Simple Speech Representation Learning from Perceptual Data

Kanru Hua
Dreamtonics Co., Ltd.

February 2019

Abstract

Efforts have been made to design or discover more perceptually correlated speech features when standard feature extraction methods (e.g. MFCC and MCEP) become the bottleneck of speech synthesis systems. This study does not attempt to design a new feature extraction algorithm from scratch but instead choses to characterize human ratings of audio similarity using a purely behavioral (and statistical) approach. I designed and ran a listening test on Amazon Mechanical Turk and trained various regression models transforming mel-spectra into an optimized representation that better maps perceptual similarity on to an Euclidean space. I show that a carefully designed shallow neural network can predict human ratings reasonably well given limited training data. The feature transformations discovered in this study yet have to be evaluated on speech processing systems.

1 Introduction

This study follows a line of research [1, 2, 3, 4] on detecting concatenation discontinuities in unit selection speech synthesis systems that was seemingly forgotten at some point when statistical parametric systems took over. However, in resource limited domains and in applications requiring manual intervention/manipulation of intermediate results such as singing synthesis, unit selection methods still exist as a compelling alternative. In the case of hybrid systems, the boundary between the two major paradigms may be blurry.

Motivated by the recent advent of deep learning frameworks with automatic differentiation, differentiable dynamic-programming methods and crowdsourcing platforms, I believe it would be interesting to continue the research on discontinuity detection but in a broader context so the result may take the form of a plug-and-play component for speech processing systems.

The task of discontinuity detection can be summarized as the follows. In a unit selection system [5], short segments of speech (usually diphones) are selected from a database by minimizing the sum of a context distance (target cost) describing how likely the unit appears in the given phonetic and linguistic context and an acoustic distance (join cost) describing how well the unit matches its left/right neighbors so concatenation artifacts will not be heard. Older systems use hand-crafted criterions for target cost and dynamic time warping (DTW) distances on MFCC and F0 features for join cost. In recent incarnations of the system, both target cost and join cost can be predicted by deep neural networks (DNN) [6, 7].

To make sure that unit sequences with low acoustic distances at the joints indeed lead to smoother results, it is often preferred that the model-predicted join cost agrees with human perception of speech similarity. Thus, there had been various attempts at quantifying such perception, for example, choosing the best-performing distance metric [2, 4] and weighting several sub-costs based on perceptual experiments [1]. However, I found it hard to apply the findings from one setup to another since the perceptual experiment is often based on a specific speech synthesis system, let alone the possibility of using the perceptual data for purposes other than unit selection speech synthesis.

Proposal As opposed to finding task-specific weights or functions optimizing the naturalness of a particular speech synthesis system, I seek feature transformations that, in principle, can be plugged into any speech

processing frontend. I designed a naive experiment asking subjects for an 1-to-5 rating of the similarity between two isolated monophone instances. Data collected from the said experiment is used to train a function that transforms a sequence of speech feature of choice (e.g. MFCC, MCEP or even WORLD features for vocoding) into an intermediate feature, and a second function mapping the DTW distance between the intermediate features to subjective ratings.

Assumption The proposed method relies on two assumptions. First, the naive experiment should reveal enough information about perceptual similarity (control of variance). Second, the isolated testing of tokens should not induce a significant bias (control of bias). In the following sections, I will characterize the variance in the ratings and show how to reduce the randomness. However, the issue regarding bias is more complicated and left to subjective judgement and future studies testing the discovered feature transformations on actual applications.

2 Experiment Design

This study uses VCTK database [1] which consists of 108 English speakers and 44k utterances in total. From each speaker, a list of random phoneme instances are selected and paired with another random list (without replacement). The second list of phonemes are chosen from phoneme pairs with low DTW-MFCC distances so that the distribution of ratings becomes skewed toward the higher (more similar) end to focus more on the nuanced differences.

Each time a pair of phoneme instances (referred to as "token pair" in the follows) is presented to a test participant. The two phonemes are isolated from context with 3 ms fade in/out on both ends. By pressing the play button, the listener will hear the first phoneme, followed almost immediately by the second phoneme after a 0.5 second pause. Unlimited replay is allowed. The listener is asked to rate the similarity on a 5-point discrete opinion scale (Table 1).

Rating	Category	Description
5	Identical	they sound the same
4	Very Similar	almost the same but distinguishable
3	Fair	similar but not the same
2	Poor	not so similar but related
1	Bad	completely different sounds

Table 1: Choices presented to the listeners.

All listening tests in this study are carried out on Amazon Mechanical Turk (AMT) [2], a crowdsourcing platform frequently used for data preparation. Although it has been reported that AMT is a "viable" option for speech intelligibility tests [8], in this experiment where the similarity objective lacks a clear definition, it was still necessary to run a small scale screening to verify the setup and to determine how much data is needed for producing any meaningful result. As a naive measure to filter out unqualified listeners, a minimum approval rate of 97% is required to particpate in the tests.

3 Preliminary Study

Before data collection for model training, I conducted a few small scale tests to verify the aforementioned design. In particular, concerning that monophone tokens might be too short for reliable identification, I compared the distribution of ratings from monophone token pairs to those from diphone token pairs. Each

[1] http://dx.doi.org/10.7488/ds/1994

[2] https://www.mturk.com/

test involves a total of 61 token pairs from 10 VCTK speakers (5 male and 5 female), with each token pair rated by 25 participants.

