Machine Learning for Text Selection with Expressive Unit-Selection Voices

Dominic Espinosa, Michael White, Eric Fosler-Lussier, and Chris Brew

Department of Linguistics, The Ohio State University, USA

{espinosa,mwhite,cbrew,fosler}@ling.ohio-state.edu

Abstract

We show that a ranking model produced by machine learning outperforms two baselines when applied to the task of selecting texts for use in creating a unit-selection synthesis voice with good domain coverage. The model learns to predict the estimated utility of an utterance based on features relating it to the utterances selected so far and a corpus of target utterances. Our analyses indicate that our discriminative approach continues to work well even though the presence of rich prosodic and non-prosodic features significantly expands the search space beyond what has previously been handled by greedy methods.

Index Terms: speech synthesis, unit selection, machine learning

1. Introduction and Background

Unit selection is a concatenative technique for speech synthesis which employs a database of recorded speech, labelled at the level of segments. Novel utterances are then synthesized by choosing the best sequence of segments available in the database and concatenating them with minimal or no digital signal processing (DSP). The "best" sequence of segments is the sequence that minimizes certain cost parameters evaluated at each step in building the sequence. The voice-building system used for this work employs two general parameters, known as *target cost* and *join cost*. The target cost represents the degree to which a segment matches the linguistic context, and the join cost represents the acoustic fit of the segment. The best sequence of segments is chosen by Viterbi search over the contents of the database [1]. If the match is perfect (for example, when the voice is asked to produce an utterance the database already contains), the costs are zero.

The quality of the utterances that can be synthesized with a unit-selection voice improves with the size of the recorded database, as more segments are available from which to choose. We hypothesize that the improvement can be usefully correlated with target and join costs of synthesized utterances. Impressionistically, a high total join cost suggests there will exist acoustic artifacts in the output, while a high target cost tends to mean the utterance will sound unnatural. For example, Table 1 displays the relationship between the number of utterances in the database, target costs, and join costs for a synthesized version of the following ToBI-annotated utterance (taken from the *directions* corpus described in Section 2):

(1)

Hit	that	button	in	front	of	you
H*	!H*	!H*				L-L%

With only 100 utterances in the database, the synthesized sentence contains unpleasant glitching sounds due to poor joins, and this is reflected by its high join cost. The prosody of the synthesized utterance is also incorrect, as syllables or words with

DB size	Target cost	Join cost
100	2.78	1.71
500	1.37	1.66
1000	1.43	1.05

Table 1: Relationship of DB size to costs for Example 1

L* accents must be substituted for some of the desired H* accents which do not exist in the database. Such substitutions are often perceived as highly unnatural, and in a dialogue situation, can markedly affect a human participant's sense of the synthetic voice's fluency, or even cause an utterance to become infelicitous.

Generally, a compromise must be sought between the size of the database of recorded speech, and the coverage desired for the intended domain. But how much is enough? Even with large amounts of recorded data, coverage suffers due to the well-known phenomenon that "rare events" in language occur in large numbers. Collecting all combinations of phonetic events (phone, diphone, pitch-accent, boundary-tone, etc.) would require a recorded database of gargantuan size, and the effort required to label it would be unrealistic [2]. Given that one would like to achieve the best performance from some set of utterances that can feasibly be recorded, the following problem arises: given a domain of utterances that a voice is to cover, and an existing database (which may be empty), what set of utterances should be recorded and added to this database next?

Previous approaches to this problem [3, 4, 5] have generally involved analyzing the domain, establishing a list of units to be covered and then iteratively selecting the sentence from the domain that contains the largest number of units remaining on the list (though acoustic information is sometimes used, e.g. in [6]). For example, the list might contain all triphones known to occur in the domain, and their frequencies. The selection process is then run until a) enough units are covered and b) enough examples of each unit have been collected, according to the frequency with which they occur in the domain [4]. A similar approach was used with the Festival speech synthesis system in [5], where selection was based on weighted frequencies of uniphones, diphones, triphones, quinphones, and context-dependent diphones. Even with the relatively small number of unit types described in both of these procedures, the search space was extremely large and pruning was necessary. With the addition of explicit prosodic information, which is needed in building expressive voices, the search space becomes even larger.

