Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

**Name:-**Jay Ganesh Thakare

**Roll no.** 118

**Batch:-**B3

**Subject:-**AI

**Aim**: Case study of any Exiting Successful AI System.

**LO Mapped**: LO1. Explain Basic techniques to build intelligent systems.

LO2. Implement expert systems and AI Application for real world problems.

**Problem Statement:** One case study given to induvial student on AI related topics. Case Study should cover all the relevant points about the topic.

**Objective:** A case study analysis requires you to investigate a business problem, examine the alternative solutions, and propose the most effective solution using supporting evidence.

**Experiment No. 1:** Case study of any Exiting Successful AI System.

**Aim:** Study and analyze one case study on AI applications in Healthcare, Retail and Banking published in IEEE/ACM/Springer or any prominent journal.

**Theory:**

**Topic:-Deep Learning for Siri's Voice: On-device Deep Mixture Density Networks for Hybrid Unit Selection Synthesis.**

Siri is a personal assistant that communicates using speech synthesis. Starting in iOS 10 and continuing with new features in iOS 11, we base Siri voices on deep learning. The resulting voices are more natural, smoother, and allow Siri's personality to shine through. This article presents more details about the deep learning based technology behind Siri's voice.

**Introduction**

Speech synthesis—the artificial production of human speech—is widely used for various applications from assistive technology to gaming and entertainment. Recently, combined with speech recognition, speech synthesis has become an integral part of virtual personal assistants, such as Siri.

There are essentially two speech synthesis techniques used in the industry: unit selection  and parametric synthesis . Unit selection synthesis provides the highest quality given a sufficient amount of high-quality speech recordings , and thus it is the most widely used speech synthesis technique in commercial products. On the other hand, parametric synthesis provides highly intelligible and fluent speech, but suffers from lower overall quality. Therefore, parametric synthesis is often used when the corpus is small or a low footprint is required. Modern unit selection systems combine some of the benefits of the two approaches, and so are referred to as hybrid systems. Hybrid unit selection methods are similar to classical unit selection techniques, but they use the parametric approach to predict which units should be selected.

Recently, deep learning has gained momentum in the field of speech technology, largely surpassing conventional techniques, such as hidden Markov models (HMMs). Parametric synthesis has benefited greatly from deep learning technology. Deep learning has also enabled a completely new approach for speech synthesis called direct waveform modeling (for example using WaveNet ), which has the potential to provide both the high quality of unit selection synthesis and flexibility of parametric synthesis. However, given its extremely high computational cost, it is not yet feasible for a production system.

In order to provide the best possible quality for Siri's voices across all platforms, Apple is now taking a step forward to utilize deep learning in an on-device hybrid unit selection system.

**Problem Statement:**

Conventional text-to-speech (TTS) systems used in virtual assistants like Siri rely heavily on cloud-based models or unit selection synthesis, which struggle to balance natural-sounding speech, model size, and on-device real-time performance. These methods often fail to generate smooth, expressive, and contextually appropriate speech due to limited modeling of prosody and acoustic variability. The key problem is to develop a compact, efficient, and accurate deep learning model that can be deployed on-device to enhance the naturalness and expressiveness of voice synthesis in Siri without compromising performance or memory constraints.

**How Does Speech Synthesis Work?**

Building a high-quality text-to-speech (TTS) system for a personal assistant is not an easy task. The first phase is to find a professional voice talent whose voice is both pleasant and intelligible and fits the personality of Siri. In order to cover some of the vast variety of human speech, we first need to record 10—20 hours of speech in a professional studio. The recording scripts vary from audio books to navigation instructions, and from prompted answers to witty jokes. Typically, this natural speech cannot be used as it is recorded because it is impossible to record all possible utterances the assistant may speak. Thus, unit selection TTS is based on slicing the recorded speech into its elementary components, such as half-phones, and then recombining them according to the input text to create entirely new speech. In practice, selecting appropriate phone segments and joining them together is not easy, because the acoustic characteristics of each phone depend on its neighboring phones and the prosody of speech, which often makes the speech units incompatible with each other. Figure 1 illustrates how speech can be synthesized using a speech database segmented into half-phones.

