Domain-Optimized Unit Selection Speech Synthesis for Reading Speech Synthesis Papers

Henry Heyden

***Abstract*—Unit selection speech synthesis is an ideal medium through which to study the intricacies of the speech synthesis problem. To exemplify this, a unit selection synthesizer was designed with the goal of being better at synthesizing utterances from papers about speech synthesis than a baseline, with the dataset source domain as the independent variable. In building the voice, the amount of training data used in the alignment step, the importance of quality F0 estimation, and a reduction of target cost weight are motivated through experiments. In a pairwise forced-choice experiment, listeners chose the domain-optimized voice over other synthesized voices 8 times out of 10, despite comparable MOS scores.**

***Index Terms*—speech synthesis, unit selection, script selection**

I. Introduction

SPEECH synthesis is the task of taking some input prompt, usually in the form of text, and synthesizing audio that sounds like a human speaking the desired utterance. One method that has been a mainstay in the field for many years is unit selection synthesis [1] which will be explained in the following Section I-A. Other methods of speech synthesis have been developed, as will be further explored in Section I-B.

In the following, experiments are presented with the goal of constructing a speech synthesizer specialized to the task of reading sentences from informational materials on the topic of speech synthesis.

A. Unit Selection

As the name suggests, unit selection is comprised of two parts: units are sections of recordings of natural speech, and selection is the process by which these units are chosen from a database. Once the units have been selected from the database, they are strung together to synthesize the desired utterance.

The units can be any type of phonetic unit, but one common choices is the diphone, which is a segment that spans approximately from the middle of one phone to the middle of the following phone [1]. Regardless of the unit type, phone-level alignment is a vital aspect of the process, as the alignments mark the boundaries of each phone, allowing the system to find the unit in the database of recordings at synthesis time.

Two cost functions, the target cost and the join cost, govern the selection process. The join cost is based on a prediction of how audible the join will be when playing two units in succession. The Festival unit selection framework, which will be explored in great depth in section II, calculates the join cost between two units by comparing their fundamental frequency (F0), energy, and spectral envelope [2]. The target cost is based on the difference between the unit being selected and the desired unit; e.g. if the desired unit were a [p], an instance of the phone [s] would have a high target cost, whereas a [p] would have a low target cost [3].

B. State of the Art Methods

Throughout the present discussion, references will be made to state-of-the-art methods of speech synthesis. Although these methods are very popular in the current landscape, the current discussion utilizes unit selection methods for two reasons. First, the task of recording enough data to train one of the newer models takes a very long time (upwards of 60,000 hours of training data in [4]), whereas a unit selection model can be trained on hours of speech (as in [2]). Second, neural network based systems are less interpretable, which makes it difficult to attribute a change in performance to a single change in procedure. In contrast, unit selection systems are much more transparent, which allows the discussion to emphasize the data and the evaluation, two aspects of the speech synthesis problem that are more constant than neural-network architecture and interpretation.

II. Unit Selection Voice Design

A number of important design decisions are made when constructing a synthesized voice. In this section, experiments will be presented which were performed in order to motivate choices made when building the voices used in subsequent studies. The relative importance of these decisions in the context of the modern methods mentioned in Section I-B will be explored as well.

All of the following experiments were done using the Festival framework for unit selection synthesis [2]. The speech data used were recordings of the author reading the first 400 utterances from the Arctic A script [5]. The recordings were made in a soundproof booth in the basement of Appleton Tower at the University of Edinburgh.

A. Alignment

1) Motivation and Hypothesis: As was discussed in Section I-A, forced alignment is a crucial part of the unit selection speech synthesis pipeline. If the system has incorrect information surrounding where the units being selected are located within the database, incorrect sections of recordings will be selected. As such, I hypothesize that a voice built with grossly incorrect alignment will produce unintelligible utterances.

2) Methodology: I built a synthetic voice wherein the alignment was trained only on 13[1] utterances, rather than all 400 available utterances. The voice was tasked with synthesizing 20 test utterances, at which point an expert listener (the author) was tasked with determining if the synthesized utterances were intelligible.

