Speech Synthesis Unit Selection Voice Report

B117474

***Abstract*—Unit selection is a conventional yet practical method of speech synthesis (SS). The basic idea is to concatenate pre-recorded units. The selection of units is based on the principle of choosing from candidate units that constitute an utterance which could minimise its overall gap or mismatch with the target utterance, in terms of various phonetic and prosodic features. Despite the simplicity of this idea, the actual realisation has to take into account a lot of factors that could influence the outcome. The factors include but are not limited to, the size and the domain of recording database, the costs which measures how close the candidate units are to the target utterance, etc. In this report, we will see whether and how unit selection is sensitive to various design choices. We hope to be inspired by the results thus able to design unit selection system in a more robust way.**

***Index Terms*—speech synthesis, unit selection, design choices.**

I. Introduction

UNIT selection is a method of synthesising speech by concatenating pre-recorded units selected according to some constraints. These constraints require the linguistic distance between candidate units and target units to be as short as possible, and the acoustic differences between adjacent units to be as small as possible.

Unit selection [1] was proposed more than two decades ago. Since then, Statistical Parametric Speech Synthesis (SPSS) and Deep Neural Network Speech Synthesis (DNN-SS), have been applied and showing satisfactory outcome. Despite of that, it is still meaningful to optimise unit selection. Compared to its successors yet competitors, it can produce more natural speech, given a high quality database and well-tuned parameters (details in Section II). Unlike SPSS and DNN-SS, which generate speech sound from parameters, unit selection can at least guarantee the units are from real voice without parameterisation and re-generation. However, unit selection systems are sensitive to design choices. Tiny changes in any step could lead to big differences eventually. We will cover some design choices of our most interest, and observe their impacts by experiments and evaluation. The unit selection system we use is Festival[1].

II. Unit selection

A. Data preparation

The first thing to prepare is recording scripts. Scripts may vary in content(genre/domain), years, size, unit diversity, etc.

Choosing the first two aspects depends on the type of system. For example, if a system is to be used for synthesising weather forecast, we would prefer scripts of past weather forecast rather than random text scrapped from the Internet, since it is more likely for past weather forecast scripts to include frequently-used weather forecast expressions which the system can use the recording of them as a whole, e.g. *cloudy with rain at times*. The advantage of using domain-specific scripts will be explained again in Section II-F, using the terminology *join cost*. The year of the scripts is vital too. If the system is to synthesis modern language, archaic scripts are not wise choices. They contain old-fashioned vocabulary and pronunciation which are no longer used in modern days. They pose challenges to voice talents, i.e. people who read the scripts. Their reading might be impeded by the archaic words, thus resulting in unnatural prosody or longer recording time. In addition, speech corpora are preferred than written text as they come from natural speech; written text is harder for reading out.

Script size means how many sentences, words, or smaller units, e.g. phones or diphones, are contained. A larger size does not necessarily guarantee better synthesis outcome. There is a trade-off between the size and the recording time. Longer recording time causes voice talents tired. If split into several recording sessions, the consistency of recording is hard to ensure. Simply increasing the size is crude and hasty. It is wiser to promote the diversity of the scripts at the same time. As concatenation happens at diphone[2] level, we define the diversity as the number of diphone types in the script versus the total amount of diphones.

We hope the script could cover as many diphone types as possible, while striking a balance between the diversity and the size (*type:token*). When selecting diphones to join, we need to consider context as well, including phonetic context (the preceding and following sounds), prosodic environment (stress, prosody) and position (in the syllable, word or sentence). A perfect match between a candidate unit and a target unit would be matching the diphone type and the context (more explanation using terminology *target cost* in Section II-F). However, that is almost impossible for lots of diphone types. Meanwhile, a missing diphone type could result in a large number of missing context-dependent diphone types. According to Zipf's Law [2], the natural distribution of diphones has a very long tail, i.e. many of them have extremely low frequency or are never unseen. This is also known as data sparsity or the phenomenon of *Large Number of Rare Events* [3]. Thus, we need a wise test selection algorithm

[1] http://www.festvox.org/docs/manual-2.4.0/festival_toc.html **Festival 2.4.0 Manual** *retrieval date: 04/04/2018*

[2] A diphone expands from the second half of the first phone to the first half of the second phone. For example, *sil_k*, *k_a*, *a_t*, are the diphones of *cat* (*sil* means silence); a more fine-grained version would be: *sil_k*, *k_a*, *a_t_cl*, *t_cl_t* where *t_cl* means the closure part of *t*. As both sides of a diphone have constant acoustic properties, they are ideal units for waveform concatenation.

(TSA) to ensure a larger *type:token* and a flatter diphone distribution (Section III-A).

B. Recording

The consistency of recording is vital. First, the recording devices and settings should be the same for every recording session. Hopefully, the recording could be completed in a single pass or as few times as possible. Even for the latter case, always recording within the same time period in a day could compensate for splitting recording to some extent. Besides, the average length of sentences should be controlled, ideally 5 to 15 words. Sentences too long increase the difficulty of being read out naturally. Voice talents should keep calm, use natural voice, and stick to a specific accent (Section II-C). Breaks should be taken appropriately.

After recording, sanity check should be done on all the recordings and the transcript. Any mismatch between them should be fixed, as preparation for labelling (Section II-D).

C. Dictionary selection

In Festival, three dictionaries are available:

- `unilex-gam`-General American English
- `unilex-rpx`-British English
- `unilex-edi`-Scottish English (Edinburgh)

Dictionaries determine the phone set to use. A phone set contains a set of symbols which are defined with features such as vowel/consonant, frontness of vowels, articulation place of consonants, etc. There are 78 phones in `unilex-gam` and `unilex-edi` while 60 in `unilex-rpx`. Besides, as some vowels are often reduced to schwa in natural utterance, there are rules of phone substitution. There are more vowels reduced to schwa in `unilex-gam` and `unilex-edi` than in `unilex-rpx`.

Dictionaries are used in labelling and alignment (Section II-D), and voice building (Section II-F). Choosing a proper dictionary and using it all the time is important. Otherwise, the inconsistency is problematic. For example, if we use `unilex-gam` to build voice on an RP recording database, it will cause a lot of missing diphone problems, as there are more phones in `unilex-gam` than in `unilex-rpx`.

D. Labelling and Alignment

To align the recordings with the transcript, we convert word sequences into phone sequences, by looking up words in the lexicon (the same dictionary in Section II-C). Then, we align the waveform with the phones, i.e. adding timestamps to the phones. There are two ways of doing that: hand labelling and automatic labelling [4]. Hand labelling relies on either annotators' built-in knowledge or rules. It could be time-consuming, laborious and error-prone. Automatic labelling could solve most of the problems. Borrowing the idea of forced alignment from Automatic Speech Recognition (ASR), we treat labelling and alignment as training the Hidden-Markov-Model-Gaussian-Mixture-Model-Acoustic-Models (HMM-GMM-AMs) in ASR.