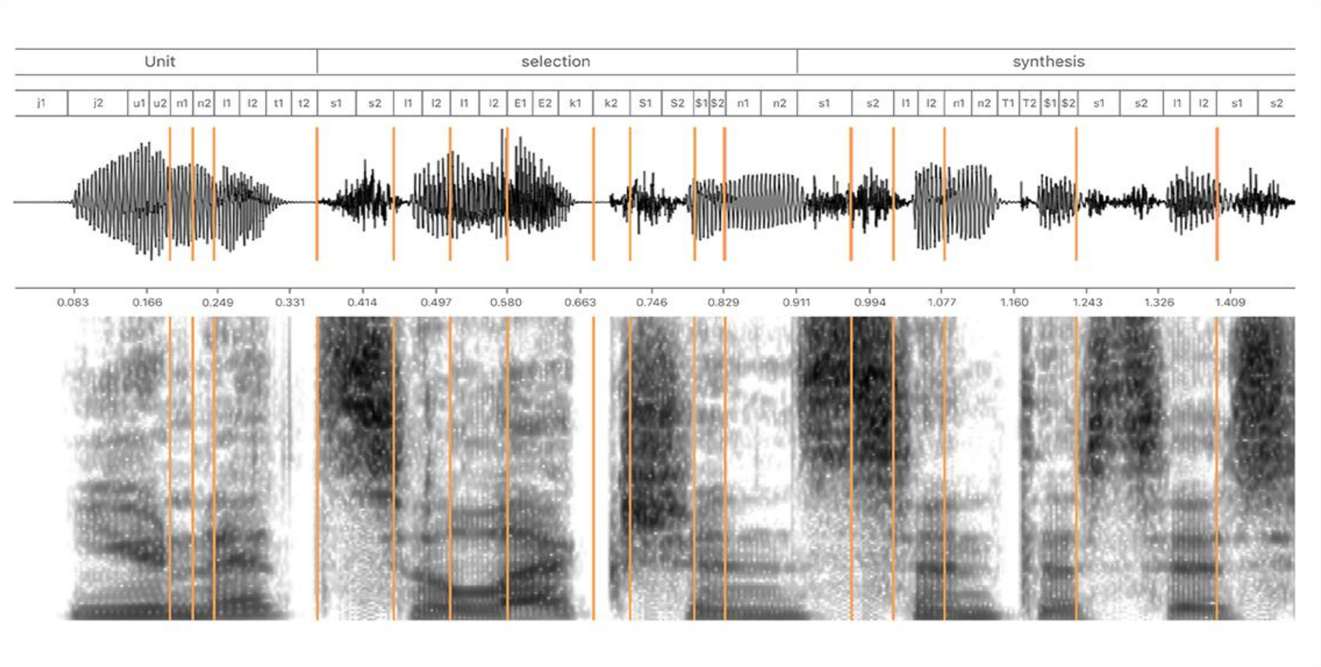

**Figure 1. Illustration of unit selection speech synthesis using half-phones. The synthesized utterance "Unit selection synthesis" and its phonetic transcription using half-phones are shown at the top of the figure. The corresponding synthetic waveform and its spectrogram are shown below. The speech segments delimited by the lines are continuous speech segments from the database that may contain one or more half-phones.**

The fundamental problem of unit selection TTS is to find a sequence of units (e.g. half-phones) that satisfy the input text and the predicted target prosody, provided that the units can be joined together without audible glitches. Traditionally, the process consists of two distinctive parts: front-end and back-end (see Figure 2), although in modern systems the boundary can sometimes be ambiguous. The purpose of the front-end is to provide phonetic transcription and prosodic information based on the raw text input. This includes normalizing the raw text which may include numbers, abbreviations, etc., into written out words, assigning phonetic transcriptions to each word, and parsing syntax, syllable, word, stress, and phrasing-related information from text. Note that the front-end is highly language dependent.