3) Results: After listening to the 20 synthesized utterances, I determined that they were all unintelligible.

4) Analysis: As hypothesized, accurate forced alignment is crucial in the construction of a unit selection voice. In state-of-the-art methods of speech synthesis, alignment remains crucial as it is a significant aspect of the training process [4], [6]. However, unlike unit selection, some of these methods are fairly robust to inadequate alignment (as in [6]), and some do not use phoneme-level alignment at all, using only normalized text and audio to train the model (as in [7]).

B. F0 Estimation

1) Motivation and Hypothesis: As was discussed in Section I-A, the difference in F0, the acoustic correlate of pitch, is one aspect of the join cost. I will examine the effect of inconsistent F0 estimation on the unit selection system. A system with more consistent F0 estimation will theoretically have the ability to join units at points where they have similar fundamental frequencies. In contrast, I hypothesize that inconsistent F0 estimation will lead to a significant increase in audible joins.

2) Methodology: The most common errors that occur during the process of F0 estimation occur when a pitch either double or half the correct frequency is estimated to be the true F0. This occurs when the algorithm mistakenly claims that either two fundamental periods or half of a fundamental period is the fundamental period of the signal[2]. To construct a voice with inconsistent F0 estimation, I modified parameters within the algorithm's code that correspond to the upper and lower bounds of the fundamental frequency of the speech signal. I invited a higher likelihood of pitch doubling when estimating F0 using these parameters. Both original bounds and the experimentally modified bounds are given in Table I.

TABLE I
RAISED AND WIDENED FREQUENCY RANGE FOR SUBOPTIMAL F0 ESTIMATION

	Baseline	Experimental
Mimimum Frequency	70 Hz	115 Hz
Maximum Frequency	170 Hz	400 Hz

In pilot experiments, performing F0 estimation with these experimentally expanded ranges did not create a sufficiently inconsistent result. Instead, I used the baseline range to estimate F0 for half of the files, and the experimental range for the other half. A voice was built that was identical to the Baseline voice, save for these inconsistent F0 estimations, which will be referred to as the Experimental voice.

After script selection, it is very likely that the voices have all necessary units in their database already, because earlier-selected utterances are more likely to already have the units that are present in utterances selected later [5]. As such, the Baseline and Experimental voices were tasked with synthesizing 11 utterances, which were late Arctic A and early Arctic B utterances. Since proper names are often transcribed incorrectly by the text-to-phone system, utterances containing them were skipped. Both of these considerations were made because joins caused by units missing in the dictionary would be a confounding factor in this experiment.

An expert listener (the author) was then tasked with listening to the synthesized utterances and counting the number of audible joins were in each. I listened to each utterance at most 5 times, and utilized Praat for closer analysis to confirm my perception [9]. After recording these results, a two-tailed, paired t-test was performed to determine the statistical significance of any observations, with a value of $p < 0.05$ suggesting significance.

3) Results: Table II provides the number of audible joins for each voice for each utterance, as well as the source Arctic script and index for each utterance.

TABLE II
NUMBER OF AUDIBLE JOINS PER VOICE.

Script	Index	Baseline	Experimental
A	500	2	3
A	501	5	6
A	503	2	3
A	504	2	1
A	511	3	3
B	1	1	1
B	2	3	5
B	3	0	1
B	11	2	3
B	13	3	3
B	17	2	3

The average number of audible joins per utterance was 2.27 for the Baseline voice, and 2.90 for the Experimental voice. The p value of the two-tailed, paired t-test was 0.026.

4) Analysis: It can be seen in Table II that the experimental voice produced speech with significantly more audible joins ($p = 0.026 < 0.05$). As such, it can be seen that consistent F0 estimation is a crucial aspect of unit selection synthesis, as audible joins reduce the quality of a voice.

In the state-of-the-art, methods that explicitly model F0 are not very common, although they have shown promising results [6]. Much more common are models that do not explicitly model any parameter of speech, opting instead to implicitly model speech through high-dimensional representations learned by neural networks [4], [7].