Part of the responses to the monophone test case are showned in Figure 1. For each token pair, a beta-binomial distribution is fitted on the 25 ratings using max-likelihood estimation (slightly regularized to avoid extreme values). In general, I observed three trends in the histograms: left-skewed (33, 42), right-skewed (55), and ambiguous (10, 28, 50, 56) with the rest of the cases being a mix of them.

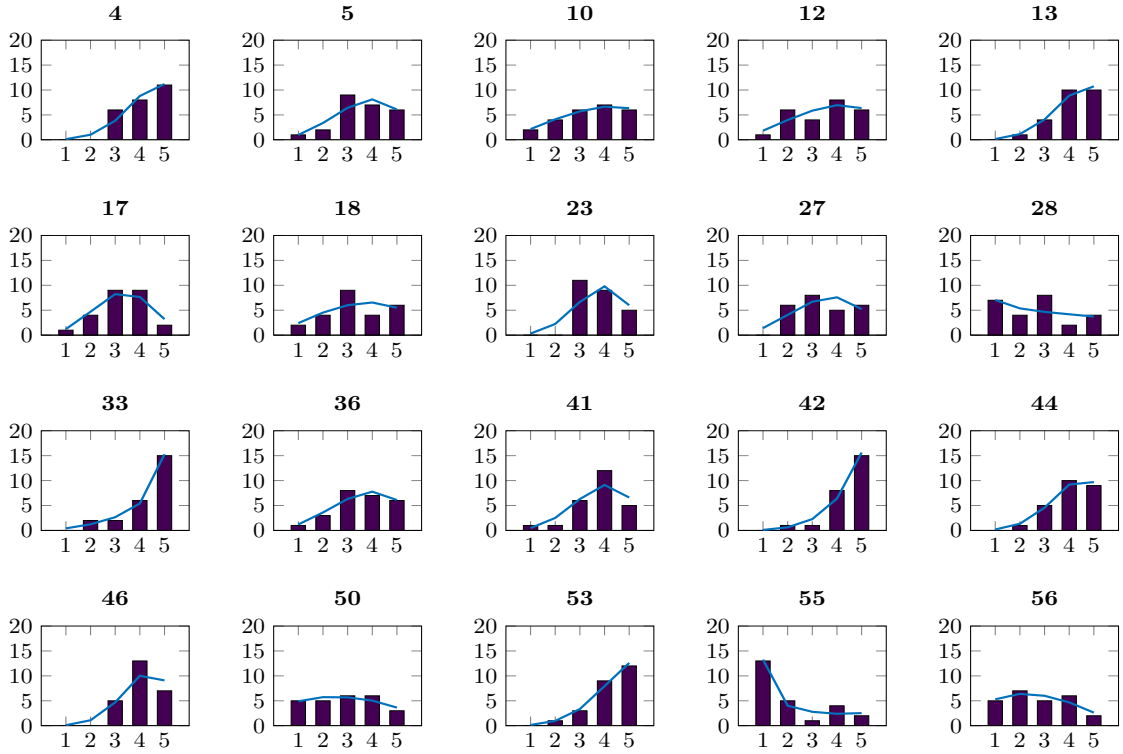
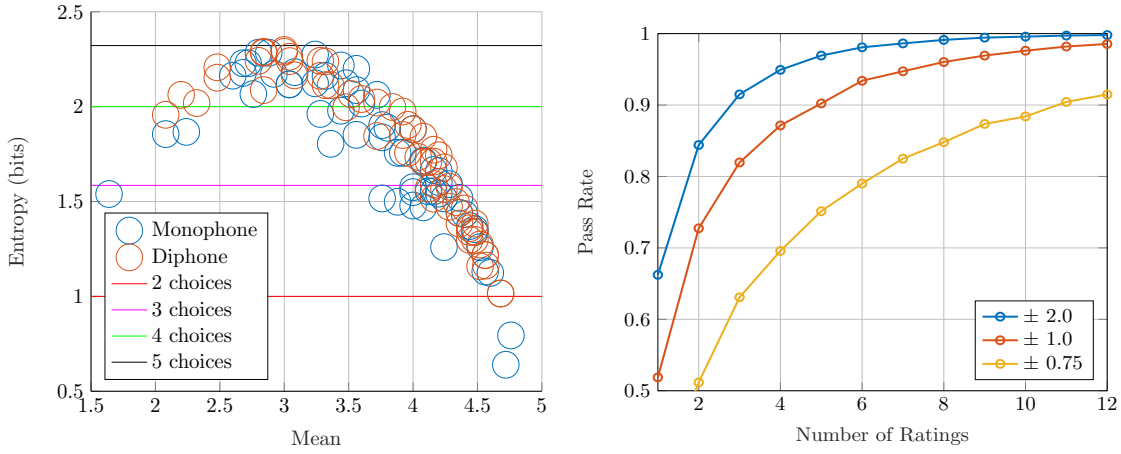


Figure 1: Histograms of similarity ratings for 20 randomly chosen (out of 61) token pairs with PMF of fitted beta-binomial distributions on the top.

Shown in Figure 2a is a scatter plot of the entropy versus mean taken from all histograms. For "very similar" and "identical" rated token pairs, the listeners give more consistent responses compared to the less similar ones. I also observed that the diphone test results are similar to the monophone test in terms of randomness but overall the ratings lean slightly more to the left (less similar) side. The rest of this study continued to use monophone tokens for listening test.