Waveforms are represented by Mel-frequency cepstral coefficients (MFCCs) [5]. MFCCs are better than spectral coefficients because modelling correlated spectral coefficients requires more parameters and computation. Uncorrelated cepstral coefficients are suitable for modelling statistical GMMs. Besides the first 13 coefficients we normally use which are representative of the filter, we can add delta (13 dimensions) or delta-delta (13 dimensions) features which model the velocity and acceleration of changes between two adjacent frames. In the experiments onwards, we will be consistent in using 39-dimension MFCCs. MFCC extraction is done by `make_mfccs` from Hidden Markov Model Toolkit (HTK)[3].

We then use MFCCs to train HMM-GMM-phone-level-AMs, allowing optional short pause between phones[4] and phone reduction rules [6]. In `do_alignment` in HTK, variance floor is computed by `HCompV` to prevent the variance from a single speaker being too small [7]. Each HMM has five states, two of which are non-emission. To train AMs, Baum-Welch (`HERest`) and Viterbi (`HVite`) algorithm are applied. Both align feature sequences to HMM state sequences.

Viterbi and Baum-Welch are each carried out for several rounds in `do_alignment`, to ensure correct labelling. During alignment, the number of GMMs are gradually increased for better acoustic modelling. However, too many GMMs might lead to overfitting, worsening the alignment. Besides, too many emission states in HMMs could lead to similar outcome. In ASR, we can keep an eye on the likelihood of training data and the word/phone error rate (WER/PER) to avoid overfitting or underfitting. When the training data likelihood becomes too large or the WER/PER goes higher after some point, we should be cautious about overfitting. In our case, as we are not doing speech recognition, we do not have WER/PER; the training data likelihood could be checked in `aligned.*.mlf`.

Building a single-voice SS system is the same as training a speaker-dependent ASR system, where the training data is single-speaker. The amount of data for our baseline is 593 sentences, including 27658 phones[5]. We conjecture, more data, essentially meaning speaker-independent AMs, would result in better alignment (Section III-C).

After forced alignment, each phone have a starting and an ending timestamp. Mismatch might cause improper choices of units.

E. Signal processing

Each phone in the database needs to be pitchmarked - finding epochs. Pitchmarking is for waveform concatenation, realised by Time-Domain Pitch Synchronous Overlap Add (TD-PSOLA) in Festival. Concatenation at epochs can eliminate unnaturalness and audible artefacts. The peak in each epoch is detected by finding the zero crossings of the derivative of the signal, as the derivative of the highest point must be

[3] http://htk.eng.cam.ac.uk **HTK** *retrieval date: 07/04/2018*

[4] represented by the special phone symbol `sp`

[5] the special phone `sil` indicating silence excluded

zero. Having marked the peaks, we can detect the epochs which are periods in between every two adjacent peaks.

The second step is pitchtracking - F0 estimation. The F0 track across the waveform is approximated by a window spanning through several epochs of the signal. A window sized too small would increase the probability of local errors if pitchmarking, as prerequisite, is accidentally wrongly marked. As speech signal normally can stay stable across several periods, we consider a window sized several periods (at least twice the longest expected period) would be a sweet spot. A common pitch determination algorithm is cross-correlation [8] [9]. It sums up the multiplication of the sample at time t and the sample at time $t+\tau$ (1), where τ is the lag, m the starting sample, n the number of samples in the frame of time length τ.

$$\chi = \Sigma_{t=m}^{m+n-1} s_t s_{t+\tau} \tag{1}$$

The bigger the χ is, the more the signals at time t and $t+\tau$ are overlapping. The pitch period is found as the smallest lag ($\tau > 0$) among all values that we could obtain local maxima of χ. Furthermore, normalised cross-correlation function (NCCF) improves cross-correlation and is put into practice as a Robust Algorithm for Pitch Tracking [9].

Besides, pre-processing and post-processing are also necessary in pitchtracking. In pre-processing, low-passing filtering, waveform downsampling and spectral flattening by inverse filtering [10] remove interfering signal components: noise, unvoiced sounds, formants, etc. [9], and reduce computation. In post-processing, Dynamic Programming (DP) selects between candidate tracks resulted from lags of multiple pitch periods. DP draws the best route among F0 candidates which has the lowest overall cost of moving from one to another.

The predicted range of F0 is an important parameter in signal processing. Normally, male has lower F0. F0 differs within the same gender too. We must ensure correct estimation of the speaker's maximal and minimal F0. A narrow range improves the precision of pitchmarking and saves computation in pitchtracking.

In Festival, running `make_pm_wave` and `make_f0` (with gender-specific F0 range specified, low-passing filtering `-L` and DP `-P`)[6], we obtain epoch and F0. Both are important signal parameters for voice building.

F. Voice building

Two core factors in voice building are target cost and join cost. Target cost determines how close the candidate is to the target. Join cost defines how likely two adjacent units can be perfect combination without audible joins. For a perfect match, the cost should be zero. The computation of the costs are based on mismatch between the features of candidates and targets, or adjacent candidates. The target mismatch is formulated by Independent Feature Formulation (IFF) [1] or Acoustic Space Formulation (ASF) [4]. IFF considers only linguistic features, e.g. phonetic context, stress, position, etc. Each unit in the database is attached linguistic features of concern. However, units with different linguistic features could *sound* similar to listeners[7]. That is why we introduce ASF, which uses acoustic properties to compare how units sound. Acoustic properties include F0, duration and energy, or even more, as long as they are accurately predicted from linguistic features; if not, reduce them. The prediction can be done by regression trees asking questions about linguistic features, i.e. context-dependent clustering, or by neural networks. ASF overcomes some limitations of IFF. It does not make independence assumptions on features, and avoids data sparsity.

In practice, IFF and ASF can be hybrid. The target cost is computed as the weighted sum of differences in features between candidates and targets. The candidates with the lowest target costs are more likely to be kept.

Join cost tells how well two units join together by a weighted sum of sub-costs. The sub-costs are calculated from F0 contours, spectra, energy, etc. While it is hard for target cost to be zero, join cost could be zero when the adjacent units are consecutive units in the database, as the concatenation is natural and perfect. Back to the example of weather forecast database (Section II-A), the reason why we prefer domain-specific database when building a domain-specific voice is exactly because we can expect more zero join costs from frequently-used words, phrases or even clauses in a specific domain. We have an experiment to justify this hypothesis in Section III-B.