C. Target Cost Weight and the Short Pause Problem

1) Motivation and Hypothesis: In the above studies, in-database utterances were occasionally used as test utterances, to examine the behavior of the system when synthesizing utterances that exist in full in the database. Despite the fact that

[1]In pilot experimentation, decreasing the number of utterances in a logical way, i.e. to 200 or 100, did not meaningfully effect the quality of the alignment. This very low number of utterances was used because the goal of this experiment was primarily to evaluate the behavior of the system when alignment is performed suboptimally.

[2]A similar algorithm to the one used by Festival is shown in [8].

the system has access to natural recordings of these utterances, audible joins were often heard in the synthesized utterances. This is due one key design decision: the prediction of short pauses.

When the system converts text into a phone sequence, it puts markers between words that represent short pauses. This allows the aligner to mark any inter-word pauses. However, since the text to phone process adds these markers between every word, two adjacent phones separated by a word boundary may not be marked as adjacent in the system.

In the context of unit selection, the phenomenon that in-database utterances are not selected in full at synthesis time is caused by the short pause model, because while synthesizing these utterances the system does not place a short pause between every word, meaning that the string of units that exists in the database representation of the utterance can be different than the string of units the system wants to concatenate together at synthesis time

In later experiments, during which the system will have domain specific technical vocabulary in its database, it is important that full words or phrases are taken in full from the database. In this section, the target cost weight will be reduced in an attempt to ensure it does so.

As was discussed in Section I-A, the target cost is how unit selection synthesis takes the context of the units in the data base into account at synthesis time [3]. Since I would like the system to be allowed, when necessary, to select units other than the exact unit predicted by the text to phone process, I propose reducing the weight of the target cost. This way, the join cost will carry more weight at synthesis time; and, since units that are adjacent in the database are given a join cost of 0, the system will become much more likely to select adjacent units.

2) Methodology: I first synthesized 10 test utterances using the baseline voice using default settings. Five utterances came from the voice's database itself, and five were unseen. After generating those utterances and listening to them each three times, I decreased the target cost weight and listened to the same utterances three times. I then recorded my observations, comparing the utterances to those which came before, repeating this process until the target cost weight was 0.008. Special care was taken to listen for audible joins, and in many cases I examined the exact units which were being concatenated to synthesize the utterance, in order to confirm my observations.

3) Results: Table III provides the observations made during this experiment in regards to the in-database test utterances. All notes are made relative to the previous row in the table, i.e. "fewer audible joins" means that there are fewer audible joins than the row above. Where ranges are given for target cost weight, no audible difference between utterances was heard within the range of weights.

No difference was heard in unseen utterances while decreasing the target cost weight from 1.0 to 0.008.

4) Analysis: Since reducing the target cost weight had no effect on the out of database utterances, I will select a target cost weight for the system which optimizes for the synthesis of in-domain words and phrases. It can be seen in III that the target cost weight of 0.008 has the best quality of synthesis for in-domain utterances, which is less than hypothesized. I argue that this quality improvement will extend to situations where the system is tasked with synthesizing utterances which contain domain-specific, in-database technical language, because utilizing this decreased target cost weight will increase the likelihood that those words and phrases are selected in full from the dictionary, which will minimize joins within them. As such, unlike the results of other experiments in this section, this result will change the design of our final system that I build later in Section IV, where I will use this target cost weight of 0.008.

TABLE III
TARGET COST WEIGHT AND PERCEPTUAL OBSERVATIONS ON IN-DATABASE TEST UTTERANCES

Target Cost Weight	Observations
1.0	All utterances have multiple audible joins.
0.8-0.4	Slight decrease in audible joins.
0.2	Significant decrease in audible joins.
0.08-0.04	Slight decrease in audible joins.
0.02	Significant decrease in audible joins. Significant quality improvement. Occasional missing units.
0.008	Further quality improvement. No missing units.

The question of whether or not to model pauses is important not just in unit selection systems, but in state-of-the-art models as well. Modern systems that do not explicitly model pauses have the possibility of learning where to naturally pause in a way that mimics natural speech, but making this decision invites the risk of the model pausing incorrectly. In this scenario, the designers are left with no way to turn off the pauses, because they are not explicitly modeled by the system.