Next, I am interested in knowing how many independent ratings are needed for each token pair to get a good estimate. The idea is to have multiple participants (not necessary to be the same group of people) rate each token pair and take the average to reduce variance. This process is simulated by drawing samples from the PMFs computed from the 25-participant experiment (assuming that 25 samples are enough to approximate the true distribution). Figure 2b plots the accuracy versus sample size. It is seen that each listener gives extremely unreliable ratings but the accuracy can be significantly improved by increasing the number of measurements. Notably, at 5 ratings per token pair, 90% averaged ratings are within ± 1 from the population mean opinion score (MOS).

It is also possible to estimate the upperbound for Pearson correlation between the population MOS and predicted scores, or the lowerbound for mean square error, which will be a useful reference for predicting MOS from acoustic signals. Intuitively, this is to estimate how much error comes from noise rather than the regression model. Again, using Monte-Carlo simulation I found that at 5 ratings per pair, the upperbound



(a) (b)

Figure 2: (a) Distribution of mean and entropy of all token pairs in monophone and diphone tests. (b) Probability of the sample MOS being within a certain radius from the population MOS as a function of number of ratings for each token pair (monophone case).

for Pearson correlation coefficient is 0.825 and the lowerbound for MSE is 0.227; at 25 ratings per pair, the correlation rises to 0.957 while the MSE drops to 0.045. The following analysis can thus be put into a meaningful context by comparing the results to these numbers.

4 Regression Analysis

Based on findings from the preliminary study, I conducted the full-scale test on AMT using all speakers from VCTK. The database is divided into a training set with 90 speakers and a test set with the rest 18 speakers. A total of 3200 monophone token pairs (2700 for training and 500 for testing) are selected and each pair is rated by 5 listeners. The training set is further divided into five parts of the same size (540 pairs each), among which 4 are used for training and the one left for validation.

Due to data scarcity, the regression models primiarily considered here for feature transformation are linear or shallow neural networks except for a few deeper architectures for comparison. Models with increasing complexity (evolving from linear to non-linear) are tuned using 5-fold cross validation. The performance on test set is not measured until the completion of all following experiments.

All models are trained using gradient descent under a shared framework. The main challenge is to find a differentiable function for comparing two vector sequences of different lengths. I found SoftDTW[3] [9], an HMM-like approximation to dynamic time warpping a working choice for the said task. Another problem is to map SoftDTW distances (which can be negative due to log-sum-exp relaxiation of the max operator) to the MOS scale. Via trial and error, I arrived at a log-based three-parameter warpping function (1).

$$w(x) = \log(\max(\alpha x + \beta, 0.01)) + \gamma \tag{1}$$

With all building blocks ready, the MOS-predicting function is built-up as,

$$\text{MOS}(X, Y) = w(\text{SoftDTW}_t[T(X), T(Y)]) \tag{2}$$

where X and Y are input acoustic features; $T(\cdot)$ is the feature transformation to be learned; $t = 0.01$ is the temperature constant for SoftDTW, which internally uses Euclidean distance. Note that T may change

[3]Readers may find my PyTorch implementation of SoftDTW helpful `https://github.com/Sleepwalking/pytorch-softdtw`.

the dimensionality. Parameters for the warpping function (α, β and γ) are trained alongside T using Adam optimizer by minimizing mean square error of MOS. I found the choice of loss function (L1, L2 or even cross entropy loss on PMFs directly predicted from SoftDTW distances) to have a negligible effect on the result.

The following sections present regression results obtained from different combinations of source features and transformation functions. Shown in the tables, validation MSE and correlation are the average across 5-folds cross validation. For test MSE and correlation, the models are trained on all training data and evaluated on the test set containing unseen speakers.

4.1 Source Features without Transformation

I first evaluate the baseline performance for various audio features by setting $T(x) = x$ and only optimizing the warpping function parameters (Table 2). Three features derived from magnitude spectra are considered: Mel-frequency cepstral coefficients (MFCC), Mel-cepstal coefficients[4] (MCEP) and Mel-frequeny band energy (MFBE, i.e., MFCC before taking DCT). MFCC and MFBE are directly computed from magnitude spectrum; MCEP is computed from WORLD[5] spectral envelope, which smoothens out the harmonics. Features with zero mean unit variance normalization are also tested. Finally, the normalized case also includes 65-dimensional WORLD vocoder parameters using the default coding scheme (continuous F0 and binary voicing status, frequency warpping and truncated DCT for spectral envelope, band aperiodicity).

In terms of structure the four features can be categorized as spacially sparse spectrum-like features (MFBE) and spacially dense cepstrum-like features (MFCC, MCEP and WORLD). In the latter case, 13-dimensional MFCC is the most compact representation, followed by 25-dimensional MCEP. Note that WORLD includes not only spectral (60-dim) but also source features (5-dim) and thus it retains more information, though not scaled and arranged in a particularly perception-oriented manner.

