssw6/>.) [2], a long-term effort financed by the European Commission within the Sixth Framework Program to advance research in speech-to-speech translation technologies.

7. References

[1] J. F. Pitrelli, R. Bakis, E. M. Eide, R. Fernandez, W. Hamza, and M. A. Picheny, "The IBM expressive Text-to-Speech synthesis system for American English," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1099–1108, July 2006.

[2] E. Eide and R. Fernandez, "Database mining for flexible concatenative Text-to-Speech," in *Proc. ICASSP*, vol. 4, Honolulu, Hawai'i, April 2007, pp. 697–700.

[3] M. Bulut, W. Narayanan, and A. Syrdal, "Expressive speech synthesis using a concatenative synthesizer," in *Proc. ICSLP*, Denver, CO, U.S.A., 2002.

[4] N. Campbell and T. Marumoto, "Automatic labelling of voice-quality in speech databases for synthesis," in *Proc. ICSLP*, vol. 4, Beijing, China, October 2000, pp. 468–471.

[5] E. O. Selkirk, *Phonology and Syntax: The Relation Between Sound and Structure*. Cambridge: MIT Press, 1984.

[6] F. Tamburini, "Automatic prosodic prominence detection in speech using acoustic features: An unsupervised system," in *Proc. Eurospeech*, vol. 1, Geneva, Switzerland, September 2003, pp. 129–132.

[7] J. Pitrelli, "Expressive Speech Synthesis using American English ToBI: Questions and contrastive emphasis," in *Proc. IEEE ASRU: Automatic Speech Recognition and Understanding Workshop*, St. Thomas, U.S. Virgin Islands, December 1-4 2003.

[8] R. Fernandez, "A computational model for the automatic recognition of affect in speech," Ph.D. dissertation, (http://affect.media.mit.edu/pdfs/04.fernandez-phd.pdf) Media Arts and Sciences. Massachussets Institute of Technology, 2004.

[2] Project No. FP6-506738