The approach developed here is to use machine-learning techniques to train a model that can predict a ranking over sets of utterances, indicating which ones will improve the coverage of the voice the most after being added to the database. To this

end, we develop a means of quantifying this improvement, *estimated utility*, which is described below.

2. Methodology

1,016 recorded utterances from a constructed domain of direction-giving speech (the *directions* corpus, based on the SCARE corpus of shared-task video and speech [7]) were used to create unit-selection voices as described below. The ToBI-annotated text for 8,135 unrecorded utterances from this same domain served as a set of target utterances over which to test synthetic voices.

In learning a ranking over some set of candidate utterances, the size of the existing voice's database is an important factor in calculating the usefulness of an utterance which may be added. Thus, the recorded utterances from *directions* were grouped into four overlapping sets for use in unit-selection voices, containing 10, 100, 400, and 700 utterances respectively, corresponding to "tiny", "small", "medium", and "large" categories.[1] These sets were chosen randomly from the 1,016 available, and their small sizes reflect the desire to minimize the amount of recording necessary to build an acceptable customized voice. A set of 100 utterances from the remainder was chosen (randomly) as a candidate set over which to learn a ranking.

2.1. Voice construction

2,249 utterances were chosen from lists of possible sentences from three corpora, and recorded by a ToBI-trained speaker. Noise-reduction was applied. The utterances were automatically labeled at the segment level through the use of HTK [8]. The ToBI-annotated utterances were converted to APML,[2] and voices were then constructed with the Festival speech synthesis system. 1,016 of the recorded utterances came from the *directions* corpus, 800 from the *news* corpus [5], and 433 from the *comic* corpus [11]. Only the *directions* utterances are directly used by the voice and experiments described here; the other two sets of utterances were used as additional data for the forced-alignment step.

The selection process for the utterances that were recorded for these corpora was not arbitrary. Some care was taken at that time to produce complete coverage of the domain. In particular, for the *directions* data, a selection process was implemented that chose 260 utterances to cover all word + pitch-accent + edge-tone combinations possible, 151 utterances to cover all possible pairs of adjacent words, and 408 utterances to cover yet-unseen combinations of pairs of words with pitch-accents and edge tones. 200 further utterances were chosen randomly from the domain, yielding 1,018 utterances.[3] Thus, a database selected randomly from among these utterances can already be expected to perform fairly well.

2.2. Estimated utility

In order to learn to rank one utterance as being better than another, a measurement of its usefulness is needed. As discussed in Section 1, we conjecture that Festival's measurements of the target costs and join costs can be used as an approximation for the goodness of a particular utterance.

The *estimated utility* of an utterance to a given database can thus be calculated by synthesizing the set of target utterances with the database, regarding the sum of the costs as the base cost, and then adding the utterance to the database and synthesizing the targets again. The difference between the two cost sums is the utility of adding that utterance to the database. This measure can be used to construct a ranking over a set of "candidate" utterances – a candidate with a higher utility will have a greater (positive) effect on the overall costs after it is added to the database. The process can be described algorithmically as follows:

foreach *database* D_i **do**
  Compute the base cost B by using D_i to synthesize the target set T, then summing the target costs (tc) and join costs (jc) across all target utterances;
  foreach *candidate utterance* C_j **do**
    Add C_j to D_i;
    Use D_i to synthesize T;
    Extract features for C_j with respect to D_i;
    Compute $Z_{c_j} = B - \sum^k \text{tc}(T_k) + \text{jc}(T_k)$;
  end
end

The result value Z_{c_j} represents the estimated utility of adding the utterance C_j to the database D_i. A set of features for C_j is also extracted.

2.3. Features and machine learning

For our machine-learning experiments, several sets of features were computed for each candidate, with respect to a given database, and a given set of target utterances. Three databases were used, containing respectively 100 utterances, 400 utterances, and 700 utterances. The set of target utterances was always the same (8,135 utterances from the *directions* corpus that were not recorded).