Source Feature	Normalization	Validation MSE	Validation Corr.	Test MSE	Test Corr.
MFCC (13-dim)	No	0.339	0.611	0.332	0.655
MCEP (25-dim)	No	0.337	0.615	0.327	0.662
MFBE (50-dim)	No	0.342	0.606	0.336	0.650
MFCC (13-dim)	Yes	0.361	0.575	0.370	0.608
MCEP (25-dim)	Yes	0.392	0.524	0.378	0.602
MFBE (50-dim)	Yes	0.353	0.589	0.348	0.634
WORLD (65-dim)	Yes	0.465	0.375	0.463	0.464

Table 2: Performance of the baseline model with different source features. Except for WORLD, unnormalized features in general are more perceptually correlated than their mean-variance normalized counterparts. This observation agrees with the intuition that normalization amplifies quiet parts with lower SNR and less perceptual importance. I also see that the performance on test set is significantly better than the validation set despite the setup where all datasets were constructed from the same source (VCTK). Inspection shows that the test set has a few speakers whose results are more easily predictable than the average. However, the model-wise trend in the last four columns of Table 2 is still consistent between datasets. Finally, there is a wide gap between the performance of raw features and the theoretical upperbound (0.227 for MSE and 0.825 for correlation, see Section 3). Although it is not known to what degree the screening dataset suffers from the speaker randomness issue, a 0.16-0.25 difference in correlation can be regarded as an evidence of plenty of room for improvement.

4.2 Linear Transformation

The first set of feature transformations evaluated are completely linear as I expect that with increasing complexity the performance gets quickly saturated by overfitting. In the following experiments, I first

[4]Extracted using SPTK (`http://sp-tk.sourceforge.net/`).

[5]`https://github.com/mmorise/World`

evaluated dense transformation matrices (i.e. fully-connected layers, Table 3) and then reduced the degree of freedom by using a low-rank-plus-diagonal (LRpD) decomposition (Table 4), which has been reported to work well on speaker adaptation tasks [10]. I also tested the projection of a feature to a higher dimensional space before taking SoftDTW in the dense matrices case.

Note that mean-variance normalization is used for all features in all following experiments.

Source Feature	Projected Dimensionality	Validation MSE	Validation Corr.
MFCC (13-dim)	13	0.321	0.637
MCEP (25-dim)	25	0.317	0.644
MFBE (50-dim)	50	0.318	0.642
	25	0.318	0.643
WORLD (65-dim)	65	0.310	0.655
	25	0.310	0.655

Table 3: Performance of dense matrix multiplication models. In all cases a moderate improvement from the baseline can be observed. It also appears that with dense transformation matrices the projected dimensionality does not really matter (above a certain order). MFCC-13 feature with the lowest dimenionality suffers loss of information due to cepstral truncation and thus behaves poorer than MFBE-50. The result from MCEP-25 is not so much different from MFBE-50. WORLD parameters, however, rise from the most weakly correlated feature to the best fitting feature.

Source Feature	Rank	Validation MSE	Validation Corr.
MCEP (25-dim)	0	0.312	0.652
	1	0.314	0.649
	5	0.312	0.652
MFBE (50-dim)	0	0.333	0.619
	1	0.312	0.651
	5	0.312	0.651
WORLD (65-dim)	0	0.347	0.600
	1	0.296	0.674
	5	0.300	0.669

Table 4: Performance of linear LRpD models of different ranks. 0-rank LRpD matrices are effectively diagonal, i.e., independent scaling for each feature dimension. The performance of MCEP does not seem to be affected by rank. For MFBE, off-diagonal entries definitely help but the correlation cease to increase as rank grows beyond 1. For WORLD, the rank has a more significant impact and the rank 1 version performs vastly better than the parameters in their original form. It is possible that any rank between 2 and 4 could give an even better fit but I did not attempt an exhaustive search. Overall, the spectral sparseness of LRpD does bring an advantage over dense matrices and rank-1 LRpD seems to perform well on all source features.

4.3 Perceptron and Multi-Layer Perceptron

Without increasing the number of parameters (except for a bias vector), I added a non-linear activation function to the dense linear transformation models ("M x N - ReLU / Tanh" entries in Table 5). Then I explored neural networks with one hidden layer by adding a second linear transformation. Finally, the 1-hidden layer models are enhanced with an input-to-output residual connection.

While the 1-hidden-layer networks fail to outperform perceptrons, there's still a possibility of incorporating LRpD layers into the architecture to mitigate overfitting. Table 6 lists the results of fully and partially

Source Feature	Architecture	Validation MSE	Validation Corr.
MCEP (25-dim)	25 x 50 - ReLU	0.306	0.660
	25 x 50 - Tanh	0.303	0.662
	25 x 50 - ReLU - 50 x 50	0.319	0.643
	25 x 50 - ReLU - 50 x 25	0.315	0.648
	25 x 50 - ReLU - 50 x 25 Res.	0.309	0.656
MFBE (50-dim)	50 x 50 - ReLU	0.293	0.678
	50 x 80 - ReLU	0.293	0.678
	50 x 50 - ReLU - 50 x 50	0.300	0.668
	50 x 50 - ReLU - 50 x 50 Res.	0.291	0.679
WORLD (65-dim)	65 x 50 - ReLU	0.289	0.683
	65 x 50 - ReLU - 50 x 65	0.302	0.667
	65 x 50 - ReLU - 50 x 65 Res.	0.291	0.681

Table 5: Performance of (multi-layer) perceptron models. "Res" stands for residual connection between input (or the first applicable layer) and output. A non-linear layer by itself (i.e. perceptron) leads to an immediate improvement from the linear transformation versions. The choice of activation function has a marginal effect on MSE and correlation. Adding the second linear transformation to become a shallow nerual network, however, worsens the performance likely due to overfitting. This disadvantage is mitigated by input-to-output residual connection, although there's still no substantial difference compared to single-layer perceptrons.