III. AUTOMATIC SCRIPT SELECTION

For unit selection speech synthesis systems, having access to a phonetically rich database is crucial. During synthesis, if the system does not have a necessary unit in the database, it will perform a workaround such as inserting silence or using a similar unit; however, these workarounds are audible, and thus reduce the quality of the synthesized utterance. Unfortunately, time spent in a recording studio is often expensive or otherwise restricted. This restriction of studio time motivates the script selection problem: given a large corpus of sentences, which sentences should be put in front of the voice talent in the studio? In this section, a corpus will be automatically sorted using multiple algorithms in order to determine the most phonetically rich list of sentences for the unit selection database.

In the experiments that follow, the corpus from which utterances are selected consists of the video transcripts from modules 1-5 of Simon King's Speech Synthesis course[3]. The corpus was converted to a phone sequence by Festival's text to phone system, with hand-created lexicon entries for out of dictionary vocabulary [2].

At every time step, the following algorithms sort the available utterances according to some scoring algorithm and some cost function. The cost function ascribes a value to each possible unit, the scoring algorithm decides which of the

[3]Course available at speech.zone/courses/speech-synthesis

units in the algorithm to count, the value of each utterance is normalized by number of units, and then the utterance with the highest score is added to the script.

Script selection is not as important of a process in state-of-the-art methods of speech synthesis, as these models tend to focus on quantity of data rather than quality. However, the contents of the training data continues to be important: where a unit selection system has an internal pronunciation dictionary, newer methods do not, meaning that the system cannot learn to pronounce words that do not adhere to letter to sound rules. So, while the problem shifts from unit coverage to vocabulary coverage, thoughtful curation of data remains an important aspect of speech synthesis.

A. Scoring Algorithms

1) Motivation and Hypothesis: In this experiment, four scoring algorithms will be compared to each other.

The **Mult** scoring algorithm operates in a very straightforward way. The score of each utterance is calculated by summing the costs of all of the units in the utterance.

The **Set** scoring algorithm operates similarly to Mult, but it only counts every unit type once per utterance. This is meant to punish redundancy in utterances, as the goal of script selection is to select utterances that contain varied unit types.

The **Aware** scoring algorithm is similar to Set, but it only scores units that have not appeared in previously-selected utterances. This is meant to ensure that every unit available in the corpus is selected at least once.

A **Random** sort, which randomly chooses an utterance at every step, was also used as a baseline.

Since Set has been designed to punish redundant units within an utterance, I hypothesize that, while all methods will have a problem where the most common units are incredibly common, and the less-common units are very rare, Set will display this phenomenon least severely. I also hypothesize that Aware will miss the fewest units, because it is the only scoring algorithm designed to prioritize the selection of unseen units.

2) Methodology: To examine the validity of the two hypotheses, I selected 700 utterances[4] from the corpus using each of the scoring algorithm explained above, using diphones as the unit type because Festival uses that unit at synthesis time [2].

Afterwards, three representations of the selected utterances were created. Firstly, a sorted histogram, which allows for the visual comparison of the three scoring algorithms. Secondly, a table which displays the quantities of the seven most common diphones selected by each scoring algorithm, which allows for more precise comparison between distributions. Lastly, a table which displays the number of diphones missed by each scoring algorithm; that is, of the 1396 diphones in the corpus, how many do not appear in the 700 utterances selected by the scoring algorithm.

3) Results: In Figure 1, the sorted histograms of diphone types by scoring algorithm are presented. In Table IV, the quantities of the seven most common diphone types for each scoring algorithm are given. Lastly Table V displays the numbers of diphones each algorithm missed.

[4]700 is comparable to the number of utterances selected for Arctic [5] before hand pruning.