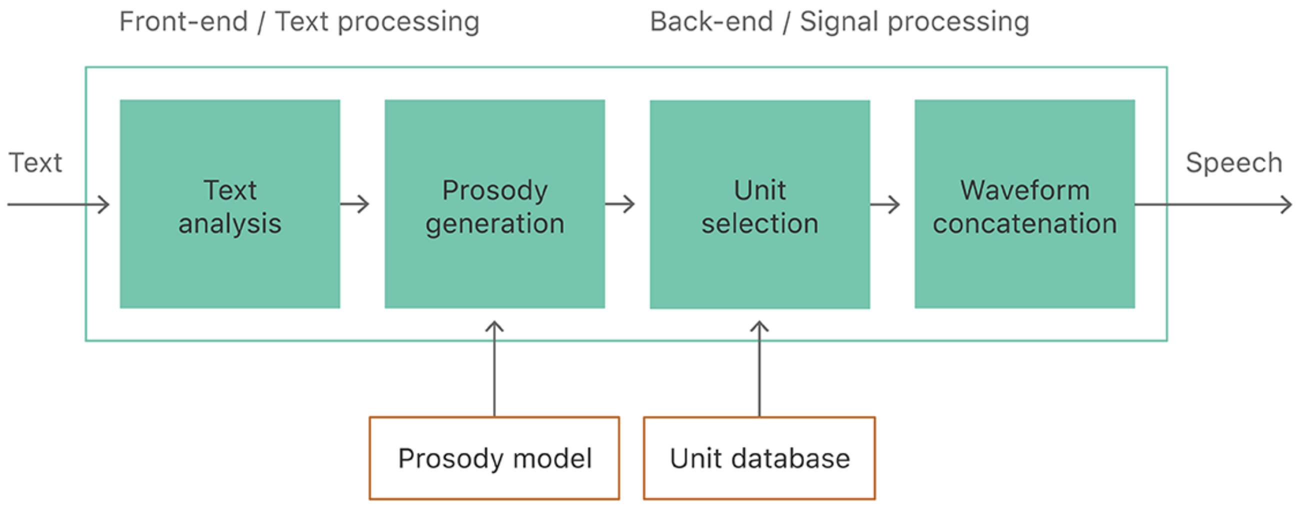

**Figure 2. Text-to-speech synthesis process**

Using the symbolic linguistic representation created by the text analysis module, the prosody generation module predicts values for acoustic features, such as intonation and duration. These values are used to select appropriate units. The task of unit selection has high complexity, so modern synthesizers use machine learning methods that can learn the correspondence between text and speech and then predict the values of speech features from the feature values of unseen text. This model must be learned at the training stage of a synthesizer using a large amount of text and speech data. The input to the prosody model are the numerical linguistic features, such as, phone identity, phone context, and syllable, word, and phrase-level positional features converted into convenient numerical form. The output of the model is composed of the numerical acoustic features of speech, such as spectrum, fundamental frequency, and phone duration. At synthesis time, the trained statistical model is used to map from the input text features into speech features, which are then used to guide the unit selection backend process where appropriate intonation and duration are crucial.

In contrast to the front-end, the backend is mostly language independent. It consists of unit selection and waveform concatenation parts. When the system is trained, the recorded speech data is segmented into individual speech segments using forced alignment between the recorded speech and the recording script (using speech recognition acoustic models). The segmented speech is then used to create a unit database. The unit database is further augmented with important information, such as the linguistic context and acoustic features of each unit. We refer to this data as the *unit index*. Using the constructed unit database and the predicted prosodic features that guide the selection process, a Viterbi search is performed in the speech unit space to find the best path of units for synthesis (see Figure 3).