LRpD networks. I also considered first upscaling the feature using a 25 x 50 fully connected layer in the case of MCEP-25.

Source Feature	Architecture	Validation MSE	Validation Corr.
MCEP (25-dim)	LRpD - ReLU - LRpD Res.	0.315	0.648
	25 x 50 - ReLU - LRpD	0.305	0.661
	25 x 50 - LRpD - ReLU - LRpD Res.	0.301	0.667
	25 x 50 - ReLU - LRpD - ReLU - LRpD Res.	0.302	0.665
MFBE (50-dim)	LRpD - ReLU - LRpD Res.	0.285	0.687
	LRpD - ReLU - 50 x 50 Res.	0.286	0.686
	50 x 50 - ReLU - LRpD Res.	0.291	0.680
WORLD (65-dim)	LRpD - ReLU - LRpD Res.	0.284	0.690
	65 x 50 - LRpD - ReLU - LRpD Res.	0.294	0.678

Table 6: Performance of multi-layer perceptrons with Rank-1 LRpD layer(s). For the lower-dimensional MCEP-25 feature, there's little room of improvement for LRpD residual networks although with the initial upscaling layer a small improvement (0.662 to 0.667 in correlation) from the best perceptron model is observed. On the higher dimensional MFBE-50 and WORLD-65 features correlations increased by ~ 0.007. The last two rows in MFBE show that LRpD for the input connection is more important than for the output (which makes sense on MFBE because LRpD preserves the spacial sparseness of the input feature).

4.4 RNN Sequence Encoder

Although it is of less relevance to use sequence-to-vector encoders for this study, I nevertheless tested RNN-based encoders, including models pre-trained as autoencoders for a comparison with shallower models. Note that the setups differ in that sequence encoders transform a sequence of feature vectors into one embedding vector (which is the hidden cell states at the last frame) while the previous models produce sequences of the same length as the input. Consequently, SoftDTW is no longer needed and is replaced by a simple Euclidean

distance measure between sequence embeddings.

The sequence encoder used in this study follows the acoustic encoder in [6], with the exception that LSTM cells are replaced by GRU cells for fewer parameters. I tested encoders trained directly on ratings with varying size of hidden layers and encoders pre-trained on VCTK for the task of autoencoding (Figure 3). Unfortunately, the best sequence encoding model hardly even beats the baseline.

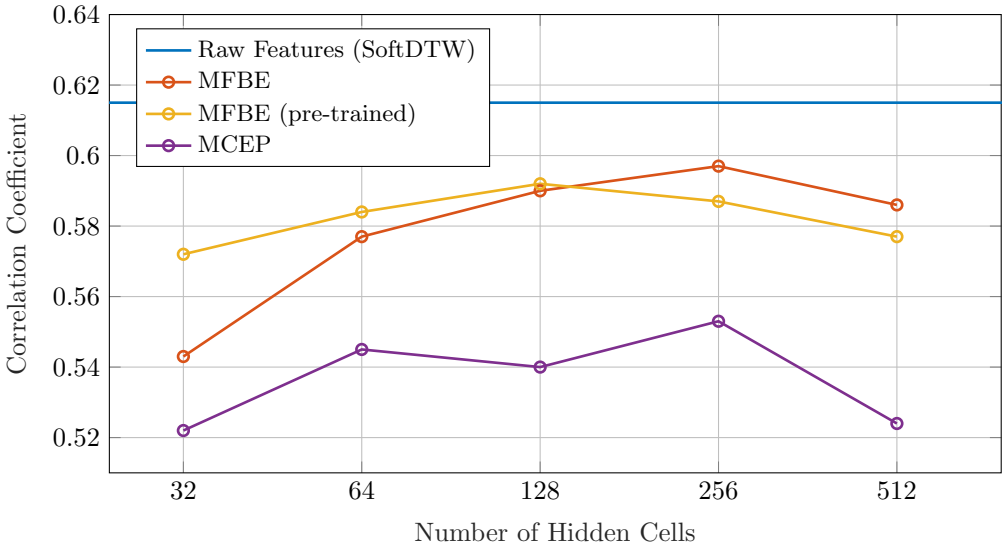


Figure 3: Performance of RNN sequence encoders as a function of hidden layer size. "MFBE" and "MCEP" series are models directly trained on ratings; "MFBE (pre-trained)" model is first trained as autoencoder on the full training set and then served as an initialization for models used for cross validation. MFBE feature consistently outperforms MCEP, although none matches the baseline (SoftDTW on unnormalized MCEP feature). Pre-training only helps for smaller models.

4.5 Other Architectures (that did not work)

Convolutional neural networks I trained shallow fully-convolutional networks with a short filter length of 3. I also combined FCNs with LRpD layers and/or residual connections. However, for all configurations the correlation slightly lags behind that of their MLP versions.

Deep denoising autoencoder I trained a 6-hidden-layer denoising autoencoder on MFBE-50 feature and repeated the experiments in the previous sections on bottleneck activations (50-dim). Again, the best results lag slightly behind that of shallow LRpD residual networks. I am fairly convinced that with the amount of data available, the better approach is to design partially enginneered and partially trainable architectures with expert knowledge, rather than completely relying on general-purpose models.

5 Testing and Verification

Having finished the architecture search, now from each category (except sequence encoders) I select the best-performing configuration for testing. For clarity, architectures and configurations are listed again in Table 7. The selected models are trained on the unpartitioned training set and evaluated on the test set. All models are trained for 10 epochs, a number I observed to be roughly optimal in the previous experiments.