Five sets of features were used: non-contextual, contextual, lexicalized, corpus-based, and interaction.

The non-contextual features of an utterance are computed solely from the contents of the utterance itself. Features included the number of diphones, number of words, number of `H*` accents, number of `L*` accents, number of `L+H*` accents, and type of boundary tone, if any. In example (1) in Section 1, the feature `num_hstars` has the value 3.

The contextual features of an utterance depend on the contents of the given database and the contents of the utterance. The idea is to represent what contributions the candidate utterance makes to the database. For example, if the utterance contains three rare diphone/pitch-accent combinations and ten common diphone/pitch-accent combinations, then two such features should be constructed. To represent the ideas of rareness or common-ness, the counts of such combinations occurring in the database were put into numbered bins by taking the floor of base-two log of the count, with one added.[4] A diphone/pitch-accent combination that only occurs in the database once or twice would be in bin 1, whereas one that occurs in the database 15-30 times would be in bin 4. Similar features were generated for words and syllables in the utterance, and also syllable/pitch-accent and syllable/boundary-tone combinations.

[1] We experienced technical difficulties when using an empty database.

[2] Affective Performative Markup Language, see [9] and [10].

[3] Two utterances were removed from this pool during the voice-construction process due to incompatibility with Festival.

[4] I.e., $b = \lfloor \log_2(c+1) \rfloor$, where c is the count and b is the bin number.

The corpus-based features are similar to the contextual features, but instead of depending on rarity with respect to the voice database, they depend on rarity with respect to the set of target utterances. The interaction features are again similar, but depend on the rarity or frequency of a combination in both the database and the target set. For example, if an utterance contains three diphone/pitch-accent combinations that have been seen only twice in the database (bin 1) but 14 times in the target set (bin 3), then the value of the feature `d/p-a.1.3` is 3.

Lexicalized features depend on the specific lexical contents of the utterance. In the case of example (1) in Section 1, the utterance contains a single instance of the diphone `b_ax` (/bə/) as part of the word *button*. Assuming this combination has only been seen once in the database (bin 1), then the feature `b_ax.button.1` has the value 1.

To learn the ranking itself, a support-vector machine (SVM) was trained over the features, using the SVMLIGHT toolkit [12], which supports an extension to the algorithm for training classifiers that allows rankers to be trained instead. In this ranking process, given a feature mapping $\Phi(c)$ over utterances c, training derives a weight vector $\vec{w}$ such that, given an utterance c_i whose utility is higher than that of utterance c_j, the following condition holds as often as possible:

$$\vec{w} \cdot \Phi(c_i) > \vec{w} \cdot \Phi(c_j)$$

Thus, training derives a vector $\vec{w}$ such that the number of pairwise violations are minimized. Feature values were linearly scaled to the range [0,1] to prevent features with large value ranges (for example, the diphone-counting feature) from receiving undue weight. In our experiments, we used a linear kernel with SVMLIGHT's default parameters.

3. Results

To evaluate the accuracy of the models, two sets of tests were performed: 1) n-fold cross-validation; and 2) using the models to perform text selection and build a database, then comparing the synthesis results against two baselines.

3.1. Cross-validation

In n-fold cross-validation, the training data is divided into n subsets. A model is trained on $n-1$ subsets, then tested on the nth subset; since the estimated utility of the utterances used in training is already known, a gold-standard ranking can be induced. This procedure is repeated n times, rotating the held-out subset each time, and the accuracy results are averaged. Accuracy here means the number of pairwise comparisons the model correctly predicts. For example, if utterance u has a higher utility than utterance v, the model should rank u higher than v. Accuracy is therefore computed by examining all pairwise comparisons in the gold-standard data versus the model's predictions, and checking how many are correct out of the total. In the cross-validation tests described below, n = 10.