To search for the best candidates, we construct a lattice with the target units sequence at the top, below each listed corresponding candidates attached with target costs and bilateral join costs. The search algorithm is Viterbi, essentially DP. By default, all the candidates are displayed. Pruning reduces the number of them by removing candidates with costs larger than the beam width. In Festival, observation pruning is implemented before searching, removing candidates with target costs greater than the beam width. Although it does speed up the search, it suffers the risk of removing the units which later might result in a lower overall cost. Search pruning removes candidates whose target cost and join cost so far are larger than the beam width. Viterbi guarantees a single best path through the candidates with the lowest overall cost. But making the beam size too narrow could cause less exact searching.

Having the units to be concatenated, we concatenate their waveforms together. Waveforms could be saved as parameters, such as Residual-excited Linear Prediction Coefficients (RELPC) in Festival.

III. Experimental design and evaluation

We focus on the design choices of most interest and of a wider range, including the property of recording database, forced alignment, and target/join costs.

[6] http://www.cstr.ed.ac.uk/projects/speech_tools/manual-1.2.0/x2152.htm **Manual of program `pda` used by `make_f0` in Festival** *retrieval date: 08/04/2018*

[7] For instance, suppose the closure part of different unvoiced stops sound similar to each other, hence the target cost between `t_cl` and `p_cl` *should* be zero. But as defined by IFF, they can never be perfectly matched.

Baseline system

In our baseline, we recorded the A set from CMU Arctic corpus[8]. It contains sentences from Jack London's work in the early 20th century. It is phonetically well balanced with a good diphone coverage with a minimal amount of text [11]. Some original statistics about A set are reported in [11]. However, recomputation is necessary. First, the phone set they used is different from ours. Second, the formula of diphone coverage computation was not revealed. Third, when defining diphone coverage, they failed to specify what *text* they refer to in "minimal amount of text". Word-based text and phone-based text make a big difference to the computation. For consistency, we use the phone set of `unilex-rpx`. We define diphone coverage as the number of diphone types covered by the script over the total number of diphone types[9]. We run A set through Festival front-end, and report the recomputation results in Table I.

TABLE I
RECOMPUTED (NUMBERS IN BOLD) STATISTICS OF ARCTIC A SET BASED ON [11])

Number of units				Coverage	
Sentence	Word	Unique word	Phone	Phoneme	Diphone
593	5284	1958	**29316**	**100.00%**	**29.67%**

The recording was done under studio conditions and finished in two times, both within 9 a.m. to 11 a.m. The voice talent recorded with normal and consistent voice, with appropriate breaks. Sanity check was done thoroughly. The recordings were downsampled to 16000 Hz.

A. Text selection: diphone coverage

Hypothesis *Higher diphone coverage could improve the synthesis outcome by reducing the number of unseen diphone types.*

Simply increasing the data size is not always helpful according to Zipf's Law; the diversity of units - diphone coverage, matters more.

Experiment design

To compare with the baseline, we built a new system using the recording of the new script described below.

We designed our TSA and executed it to create a recording script containing 517 sentences from 15000 sentences scrapped from Donald Trump's Twitter[10].

TABLE II
STATISTICS ABOUT THE NEW SCRIPT BY TSA

Number of units				Coverage	
Sentence	Word	Unique word	Phone	Phoneme	Diphone
517	4756	1427	23058	100.00%	31.34%

Our TSA scores all sentences in the database by a *bonus mechanism*: each diphone is attached a score - the inverse of its frequency in Arctic A. The rarer the diphone is, the higher the score would be. The final score for each sentence is scaled by length. Deciding the total number of sentences we want (600) and filtering out sentences longer than 15 words, we created a script as Table II describes.

Results

There are 2531 unseen diphone types in Arctic A, taking up more than 70% of the total number of diphone types.

In our new script, despite of the smaller size, the diphone coverage increases by 1.73%. It proves the effects of our bonus mechanism. We can also imagine that the new diphone distribution is flatter than previous.

A higher diphone coverage leads to fewer missing diphone problems. If we use the new system to synthesise sentences containing rare diphones, we can expect a better outcome. The sentences for synthesis should ideally be random, neither from novel nor politics, to avoid the influence of domain. Although the new diphone coverage is still far from satisfaction, at least we proved that a wiser TSA could achieve more rare diphone types[11]. Also, as not all diphone types listed by permutation and combination exist in English, next time, we will start by reducing impossible diphone types.

B. Text selection: domain

Hypothesis *Domain-specific system could produce speech more natural.*

Experiment design

Using the system in Section III-A, which is also domain-specific, we tested the hypothesis by a web-based listening test[12].

The structure of the listening test and the correspondence between the questions and the hypotheses we are testing are shown in Figure 1. In Introduction, we asked listeners about their environment, and some demographic questions: age (Figure 2), mother language or English ability (Figure 3), experience in speech synthesis (Figure 4), hearing/vision impairment, etc. There are two aspects of evaluation: naturalness and intelligibility. Naturalness means how an utterance is accepted as an utterance spoken by a human being; intelligibility means how well an utterance can be recognised by listeners. To check whether listeners are seriously taking the test, we put two questions including a natural utterance. If the listener fail to give expected responses in Q7 and Q13, we mark them as outliers.

We synthesised three random sentences from Donald Trump's Twitter for Q1 to Q3, and three from Arctic B (the same domain as Arctic A) for Q4 to Q6. In each pair of sentences, one is synthesised by the baseline and the other by the domain-specific system. Listeners choose the more natural one.

Results

The results for Q1 to Q6 are in Figure 7, 8, 9, 10, 11 and 12 (the expected choices that would support our hypothesis

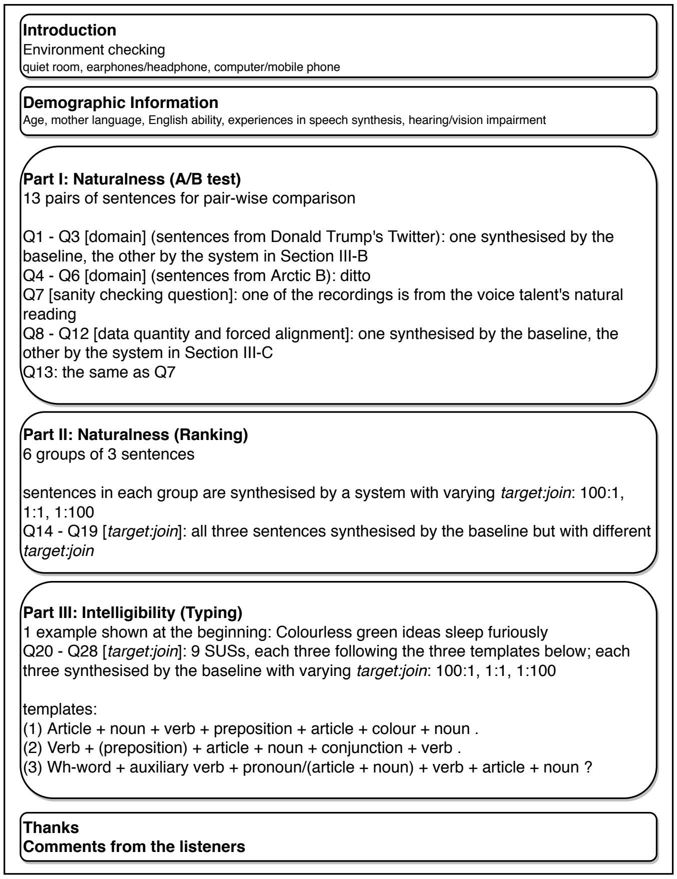
[8] http://www.festvox.org/cmu_arctic/ **CMU Arctic** *retrieval date: 09/04/2018*