Figure 4 shows the test results with reference to the baseline (unnormalized raw features) and the upperbound derived in the priminary study. I also tested models trained under the same configurations on

the 61-token pair data set used for the preliminary study (referred to as screening set)[6]. Each token pair in the screening set is rated by 25 listeners so the averaged ratings are less noisy than the test set, although there is a trade-off on the variety of token pairs. Results on the screening set shown in Figure 5 are more of an approximation to the overall performance on an ideal noise-free data set rather than any meaningful comparsion between models because of the small sample size.

Category	Configuration (MCEP)	Configuration (MFBE)	Configuration (WORLD)
Baseline	Unnormalized	Unnormalized	Normalized
Dense Matmul	25 x 25	50 x 50	65 x 65
Rank-1 LRpD	LRpD	LRpD	LRpD
MLP	25 x 50 - Tanh	50 x 50 - ReLU - 50 x 50 Res.	65 x 50 - ReLU
MLP + LRpD	25 x 50 - LRpD - ReLU - LRpD Res.	LRpD - ReLU - LRpD Res.	LRpD - ReLU - LRpD Res.

Table 7: Detailed model configuration for each category tested.

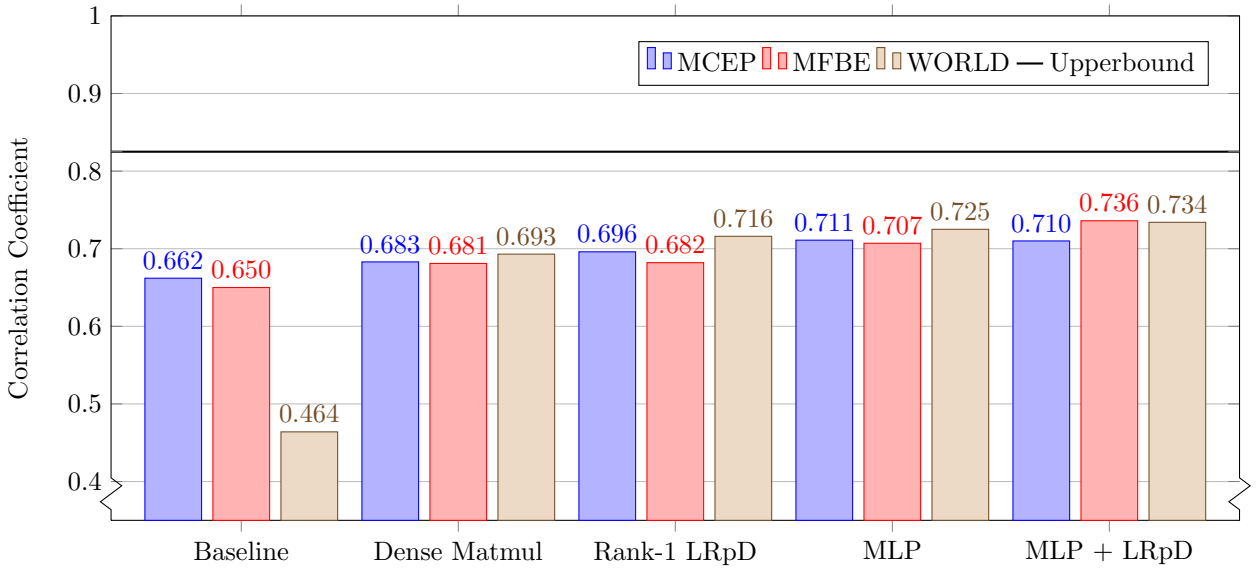


Figure 4: Comparing various combinations of features and models on the test set. Although the best fitting model (MLP + LRpD w/ MFBE) only shows a moderate improvement of 0.074 from the best baseline (unnormalized MCEP), the difference becomes more significant considering the upperbound at 0.825. In other words, the learned feature transformation reduces the gap between baseline and perfectly predicted ratings, whose correlation is still less than one due to noises in the target, by almost half. Among MCEP, MFBE and WORLD, it is hard to say which feature performs the best overall. In general the more engineered models (within the realm of shallow networks) give rise to higher correlations for all features.

The case of MLP + LRpD w/ MFBE is chosen for a more detailed analysis casting light on error patterns. Figure 6 compares the distributions of human ratings and predicted ratings for acoustic feature before and after transformation. Although the points are more densely populated near the diagonal in the second plot, a large variance persists in the lower ratings (1-3) range. One possible cause is that I tuned the database to

[6]Models evaluated on the screening set are trained on all ratings (training + test) but those on the 10 speakers who appeared in the screening set. The merged and filtered data set has 2955 token pairs from 98 speakers.