Four models were cross-validated, one for each database size, over the training data available for that size (i.e., 100 utterance descriptions each). Additionally, to examine the contributions of the various feature sets, each feature set was ablated in turn. Table 2 shows the results. The feature sets are as follows, with feature counts for the largest database size:

FULL All features included (1866)

NOCONT Contextual features removed (1383)

NOLEX Lexicalized features removed (252)

Feature set	Database size			
	T (10)	S (100)	M (400)	L (700)
FULL	89.1	71.6	63.3	58.4
NOCONT	88.7	71.1	65.3	58.4
NOLEX	88.7	68.0	66.4	56.7
NONC	88.7	71.3	64.9	58.7
NOCORP	88.2	70.7	65.1	56.7
NOINT	88.7	71.3	61.2	58.2
SIMPLE	64.7	57.1	48.6	48.8

Table 2: Percentage of pairwise rankings correct for each DB of the given size, selected using a model trained on the given feature set.

NONC Non-contextual features removed (1416)

NOCORP Corpus-based features removed (1153)

NOINT Interaction-based features removed (1269)

For a baseline comparison, we implemented a simple heuristic. SIMPLE in Table 2 reflects a ranking according to a weighted average over the binned diphone counts of each utterance, i.e. $\sum_i 2^{-b_i} \cdot c_i$, where b_i is the bin number and c_i is the count of diphones in that bin. That is, an utterance with many rare diphones (according to the database contents) is given a higher score than an utterance with few rare diphones, and an utterance with a few rare diphones is ranked higher than an utterance with many common diphones. As an implicit baseline, a random choice of utterances would be expected to give a pairwise-comparison accuracy of 50%.

The database size has a large effect on the accuracy of the model, because as coverage of the domain improves, additional utterances have a smaller impact on the synthesis quality of the target set. This means that the ranking model must make distinctions among utterances with increasingly similar estimated utility values.

Examining the feature weights of the resulting models reveals that, at the smallest database size, lexical features corresponding to a few commonly-seen diphone/word pairs and diphone/syllable pairs (e.g. /ɔr/ in *door*) received slightly higher weights than other features. As the database size increases, the prominence of the lexicalized features decreases, and the corpus- and interaction-based features predominate. In particular, features corresponding to the relative rarities of an utterance's syllable/accent pairs receive higher weights than word/accent features, especially at the small and medium sizes.

3.2. Text selection

To evaluate the models' effectiveness in selecting prompts for building a voice database, we compared the performance of the full model to that of 1) the prompt-selection tool `make_nice_prompts`, part of the Festvox toolkit;[5] and 2) randomly selecting utterances to use as prompts. The prompt-selection tool is not compatible with prosodic annotation, so these annotations were stripped; when this system chose an utterance with more than one available tune, one of these tunes was chosen at random.

In the first experiment, the three systems were used to rank 150 recorded utterances from the *directions* corpus that had not been used in training or testing. Figure 1 shows the reduction in overall costs after adding the top n utterances, for

[5] `http://festvox.org`

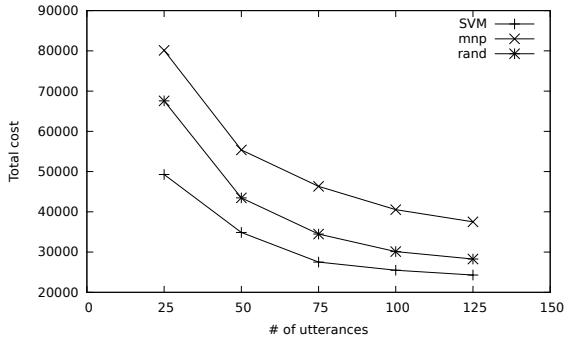

Figure 1: Total cost of synthesizing target utterances as a function of database size, after selecting a database using three systems.

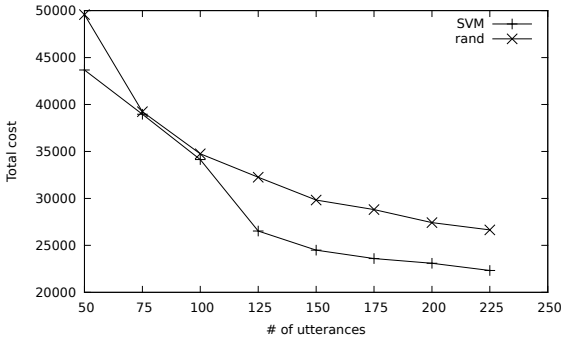

Figure 2: Total cost of synthesizing target utterances for 1) a randomly selected database and 2) a SVM model with iterative bootstrapping.