[9] $60 \times 60 - 1 = 3599$ in the current case, `sil_sil` excluded.

[10] https://twitter.com/realDonaldTrump **Donald Trump's Twitter** *retrieval date: 09/04/2018*

[11] which are the least rare among all the rare ones

[12] https://edinburgh.eu.qualtrics.com/jfe/form/SV_413EPK7BsZhDbYV **The listening test** *retrieval date: 09/04/2018*. To keep the anonymity of the assignment, we did not include any personal information. However, as the recordings might reveal our identity, please do not listen to them.



Introduction
Environment checking
quiet room, earphones/headphone, computer/mobile phone

Demographic Information
Age, mother language, English ability, experiences in speech synthesis, hearing/vision impairment

Part I: Naturalness (A/B test)
13 pairs of sentences for pair-wise comparison

Q1 - Q3 [domain] (sentences from Donald Trump's Twitter): one synthesised by the baseline, the other by the system in Section III-B
Q4 - Q6 [domain] (sentences from Arctic B): ditto
Q7 [sanity checking question]: one of the recordings is from the voice talent's natural reading
Q8 - Q12 [data quantity and forced alignment]: one synthesised by the baseline, the other by the system in Section III-C
Q13: the same as Q7

Part II: Naturalness (Ranking)
6 groups of 3 sentences

sentences in each group are synthesised by a system with varying *target:join*: 100:1, 1:1, 1:100
Q14 - Q19 [*target:join*]: all three sentences synthesised by the baseline but with different *target:join*

Part III: Intelligibility (Typing)
1 example shown at the beginning: Colourless green ideas sleep furiously
Q20 - Q28 [*target:join*]: 9 SUSs, each three following the three templates below; each three synthesised by the baseline with varying *target:join*: 100:1, 1:1, 1:100

templates:
(1) Article + noun + verb + preposition + article + colour + noun .
(2) Verb + (preposition) + article + noun + conjunction + verb .
(3) Wh-word + auxiliary verb + pronoun/(article + noun) + verb + article + noun ?

Thanks
Comments from the listeners

Fig. 1. Structure of the listening test

are coloured orange). Except for Q5, more listeners prefer the expected system when synthesising sentences from the same domain as the recording data of the system. For Q5, we guess the unexpected result is due to the very unnatural prosody on *one* word in the sentence, which makes it stand out. The could be solved by F0 manipulation or costs tuning. Overall, the results support our hypothesis.

However, the limitation in number and range of the test sentences has to be pointed out. Considering the expected time listeners would spend on the test, we did not include a large number of various sentences. But we can surly expect the same results with more sentences, since with more sentences, more frequent expressions (inherent zero join cost) used by both domains are likely to be selected.

C. Data quantity and forced alignment

Hypothesis *Forced alignment based on speaker-independent training (more data) could produce speech more natural.*

The justification for this hypothesis was stated in Section II-D. To recap, larger data size due to multiple speakers' recordings is helpful to speaker-independent training, leading to better alignment, thus more natural speech.

Experiment design

To train speaker-independent AMs, we collect recordings from three other voice talents whose gender and accent[13] are the same as our voice talent. The total amount of data is fourfold the baseline. The diphone coverage is unchanged as the speakers are using the same script. Alignment is only done on our voice talent's transcript.

In Q8 to Q12, we synthesised pairs of sentences, one by the baseline and the other by the fourfold system. The sentences are randomly selected from phonetically balanced Harvard sentences[14].

Results

The results for Q8 to Q12 are in Figure 13, 14, 15, 16, and 17. Except for Q8, most listeners think the sentences from the fourfold system are more natural. In Q8, listeners almost equally like both systems. In the clip by the fourfold system, the utterance was spoken swiftly with consistent power; in the other clip, one of the word *coat* was especially prolonged and stressed. We guess, despite one word's standing out, listeners prefer the latter because they think a natural utterance features prosody and stressed words. Generally, the results support our hypothesis.

In contrast, the objective evidence does not seem so. We computed the algebraic average of the sum of the training data likelihoods in `aligned.3.mlf` of both systems. We also computed the percentage of achieving larger phone-level likelihood in the fourfold system against the baseline (Table III).

TABLE III
TRAINING DATA LIKELIHOOD OF THE BASELINE AND THE FOURFOLD SYSTEM: LIKELIHOOD MEANS TRAINING DATA LIKELIHOOD; PERCENTAGE MEANS THE PERCENTAGE OF PHONES WITH LARGER LIKELIHOOD

	Baseline	Fourfold
Likelihood	-2444.10	-2517.75
Percentage		50.53%

Although the fourfold system marginally has more phones with larger likelihood, its overall likelihood is much lower than the baseline. The AMs might be underfitting due to insufficient training. The objective evidence contradicts the hypothesis.

For future direction, we consider augmenting the data size to at least tenfold the baseline and add more GMMs, to realise fully-trained speaker-independent forced alignment. This was impossible to carry out in the current experiment due to the small number of voice talents of the same gender and accent as our voice talent. Also, instead of taking the algebraic average of the likelihoods, we should think of more sophisticated and reliable ways of computation, e.g. weighting according to phone frequencies.

D. Target cost versus join cost

Hypothesis *Larger target cost versus join cost target:join makes robotic sound of worse naturalness and intelligibility; smaller target:join improves naturalness and intelligibility together.*

[13] To protect the anonymity of the work, these information are not revealed.

[14] http://www.harvardsentences.com **Harvard sentences generator** *retrieval date: 09/04/2018*

Larger weighting on target costs would pick up candidates that *individually* are very similar to each corresponding target. But meanwhile, low weighting on join costs causes bad joins almost everywhere. So the outcome is just robotic concatenation of diphones. In contrast, larger weighting on join costs could make the speech smooth and less robotic. It could also make liaison sounds more natural. When hearing natural utterance, we do not always have to catch every word and every sound. Given enough context, we can speculate most contents. Even with nonsensical context, people can at least tell the content at word-level. So we hypothesise that naturalness and intelligibility actually supplement each other when the join cost is higher.