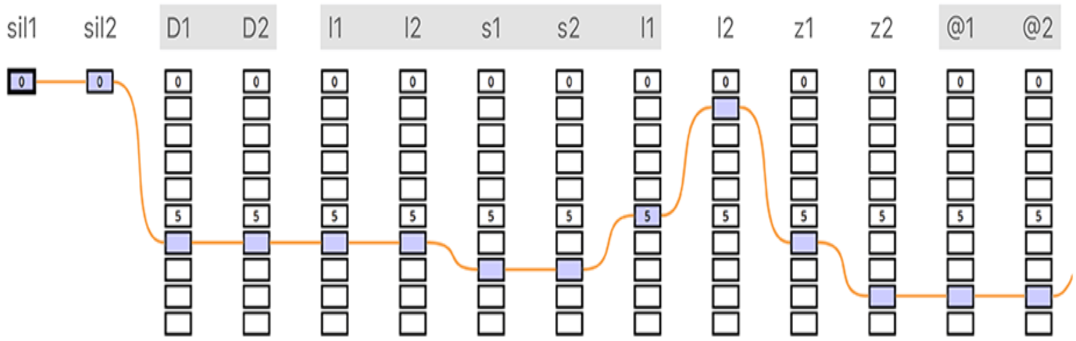

**Figure 3. Illustration of Viterbi search for finding the best path of units in the lattice. The target half-phones for synthesis are shown at the top of the figure, below which each box corresponds to an individual unit. The best path, as found by Viterbi search, is shown as a line connecting the selected units.**

The selection is based on two criteria: 1) the units must obey the target prosody and 2) the units should, wherever possible, be concatenated without audible glitches at the unit boundary. These two criteria are called the target and concatenation costs, respectively. The target cost is the difference between the predicted target acoustic features and the acoustic features extracted from each unit (stored in unit index), whereas the concatenation cost is the acoustic difference between consequent units (see Figure 4). The overall cost is calculated as follows:

$$C = w_t \sum_{n=1}^{N} \mathbf{T}(u_n) + w_c \sum_{n=1}^{N} \mathbf{C}(u_n, u_{n+1})$$

where $u_n$ is the nth unit, N is the number of units, $w_t$ and $w_c$ are the target and concatenation cost weights. After determining the optimal sequence of units, the individual unit waveforms are concatenated to create continuous synthetic speech.

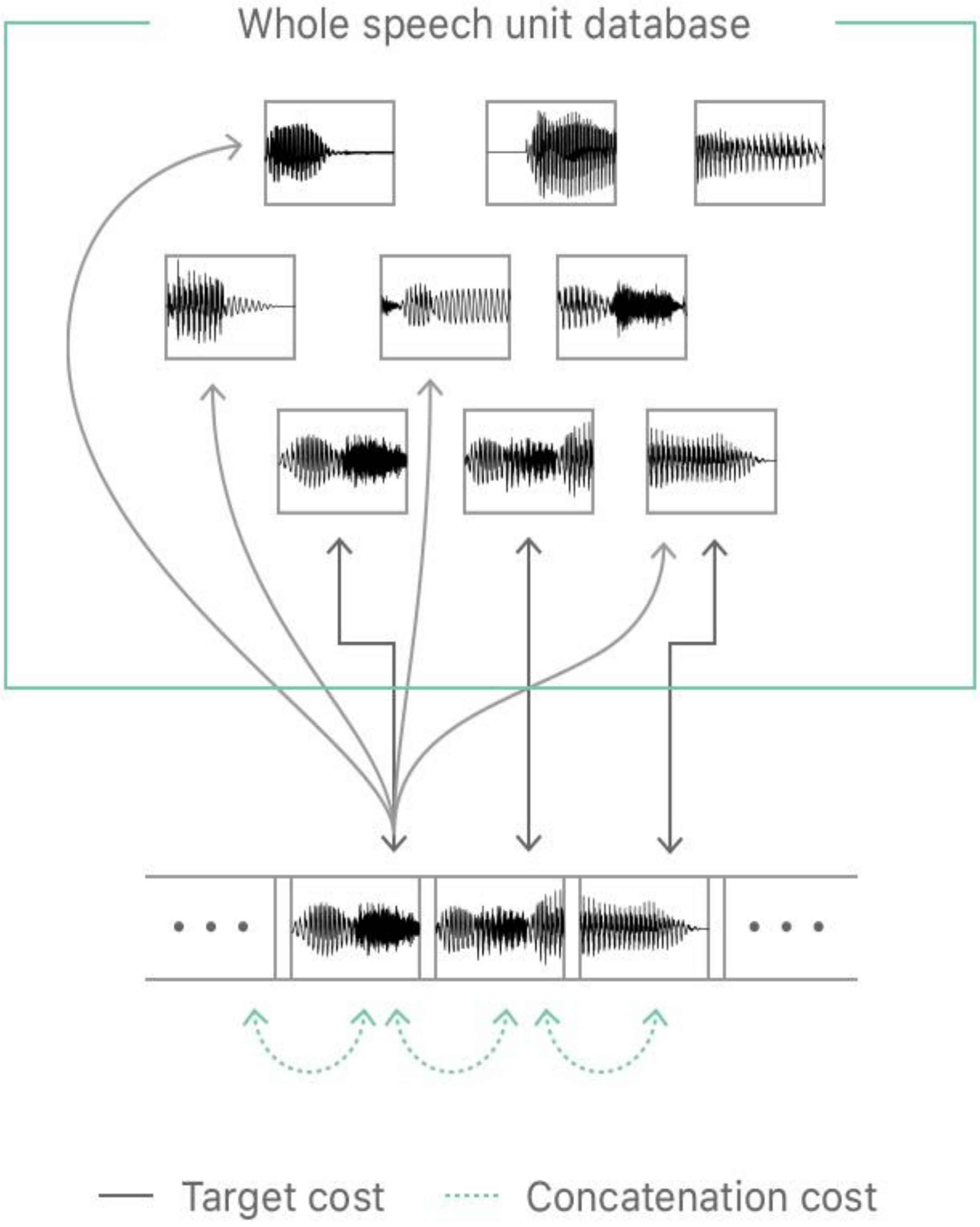
Whole speech unit database
• • •
• • •
Target cost
Concatenation cost