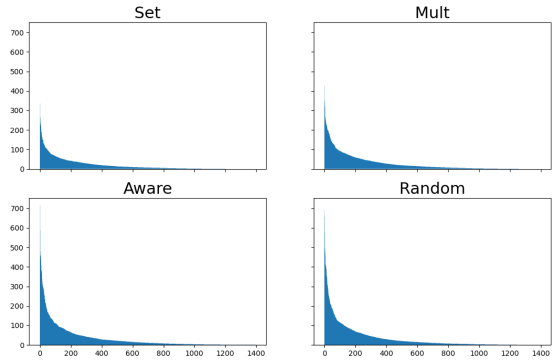

Fig. 1. Diphone Distribution over Scoring Algorithm after 700 Selected Utterances.

TABLE IV
SCRIPT SELECTION ALGORITHM AND THE COUNTS OF THE SEVEN MOST COMMON DIPHONES

Algorithm	First Seven Diphone Counts
Set	336, 268, 261, 240, 233, 224, 224
Mult	426, 353, 343, 321, 315, 266, 264
Aware	716, 588, 478, 460, 459, 455, 438
Random	691, 656, 573, 498, 489, 473, 456

TABLE V
SCRIPT SELECTION ALGORITHM AND NUMBER OF MISSED DIPHONES OUT OF 1396

Algorithm	Missed Diphones
Set	199
Mult	138
Aware	0
Random	152

4) Analysis: It can be seen visually in Figure 1 that Set has the flattest distribution of the four scoring algorithms. This observation is confirmed by Table IV, which shows that Set has the lowest peak (336), and the least distance between the peak and the quantity of the seventh most-common diphone (112). This confirms the first hypothesis. Interestingly, Aware has a higher peak than chance, and has a terribly imbalanced distribution.

Aware was the only scoring algorithm to select at least one of every possible diphone in the corpus, thereby confirming the second hypothesis (Table V). Interestingly, Set missed more diphone types than chance.

In following experiments, the Aware and Set scoring algorithms will be combined, as they both succeeded in their individual goals. The Aware-Set scoring algorithm acts as Aware until every possible unit has been selected, at which point it behaves like Set. This hybrid scoring algorithm will be compared to Set in the following experiment.

B. Cost Functions

1) Motivation and Hypothesis: In the previous experiment, the Proportional cost function was used. At the beginning of the script selection process, this function gives each unit type a score equal to the number of units throughout the entire corpus minus the number of times the given unit type appears, adding 1 to ensure no unit has a cost of 0. This cost function was designed to increase the likelihood that less-common units are more valuable, and are thus selected sooner; however, it is not the only type of cost function.

Another possible yet naive cost function is to simply ascribe each unit a cost of 1. I will test this function and refer to it as Ones[5].

Among the possible combinations of the Set and Aware-Set algorithms and these two cost functions, I hypothesize that Proportional will both miss fewer available units and account for a flatter distribution, and that Aware-Set and Proportional will be the ideal pairing.

2) Methodology: Once again, 700 utterances were selected from the corpus in four different ways; every pairing of scoring algorithm and cost function were used, with the diphone as the unit type.

3) Results: In Figure 2, the distribution of the 222 most common diphones is shown. In Table VI, the quantities of the seven most common diphones are given, in order to allow for precise analysis. Note that Set and Ones missed 232 of 1396 possible diphones, and Set and Proportional missed 199.

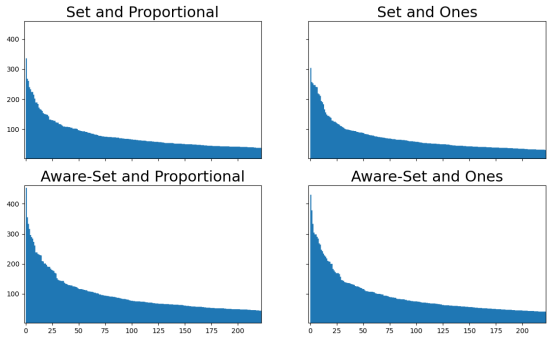

Fig. 2. Diphone Distribution of the 222 most common diphones over scoring algorithm and cost function after 700 selected utterances.

TABLE VI
QUANTITIES OF THE SEVEN MOST COMMON DIPHONES OVER SCORING ALGORITHM AND COST FUNCTION AFTER 700 SELECTED UTTERANCES.