Experiment design

To test naturalness, we prepared six groups of three sentences, each synthesised by a system with varying *target:join*: 100:1, 1:1, 1:100 (Q14 to Q19). Listeners rank the sentences within each group regarding naturalness. We on purpose added some phrases of liaison: remin**d you of it at a**ll, wi**th a** ves**t on an a**dorable, etc. The bold parts are spoken in liaison.

To test intelligibility, we put 9 semantically unpredictable sentences (SUSs) [12] consisted of frequent words (Q20 to Q28). SUSs are nonsensical sentences which follow syntactic templates. We used the modified syntactic templates based on [12] (Table 1) to generate 9 sentences, 3 of each template, and within each three of the same template, each is by a system with a different *target:join*: 100:1, 1:1, 1:100.

Results

The average ranks of each system regarding naturalness are demonstrated in Figure 18, 19, 20, 21, 22 and 23. Taking the overall average rank of all systems in all questions, we further obtain the average rank for systems with *target:join* 1:100 - 1.845, *target:join* 1:1 - 2, and *target:join* 100:1 - 2.155. The listeners prefer the systems with more weights on join cost, and rank the ones with more weights on target cost the least natural. This matches with our hypothesis that higher join costs smoothen the joins; higher target costs ruin the overall naturalness.

For intelligibility, we computed word-level F1 scores for all nine SUSs (Figure 5), and average word-level F1 scores for SUSs by systems of different *target:join* (Figure 6). The highest F1 0.85 happens when *target:join* is 1:100, much higher than the other two. This agrees with our hypothesis that naturalness and intelligibility are supplementing each other. Higher target costs impede listeners' understanding of words, as they make every phone stand out and destroy the word-level and sentence-level fluency.

We admit the ordering of sentences is not random enough to prevent listeners from recognising the patterns. Besides, we failed to compare the naturalness on liaison between different systems, as people's choices reflect more about their overall feelings of the sentence. It would be more reasonable to do intelligibility tests on liaison, because in that way we could compute F1 scores on phrases of liaison.

IV. Discussion and conclusion

From our experiments, unit selection is sensitive to design choices such as data domain, target/join cost, etc. Due to limited time and number of listeners, the results might be inconclusive or sensitive to listeners and test design. Some drawbacks we already realised are: bad ordering of test sentences, and small amount of test sentences. In addition, due to the word limit, we failed to make the best use of the demographic information: native/non-native and with/without speech synthesis experience. We expect native speakers and listeners with speech synthesis experience to have smaller divergence of views than the other two types of listeners.

As unit selection runs as a pipeline of several modules, it is hard to guarantee the overall best settings by separate optimisation of each module. Also, it could not work well when the data size is extremely small. Recent decades have witnessed the advantages of SPSS and DNN-SS. SPSS is a model-based system storing sounds as statistical parameters and generating parameters given input [3]. It could work even given a small size of data. SPSS generalises unseen data by clustering context-dependent units using decision trees. However, as the generation of parameters is based on maximum likelihood[15], the synthesised sound sounds muffled due to the effect of averaging. As a statistical method, SPSS is sensitive to the data. If there is a dataset of inconsistent speech, SPSS is better at eliminating the inconsistency. But if there is a large database of consistent speech, SPSS might not work as well as well-engineered unit selection [3].

To overcome the inefficiency and over-generalisation of decision trees in SPSS, DNN-SS is introduced. By directly modelling speech from given text, DNN learns to generate speech parameters. Using DNN to train AMs is advantageous since it does not make any independence assumption on features as HMM does. So correlated features - Filter bank are feasible in DNN. But the computation cost of DNN is higher than that of SPSS [13]. And neural network is sensitive to architecture and parameter tuning.

Unit selection, as a conventional way of speech synthesis, has been practical for many years. Although SPSS and DNN-SS do outperform it in some aspects and are more widely used in industry nowadays, we believe unit selection could be practical and robust if given a large database containing rich variety of units and well-tuned parameters.

[15] the mean always being the most likely

Appendix

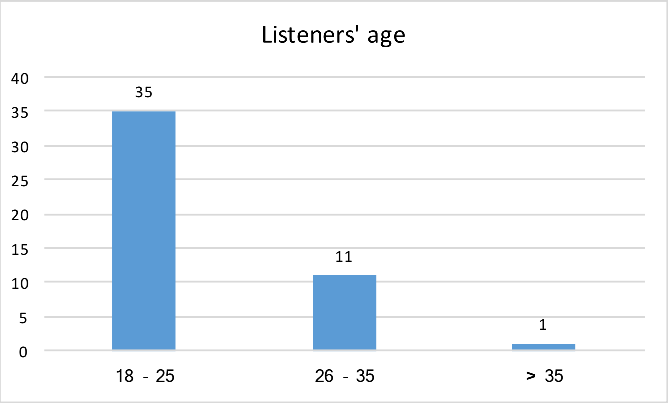


Fig. 2. Listeners' age

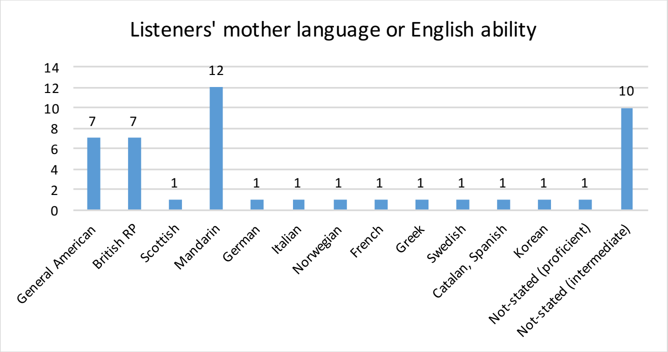


Fig. 3. Listeners' English ability (all non-native if not stated are self-claimed as proficient)