**Figure 4. Illustration of the unit selection method based on target and concatenation costs.**

### Technology Behind Siri’s New Voice

HMMs are commonly used as the statistical model for target predictions [5] [6] since they directly model the distributions of acoustic parameters, and thus for example the Kullback-Leibler divergence can be easily used to compute the target cost. However, deep learning-based approaches often outperform HMMs in parametric speech synthesis, and we expect the benefits of deep learning to be translated to hybrid unit selection synthesis as well.

The goal of Siri's TTS system is to train a unified model based on deep learning that can automatically and accurately predict both target and concatenation costs for the units in the database. Thus, instead of HMMs, the approach uses a deep mixture density network (MDN) to predict the distributions over the feature values. MDNs combine conventional deep neural networks (DNNs) with a Gaussian mixture models (GMM).

A conventional DNN is an artificial neural network with multiple hidden layers of neurons between the input and output layers. A DNN can thus model a complex and nonlinear relationship between input and output features. A DNN is trained by adjusting the weights of the network using backpropagation. In contrast, a GMM models the probability distribution of the output data given the input data using a set of Gaussian distributions, and is typically trained using the expectation maximization (EM) method. MDNs combine the benefits of DNNs and GMMs by using the DNN to model the complex relationship between input and output data, but providing probability distributions as output (see Figure 5).