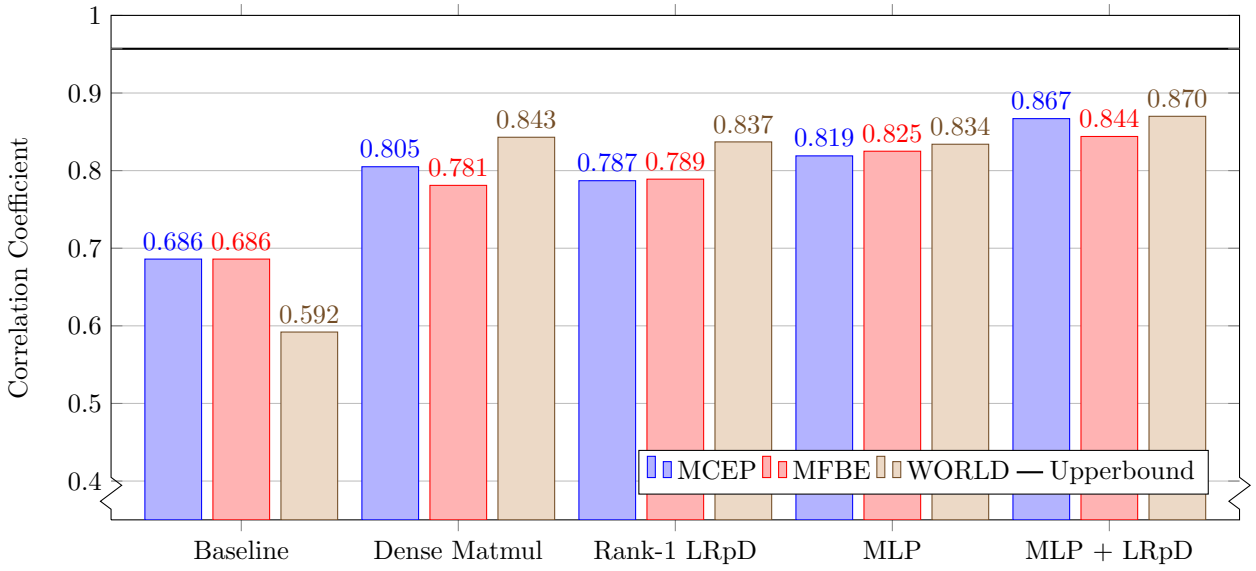


Figure 5: Comparing various combinations of features and models on the screening set. The trend agrees with the results on test set except the spike on Dense Matmul w/ WORLD and the underperforming MLP + LRpD w/ MFBE which probably occurred by chance. Feature transformation raises the correlation from the high 0.6s to the mid 0.8s.

focus more on small differences (ratings of 3-5) so there weren't enough samples for the models to generalize. By manual inspection of the outliers, I also discovered an error mode where one of the tokens is significantly longer/shorter than the other, or contains slience lasting a few hundred milliseconds.

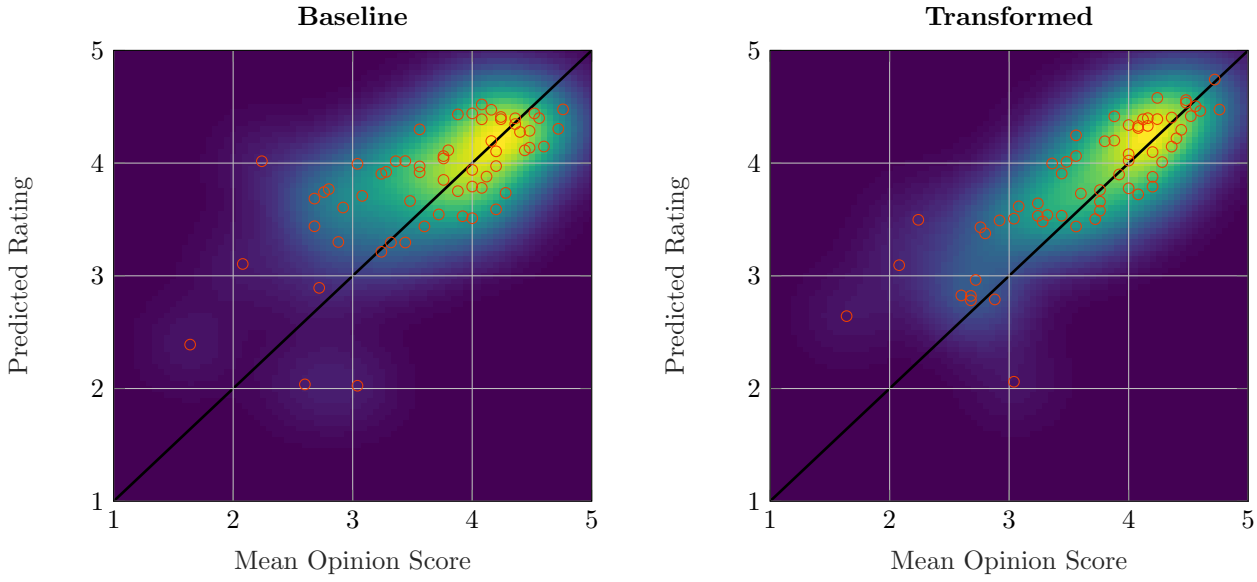


Figure 6: Scatter plots of human ratings versus predicted ratings for the baseline (unnormalized MFBE) and transformed features (MLP + LRpD), on the screening set. The background is kernel smoothed density of the distribution. In general, points in the plot for transformed features are closer to the diagonal.

Finally, it is interesting to know if the learned transformations bear any perceptual meaning. For the ease of visualization, I selected the linear LRpD model w/ MFBE for inspection. Since the off-diagonal entries only offer one degree of freedom in the rank-1 case, one can expect most of the variability to be conveyed through the diagonal, and the output should not look so different from the input. Figure 7 plots the weights on the diagonal and the left matrix (which is a column vector in this case). Firstly, it is seen that the diagonal emphasizes mid-range frequencies (0.5 - 3 kHz), although the weights corresponding to filterbank outputs above 2 kHz are somewhat noisy. Next, the left vector seems to pick up energy contrast between ~ 500 Hz and ~ 900 Hz, acting like a rudimentary 1st formant estimator. Similar structures are also found in non-linear models with LRpD layers.