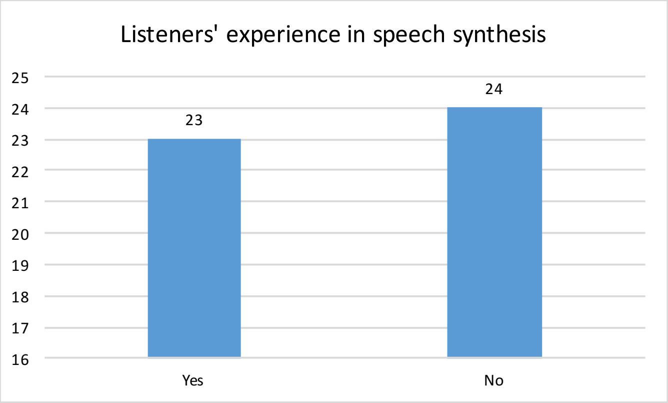


Fig. 4. Listeners' experience in speech synthesis

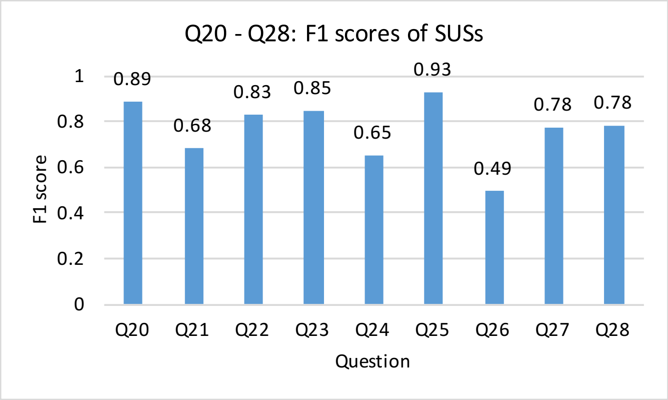


Fig. 5. F1 scores of SUSs from Q20 to Q28

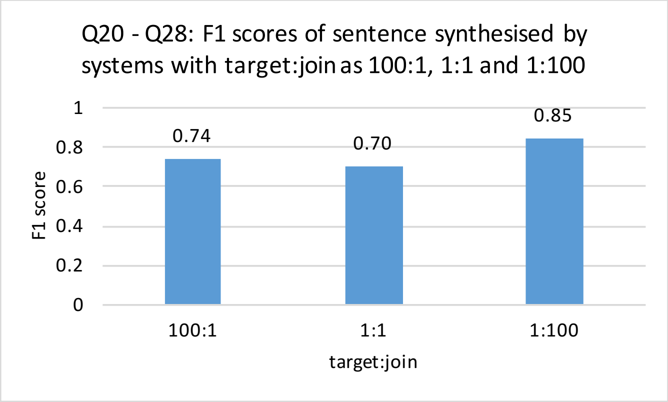


Fig. 6. F1 scores of SUSs synthesised by systems of different *target:join* from Q20 to Q28