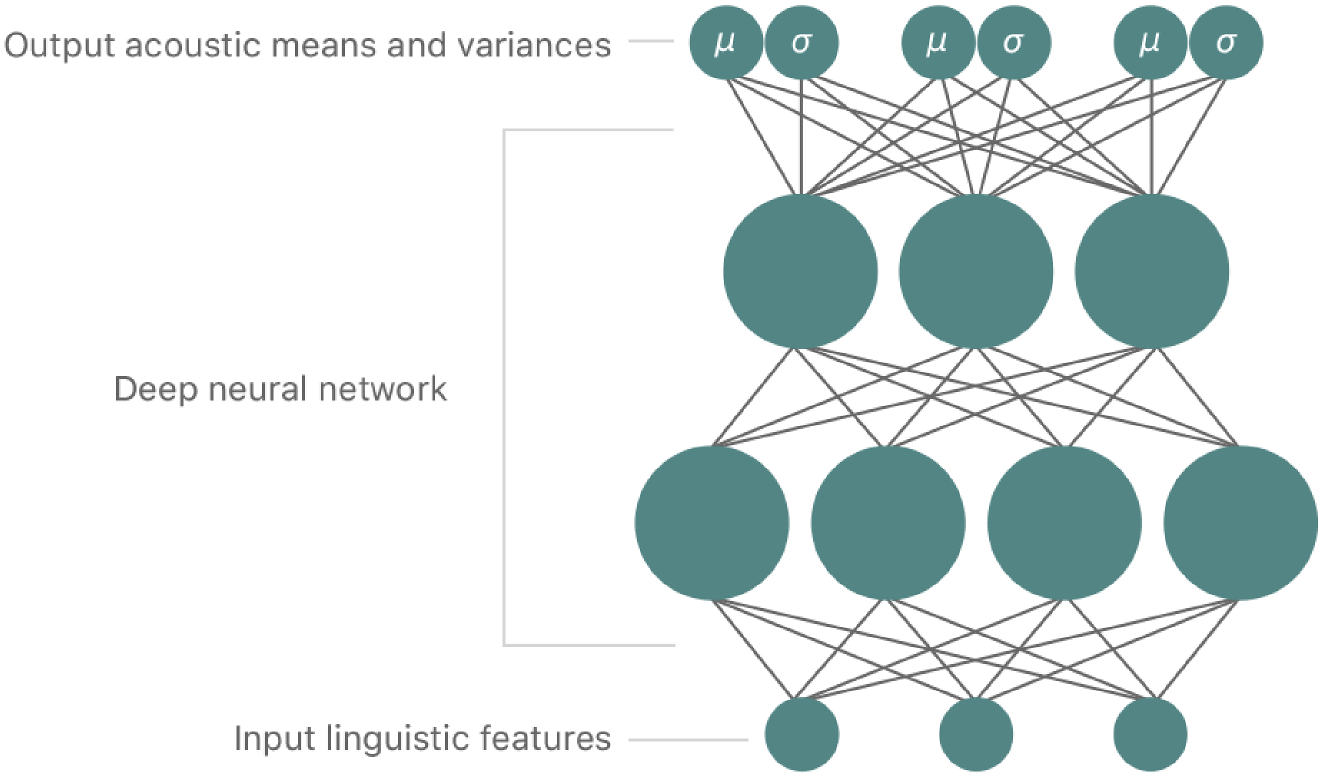
Output acoustic means and variances
μ
σ
μ
σ
μ
σ
Deep neural network
Input linguistic features

**Figure 5. Deep mixture density network used for modeling both the means and variances of the speech features used for guiding unit selection synthesis.**

For Siri, we use an MDN-based unified target and concatenation model that can predict the distributions of both the target features of speech (spectrum, pitch, and duration) and the concatenation cost between the units to guide the unit search. Since the output of the MDN is in the form of Gaussian probability distributions, we can use the likelihood as a cost function for target and concatenation costs:

$$\mathscr{L}(x|\mu,\sigma)=\prod_i \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{\frac{(x_i-\mu_i)^2}{2\sigma_i^2}\}$$

where $x_i$ is the ith target feature, $\mu_i$ is the predicted mean, and $\sigma_i^2$ is the predicted variance. In the actual cost calculation, it is convenient to use the negative log-likelihood and eliminate the constant terms, which leads to the following simple cost function:

where $w_i$ are the feature weights.

The benefit of this approach becomes more clear when we consider the nature of speech. Sometimes the speech features, such as formants, are rather stable and evolve slowly, such as in the case of vowels. Elsewhere, speech can change quite rapidly, such as in transitions between voiced and unvoiced speech sounds. To take this variability into account, the model needs to be able adjust its parameters according to the aforementioned variability. The deep MDN does this using the variances embedded in the model. Since the predicted variances are context-dependent, we can see that they act as automatic context-dependent weights for the costs. This is important for improving the synthesis quality as we want to calculate the target and concatenation costs specific to the current context. The overall cost is a weighted sum of target and concatenation costs:

$$C_{total}(x|\mu,\sigma) = w_t \sum_i w_i \frac{(x_i-\mu_i)^2}{2\sigma_i^2} + w_c \sum_j w_j \frac{(x_j-\mu_j)^2}{2\sigma_j^2}$$

where $w_t$ and $w_c$ are the target and concatenation cost weights. In this final formula, the target cost aims to ensure that prosody is appropriately reproduced in the synthesized speech (intonation and duration), and the concatenation cost ensures both fluent prosody and smooth concatenation.