Algorithm	Cost Function	First Seven Diphone Quantities
Set	Proportional	336, 268, 261, 240, 233, 224, 224
Set	Ones	304, 257, 253, 247, 247, 241, 240
Aware-Set	Proportional	453, 355, 333, 316, 296, 289, 284
Aware-Set	Ones	430, 378, 333, 304, 298, 298, 290

[5]Another possible cost function was constructed, wherein the units are sorted by frequency and the most common unit is given a score of 1, and scoring continues linearly until the least common unit is given a score equal to the total number of unit types in the data. This function proved worse than both Proportional and Ones in pilot studies

4) Analysis: Among the possible combinations of the Set and Aware-Set algorithms and these two cost functions, I hypothesize that Proportional will both miss fewer available units and account for a flatter distribution, and that Aware-Set and Proportional will be the ideal pairing.

Set and Ones missed more diphone types than Set and Proportional, thereby confirming the first part of the first hypothesis; however, Set and Ones has a flatter distribution than Set and Proportional, as can be confirmed visually in Figure 2 and quantitatively using Table VI. Set and Ones has the lowest peak (304) and the least distance between the peak and the quantity of the seventh most-common diphone (64).

However, the Festival system sounds very unnatural when it is tasked with synthesizing an out of database diphone. As such, selecting a script without choosing all available diphones has drastic consequences. While it was hypothesized that Aware-Set and Proportional would have a flatter distribution, it can be seen visually that Aware-Set and Ones has the best distribution while still including all diphones. As such, that selection method will be used in further research.

IV. Building the Final Voices

Three unit selection voices were constructed, *Arctic*, Combo, and *Phoneticist*. *Phoneticist* contains recordings of the first 250 utterances selected from the in-domain corpus, which accounts for 9,993 diphones. To keep total number of diphones approximately even over the three voices, the 400 arctic recordings were downsampled using script selection, to create a database of 300 recordings (10,001 diphones). To build the combo voice, utterances were selected from all available recordings, providing 310 recordings (9,998 diphones). The alignments for each voice were trained with all available data, motivated by the results in Section II-A. In all voices the target cost weight was reduced to 0.008, motivated by the results in Section II-C.

V. Subjective Evaluation

In evaluating speech synthesis methods, the gold standard is the subjective listening test. In this section, the results of an online listening test are examined in order to evaluate the quality of the voices built in Section IV.

A. Comparison of Mean Opinion Score

1) Motivation and Hypothesis: Although naturalness is an ill-defined concept, perceived naturalness remains a useful metric in evaluating speech synthesis systems; as such, I ask listening test participants to rank the naturalness of the voices. Since such a crucial aspect of building these voices was ensuring quality synthesis on in-database utterances, I hypothesize that listeners will rank all voices as more natural when synthesizing utterances that exist in their database than otherwise.

2) Methodology: 63 online participants were tasked with listening to four utterances and ranking their naturalness on a scale of 1-5. In-between each utterance, a recording of the same utterance being read by the voice talent was played, to reduce the influence of previous answers. No participant was ever asked to rank the naturalness of the same utterance spoken by multiple voices. Participant responses were used to calculate mean opinion score (MOS). Note that although mean opinion score refers to the mean, the medians of the responses were taken.

3) Results: In Figure 3, the listener responses to each voice are recorded in a box and whisker plot, where the box displays the first and third quartile of the data, and the whiskers display the minimum and maximum responses. In Table VII, the mean opinion score is given for each voice in each context.

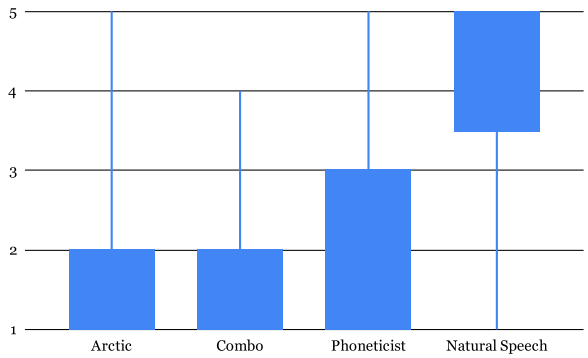

Fig. 3. Overall MOS by Voice.