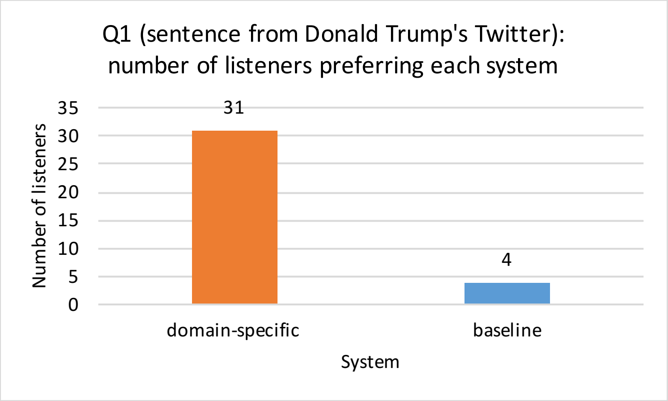


Fig. 7. Number of listeners preferring each system in Q1

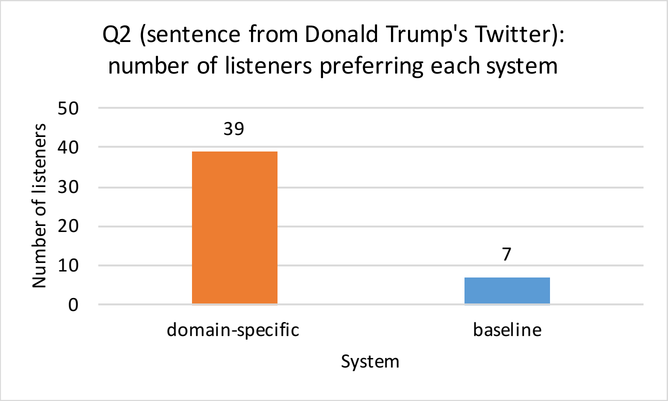


Fig. 8. Number of listeners preferring each system in Q2

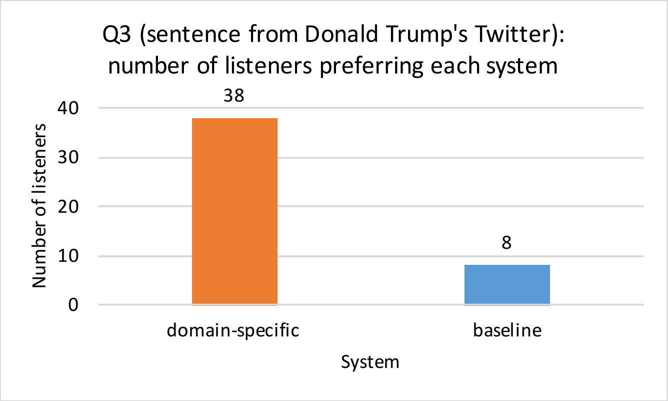


Fig. 9. Number of listeners preferring each system in Q3

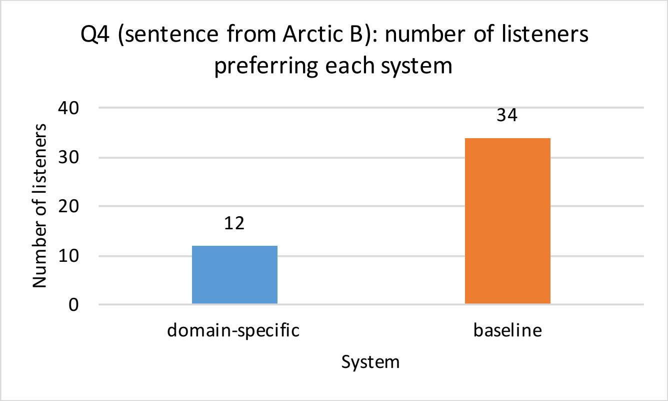


Fig. 10. Number of listeners preferring each system in Q4

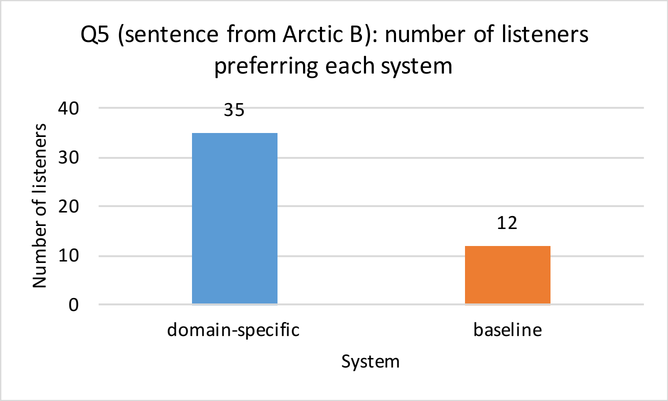


Fig. 11. Number of listeners preferring each system in Q5

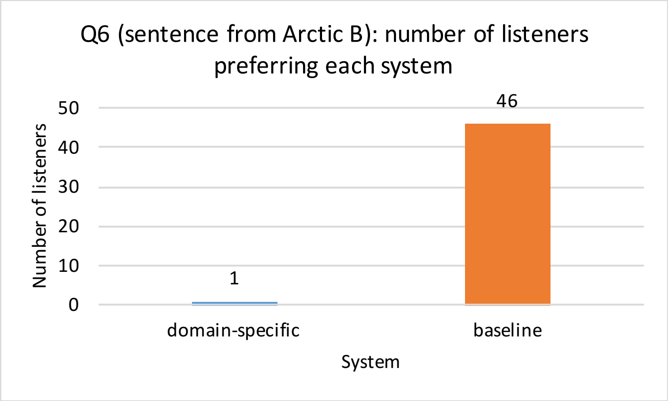


Fig. 12. Number of listeners preferring each system in Q6

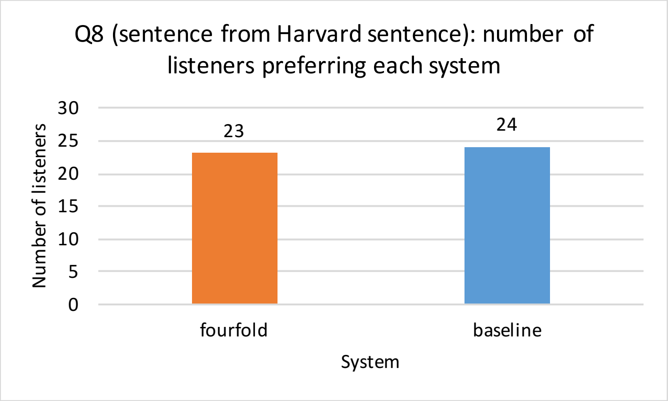


Fig. 13. Number of listeners preferring each system in Q8

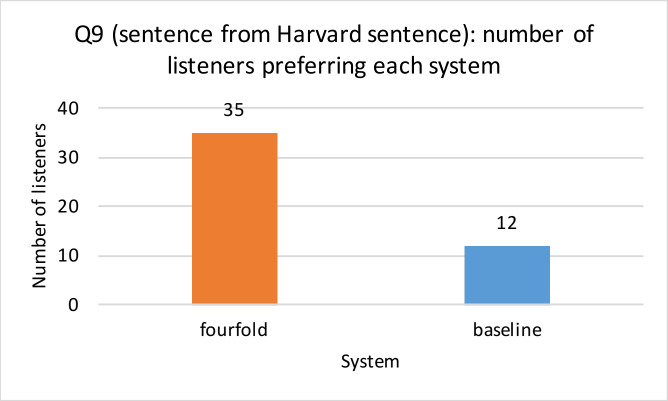


Fig. 14. Number of listeners preferring each system in Q9

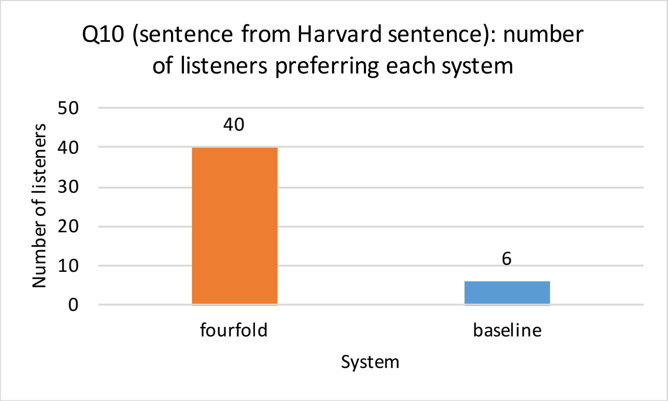


Fig. 15. Number of listeners preferring each system in Q10

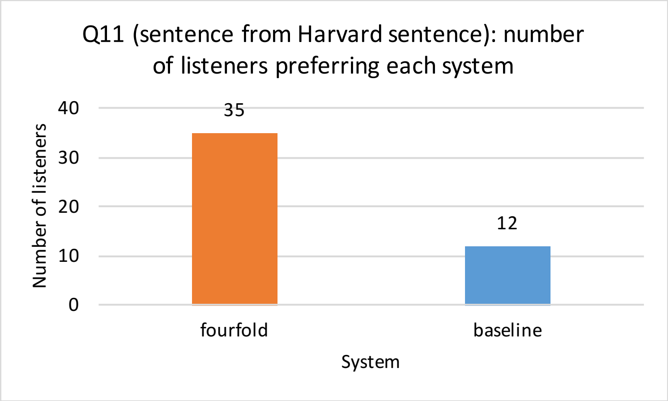


Fig. 16. Number of listeners preferring each system in Q11

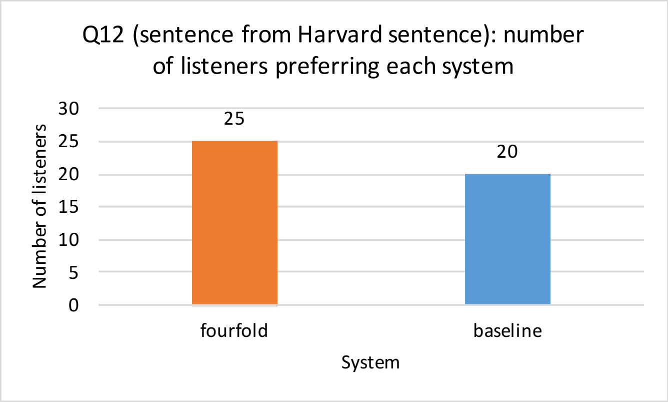


Fig. 17. Number of listeners preferring each system in Q12

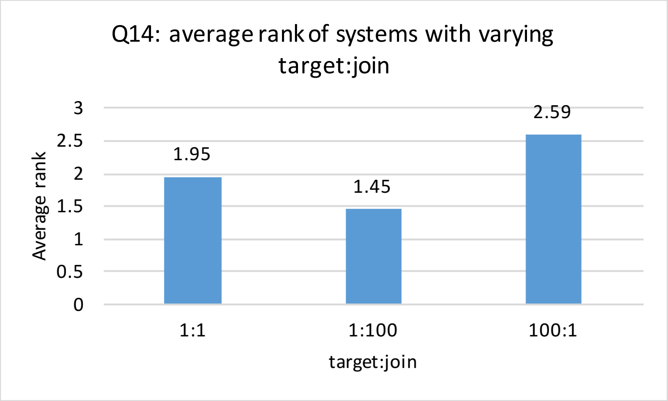


Fig. 18. Average rank of each system with varying *target:join* in Q14

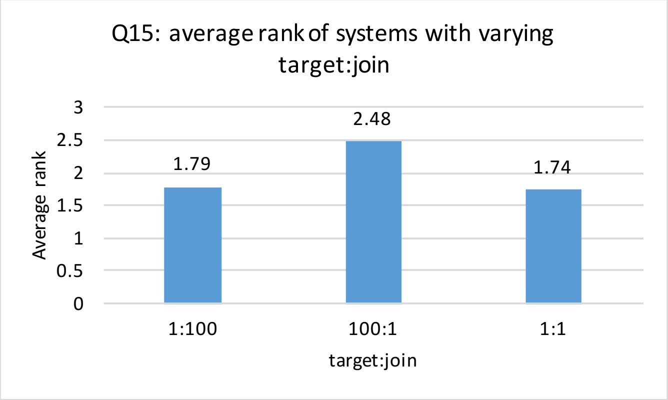


Fig. 19. Average rank of each system with varying *target:join* in Q15

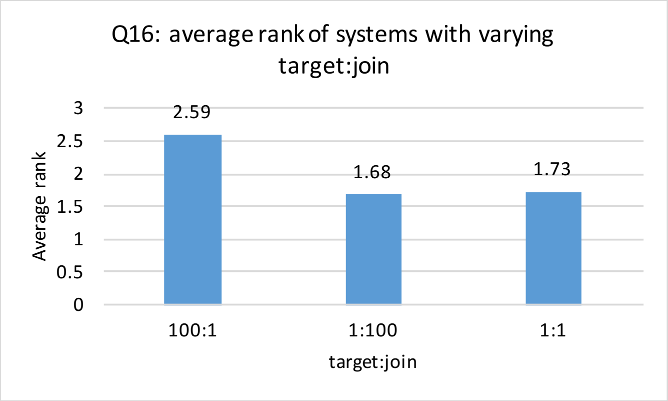


Fig. 20. Average rank of each system with varying *target:join* in Q16

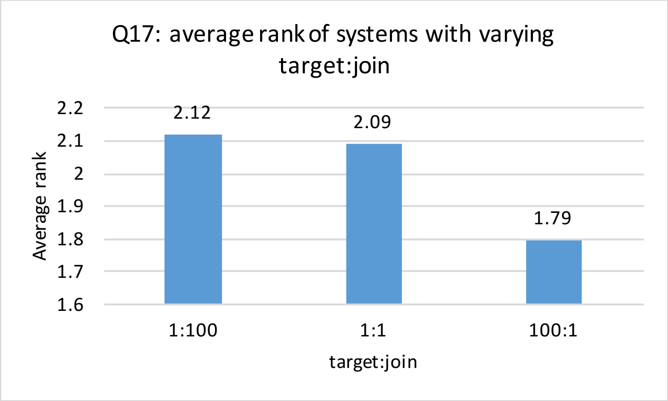


Fig. 21. Average rank of each system with varying *target:join* in Q17

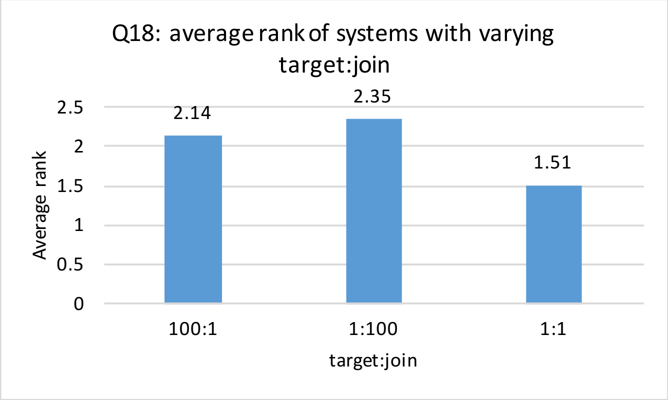


Fig. 22. Average rank of each system with varying *target:join* in Q18

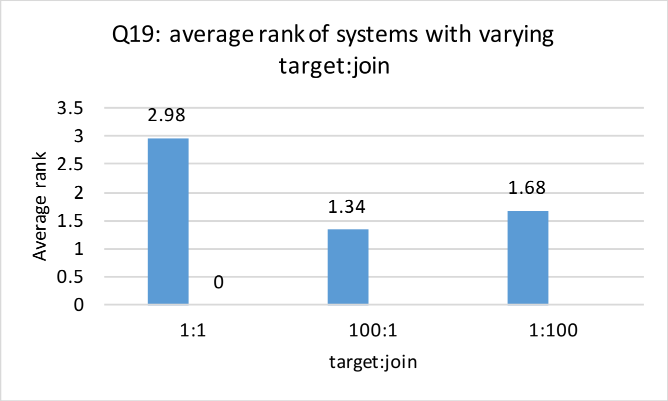