After scoring the units based on the total cost using the deep MDN, we perform a traditional Viterbi search to find the best path of units. Then, the units are concatenated using the waveform similarity overlap-add (WSOLA) method to find the optimal concatenation time instants to create smooth and continuous synthetic speech.

## Results

We built a deep MDN-based hybrid unit selection TTS system for the new Siri voices. The training speech data contains a minimum of 15 hours of high-quality speech recordings sampled at 48 kHz. We segmented the speech into half-phones using forced alignment, i.e., automatic speech recognition to align the input phone sequence with acoustic features extracted from the speech signal. This segmentation process results in around 1–2 million half-phone units, depending on the amount of recorded speech.

To guide the unit selection process, we trained the unified target and concatenation model using a deep MDN architecture. The input to the deep MDN consists of mostly binary values with some additional continuously-valued features. The features represent information about the quinphone context (2 preceding, current, and 2 succeeding phones); syllable, word, phrase, and sentence level information; and additional prominence and stress features. The output vector consists of the following acoustic features: Mel-frequency cepstral coefficients (MFCCs), delta-MFCCs, fundamental frequency (f0), and delta-f0 (including values both at the beginning and the end of each unit), and the duration of the unit. Since we are using an MDN as an acoustic model, the output also contains the variances of each feature that act as automatic context-dependent weights.

In addition, because the fundamental frequency of speech regions is highly dependent on the utterance as a whole, and in order to create natural and lively prosody in synthesized speech, we employed a recurrent deep MDN to model the f0 features.

The architecture of the trained deep MDN consists of 3 hidden layers with 512 rectified linear units (ReLU) as nonlinearities in each layer. Both input and output features are mean and variance normalized before training. The final unit selection voice consists of the unit database including feature and audio data for each unit, and the trained deep MDN model.

The quality of the new TTS system is superior to the previous Siri system. In an AB pairwise subjective listening tests, listeners clearly preferred the new deep MDN-based voices over the old voices. The results are shown in Figure 6. This better quality can be attributed to multiple improvements in the TTS system, such as deep MDN-based back-end resulting in better selection and concatenation of the units, higher sampling rate (22 kHz vs 48 kHz), and better audio compression.