$n = 25, 50, ..,$ for each system. For the random system, 10 shuffles were performed, and the median of the total costs at each increment was used.

The results show that the SVM achieves a substantially lower total cost for synthesizing the test utterances, with the differences significant at $p < 0.0001$ by paired t-tests. Perhaps surprisingly, `make_nice_prompts` resulted in higher total costs than the random selection. We attribute this to the fact that it attempts to achieve complete coverage with a minimum number of utterances, with no attempt at frequency weighting, and to the pre-optimization issue discussed in Section 2.1.

In the second experiment, we simulated a bootstrapping scenario, in which the utterances recorded so far are used to train a model to select which utterances to record next. This scenario is more realistic than the one underlying the first experiment, in that it does not assume a set of utterances have already been recorded to train a ranker for text selection. A starting database of 25 utterances was chosen at random from the complete pool of recorded utterances (1019). The SVM was trained on these utterances in jack-knife fashion, then used to rank the utterances remaining in the pool. The top 25 were added to the database, then the SVM was retrained on this new database. This process was repeated 7 times. The results are shown in Figure 2, with comparison to adding 25 utterances chosen at random at each iteration. To achieve a random sample, this was run 10 times, taking the mean of the per-utterance costs.

Even with only the 25 initial utterances to train on, the SVM makes a good prediction of what to add next. As the database grows, the ranker's models improve, as we would expect. By the time the database grows to 125 utterances, the ranker substantially outperforms random selection (except at $n = 75$, the differences between the systems are significant at $p < 0.0001$).

4. Conclusion and Future Work

Our results suggest that a ranking can be learned for a set of utterances using the estimated utility measure defined via the utterance costs. Cross-validation results show that the model can learn to rank utterances with accuracy significantly better than random, and two text-selection experiments demonstrate that a SVM model trained to rank utterances by their estimated utility performs significantly better than two baseline systems.

Future work could proceed along three lines. First, perception tests can be conducted to correlate the performance of the ranking models with human judgements. Second, experiments can be undertaken to determine the accuracy of a model trained on one domain in ranking utterances from another domain, and also to determine the efficacy of our approach in extending the domain of an existing voice. Third, our approach could be compared to other algorithms which attempt to achieve good domain coverage through selection on the basis of the frequencies of unit types in the domain, such as those described in [4] and [5].

5. References

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," 1996.

[2] B. Möbius, "Rare events and closed domains: Two delicate concepts in speech synthesis," *International Journal of Speech Technology*, vol. 6, no. 1, pp. 57–71, 2003.

[3] A. Black and K. Lenzo, "Limited domain synthesis," in *Sixth International Conference on Spoken Language Processing*, 2000.

[4] H. François and O. Boeffard, "Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem," in *Seventh European Conference on Speech Communication and Technology*, 2001.

[5] Y. Saikachi, "Building a unit selection voice for festival," Master's thesis, University of Edinburgh, 2003.

[6] A. Black and K. Lenzo, "Optimal data selection for unit selection synthesis," in *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[7] L. Stoia, D. Shockley, D. Byron, and E. Fosler-Lussier, "SCARE: A situated corpus with annotated referring expressions," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, 2008.

[8] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University*, vol. 1996, 1995.

[9] B. de Carolis, C. Pelachaud, I. Poggi, and M. Steedman, "APML, a mark-up language for believable behavior generation," *Life-like Characters. Tools, Affective Functions and Applications*, pp. 65–85, 2004.

[10] M. Steedman, "Using APML to Specify Intonation," 2004. [Online]. Available: http://www.ltg.ed.ac.uk/magicster/deliverables/annex2.5/apml-howto.pdf

[11] M. Foster and M. White, "Assessing the impact of adaptive generation in the COMIC multimodal dialogue system," in *Proceedings of the IJCAI 2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 2005.

[12] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2002, pp. 133–142.