Fig. 23. Average rank of each system with varying *target:join* in Q19

References

[1] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 373–376.

[2] E. ZGK, “Human behavior and the principle of least-effort,” 1949.

[3] S. King, “A beginners guide to statistical parametric speech synthesis,” *The Centre for Speech Technology Research, University of Edinburgh, UK*, 2010.

[4] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[5] D. Jurafsky, “Speech and language processing: An introduction to natural language processing,” *Computational linguistics, and speech recognition*, 2000.

[6] R. A. Clark, K. Richmond, and S. King, “Multisyn: Open-domain unit selection for the festival speech synthesis system,” *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[7] H. Melin and J. Lindberg, “Variance flooring, scaling and tying for text-dependent speaker verification,” in *Sixth European Conference on Speech Communication and Technology*, 1999.

[8] Y. Medan, E. Yair, and D. Chazan, “Super resolution pitch determination of speech signals,” *IEEE transactions on signal processing*, vol. 39, no. 1, pp. 40–48, 1991.

[9] D. Talkin, “A robust algorithm for pitch tracking (rapt),” *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[10] L. Rabiner, “On the use of autocorrelation analysis for pitch detection,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 24–33, 1977.

[11] J. Kominek and A. W. Black, “The cmu arctic speech databases,” in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[12] C. Benoît, M. Grice, and V. Hazan, “The sus test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences,” *Speech Communication*, vol. 18, no. 4, pp. 381–392, 1996.

[13] H. Ze, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.