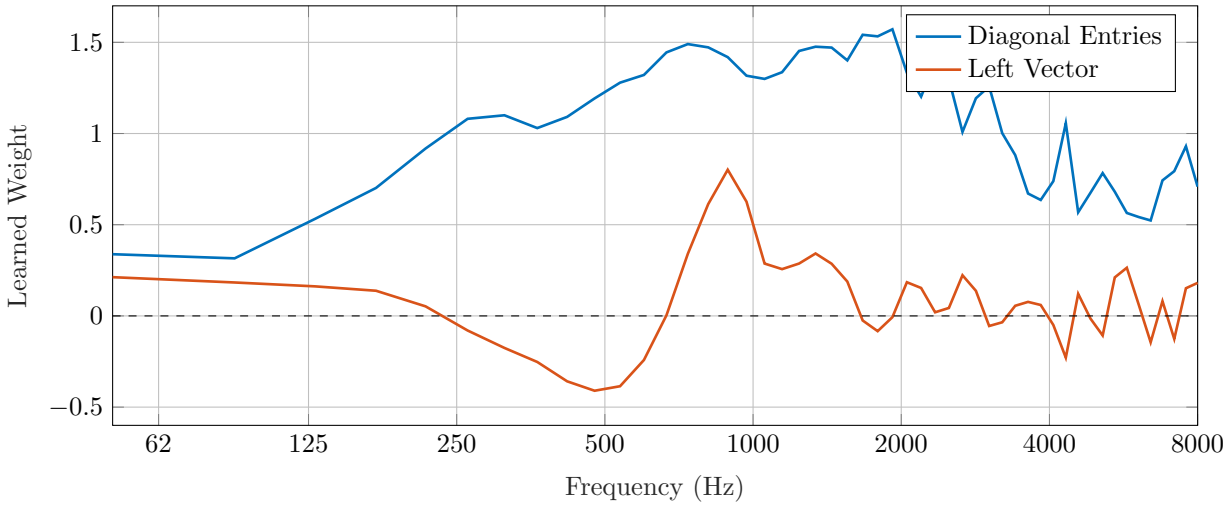


Figure 7: Visualizing learned weights in a rank-1 LRpD transformation trained on MFBE feature.

6 Conclusion

I collected human ratings of perceptual similarity between pairs of short speech segments and trained various feature transformation models to minimize the difference between DTW distance and human ratings. I showed that commonly used acoustic features can be improved for the said task with the assist of shallow neural networks. The weights of a trained network demonstrated some degree of perceptual relevance.

The current study lacks any testing of the trained feature transformations on actual speech synthesis/recognition systems. While I do not expect them to make much difference on DNN-based systems because the DNN may learn to compensate for simple feature distortions, it will be interesting to conduct a benchmark on hybrid or traditional unit selection systems, GMM-based voice conversion systems, HMM-based speech synthesizer and aligners, etc. For applications requiring reversal of the transformation (e.g. converting transformed feature back to WORLD parameters for synthesis), an inverse yet has to be derived, although I do not expect this to be difficult since most models in this study are shallow and simple enough to impose some form of invertibility constraint [11]. Besides direct usage of the transformed features, one may also make use of perceptual distances predicted by the proposed methods.

References

[1] J. Wouters and M. W. Macon. "A perceptual evaluation of distance measures for concatenative speech synthesis." in *5th International Conference on Spoken Language Processing,* 1998.

[2] Y. Stylianou, and A. K. Syrdal. "Perceptual and objective detection of discontinuities in concatenative speech synthesis," in *ICASSP 2001.*

[3] A. K. Syrdal and A. D. Conkie. "Data-driven perceptually based join costs," in *5th ISCA Workshop on Speech Synthesis,* 2004.

[4] J. Vepa and S. King. "Subjective evaluation of join cost and smoothing methods for unit selection speech synthesis." IEEE Trans. on Audio, Speech, and Language Processing, vol. 14, no. 5, p.p. 1763-1771, 2006.

[5] A. J. Hunt and A. W. Black. "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP 1996.*

[6] V. Wan, Y. Agiomyrgiannakis, H. Silen and J. Vit. "Google's Next-Generation Real-Time Unit-Selection Synthesizer Using Sequence-to-Sequence LSTM-Based Autoencoders," in *Interspeech 2017.*

[7] A. Perquin, G. Lecorvé, D. Lolive and L. Amsaleg. "Phone-level embeddings for unit selection speech synthesis," in *International Conference on Statistical Language and Speech Processing.* 2018.

[8] M. K. Wolters, K. B. Isaac and S. Renals. "Evaluating speech synthesis intelligibility using Amazon Mechanical Turk," in *7th ISCA Workshop on Speech Synthesis.* 2010.

[9] M. Cuturi and M. Blondel. "Soft-DTW: a differentiable loss function for time-series," in *34th International Conference on Machine Learning*, 2017.

[10] Y. Zhao, J. Li, and Y. Gong. "Low-rank plus diagonal adaptation for deep neural networks," in *ICASSP 2016.*

[11] J. Behrmann, D. Duvenaud and J-H Jacobsen. "Invertible residual networks." *arXiv preprint,* arXiv:1811.00995. 2018.