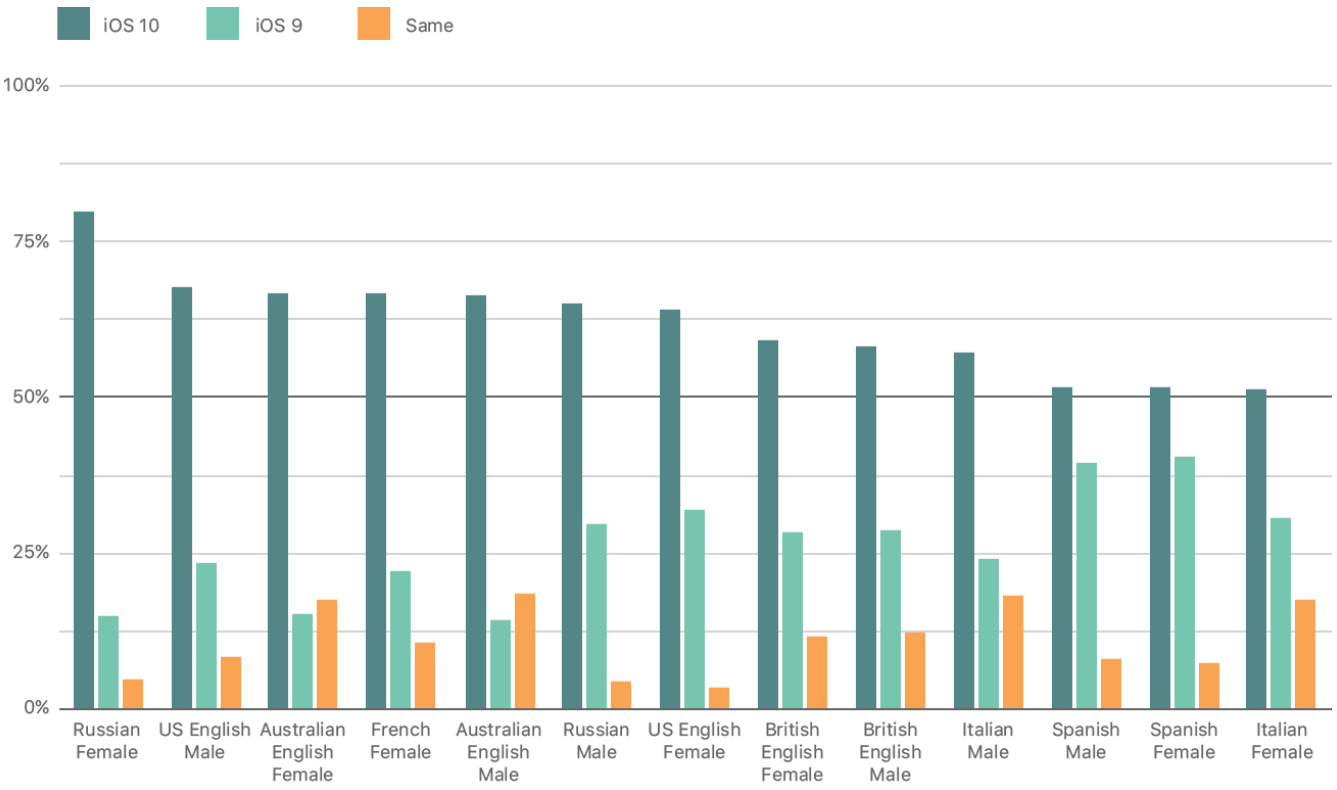

**Figure 6. Results of the AB pairwise subjective listening tests. The new voices were rated clearly better in comparison to the old ones.**

Since the TTS system needs to run on mobile devices, we optimized its runtime performance for speed, memory usage, and footprint by using fast preselection, unit pruning, and parallelization of the calculations.

## A New Voice

For iOS 11, we chose a new female voice talent with the goal of improving the naturalness, personality, and expressivity of Siri's voice. We evaluated hundreds of candidates before choosing the best one. Then, we recorded over 20 hours of speech and built a new TTS voice using the new deep learning based TTS technology. As a result, the new US English Siri voice sounds better than ever. Table 1 contains a few examples of the Siri deep learning -based voices in iOS 11 and 10 compared to a traditional unit selection voice in iOS 9.

## References

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

[2] H. Zen, K. Tokuda, A. W. Black. **Statistical parametric speech synthesis** *Speech Communication*, Vol. 51, no. 11, pp. 1039-1064, 2009.

[3] S. King, **Measuring a decade of progress in Text-to-Speech**, *Loquens*, vol. 1, no. 1, 2006.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, K. Kavukcuoglu. **Wavenet: A generative model for raw audio**, arXiv preprint arXiv:1609.03499, 2016.

[5] Y. Qian, F. K. Soong, Z. J. Yan. **A Unified Trajectory Tiling Approach to High Quality Speech Rendering**, *IEEE Transactions on Audio, Speech, and Language Processingv*, Vol. 21, no. 2, pp. 280-290, Feb. 2013.

[6] X. Gonzalvo, S. Tazari, C. Chan, M. Becker, A. Gutkin, H. Silen, **Recent Advances in Google Real-time HMM-driven Unit Selection Synthesizer**, *Interspeech*, 2016.

[7] C. Bishop. **Mixture density networks**, Tech. Rep. NCRG/94/004, Neural Computing Research Group. Aston University, 1994.

[8] H. Zen, A. Senior. **Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis**, *ICASSP*, 2014.

[9] T. Capes, P. Coles, A. Conkie, L. Golipour, A. Hadjitarkhani, Q. Hu, N. Huddleston, M. Hunt, J. Li, M. Neeracher, K. Prahallad, T. Raitio, R. Rasipuram, G. Townsend, B. Williamson, D. Winarsky, Z. Wu, H. Zhang. Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System, *Interspeech*, 2017.