TABLE VII
MOS BY VOICE AND CONTEXT.

	Arctic	Combo	Phoneticist	Natural Speech
Unseen MOS	2	2	1	N/A
In-Database MOS	2	2	3	N/A
Overall MOS	1	2	2	4

4) Analysis: As can be seen in Table VII, although the MOS score for *Phoneticist* raises two full points when moving from unseen sentences to in-database sentences, the scores are the same for *Arctic* and *Combo*.

B. Suitability of Phoneticist

1) Motivation and Hypothesis: Throughout the present discussion, the *Phoneticist* voice has been built for the task of reading speech technology papers and video transcripts. In this experiment, I will examine whether or not the voice is satisfactory at this task.

I hypothesize the following:

i. *Phoneticist* will outperform both *Arctic* and *Combo* on this task, as it was constructed to do so.
ii. *Combo* will outperform *Arctic* on this task by a wide margin, because it has access to in-domain data.
iii. Actual recorded speech will outperform *Phoneticist* on this task, because it was overwhelmingly preferred by listeners in the previous experiment (see Section V-A4).

2) Methodology: 60 online participants were tasked with listening to four pairs of utterances, selecting which of the two they would rather "read [them] an entire paper".

3) Results: In Table VIII, Voice A is the voice that got more votes when being compared to Voice B. Each pairing has exactly 60 votes between them.

TABLE VIII
PAIRWISE VOTING RESULTS ACROSS ALL QUESTIONS

Voice A Voice B	Phoneticist Arctic	Phoneticist Combo	Combo Arctic	Natural Speech *Phoneticist*
A Votes	49	51	43	58
B Votes	11	9	17	2
% of Total Votes	81.7%	85.0%	71.7%	96.7%

4) Analysis: As can be seen in Table VIII, the *Phoneticist* voice outperformed both *Arctic* and *Combo*, and natural speech vastly outranked the *Phoneticist* voice, as predicted. However, while *Combo* did outperform *Arctic*, *Arctic* earned 28.3% of the votes. Both in this experiment and in Section V-A4, *Combo* has performed worse than expected, presumably because recordings in its database were recorded in two sessions, 28 days apart.

VI. CONCLUSION

Unit selection speech synthesis, with a large database, can theoretically be used to synthesized any utterance. However, it has been shown that this method of synthesis is best suited for situations with a narrow domain vocabulary, using the field of speech technology as an example. Further, techniques such as reducing target cost weight (see Section II-C) and automatic script selection (see Section III) have been shown to improve performance while constructing a voice for domain-specific purposes. Further research could be done to identify if listeners can hear the improvements made by those techniques, as subjective evaluation was not used while exploring those decisions. Another avenue for further research could be examining whether or not state-of-the-art models perform significantly better within narrow domains when trained on domain-specific speech.

REFERENCES

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, pp. 373–376 vol. 1, 1996.
[2] R. A. J. Clark, K. Richmond, and S. King, "Festival 2 – build your own general purpose unit selection speech synthesiser," June 2004.
[3] V. Strom and S. King, "Investigating festival's target cost function using perceptual experiments," in *Interspeech 2008*, pp. 1873–1876, ISCA, Sept. 2008.
[4] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural Codec Language Models are Zero-Shot Text-to-Speech Synthesizers," 2023.
[5] J. Kominek and A. W. Black, "CMU ARCTIC databases for speech synthesis," Technical Report CMU-LTI-03-177, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2003.
[6] A. Łańcucki, "FastPitch: Parallel Text-to-speech with Pitch Prediction," 2020.
[7] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," 2017.

[8] D. Talkin and W. B. Kleijn, “A robust algorithm for pitch tracking (RAPT),” *Speech coding and synthesis*, vol. 495, pp. 497–518, 1995.
[9] P. Boersma and D. Weenink, “Praat, a system for doing phonetics by computer,” *Glot international*, vol. 5, pp. 341–345, 01 2001.