Interspeech 2018
2-6 September 2018, Hyderabad

Experiments with training corpora for statistical text-to-speech systems

Monika Podsiadło[1] Victor Ungureanu[2]

[1]Google Inc.
[2]Google Switzerland
{mpodsiadlo, ungureanu}@google.com

Abstract

Common text-to-speech (TTS) systems rely on training data for modelling human speech. The quality of this data can range from professional voice actors recording hand-curated sentences in high-quality studio conditions, to found voice data representing arbitrary domains. For years, the unit selection technology dominant in the field required many hours of data that was expensive and time-consuming to collect. With the advancement of statistical methods of waveform generation, there have been experiments with more noisy and often much larger datasets, testing the inherent flexibility of such systems. In this paper we examine the relationship between training data and speech synthesis quality. We then hypothesise that statistical text-to-speech benefits from high acoustic quality corpora with high level of prosodic variation, but that beyond the first few hours of training data we do not observe quality gains. We then describe how we engineered a training dataset containing optimized distribution of features, and how these features were defined. Lastly, we present results from a series of evaluation tests. These confirm our hypothesis and show how a carefully engineered training corpus of a smaller size yields the same speech quality as much larger datasets, particularly for voices that use WaveNet.

1. Introduction

Text-to-speech typically requires a corpus of human voice data as the initial input. For concatenative systems, this corpus is analysed, and at runtime, units deemed optimal are selected to form new utterances. For statistical systems, acoustic models are trained on the corpus. At runtime, the models are used to synthesise new utterances. For both approaches, the runtime speech quality is heavily dependent on the quality of the initial corpus. For concatenative systems, [1], a large, high quality corpus allows for better join and target cost matching. For statistical systems, the cleaner the data, the higher the quality of observations and thus of the synthesized speech. Therefore, the main difficulty in creating new text-to-speech voices is in collecting new speech data.

There have been numerous attempts at solving this data bottleneck. Text-to-speech voices have been successfully trained on crowdsourced speech data [2], on filtered acoustic data for training speech recognition systems [3], or commercial audiobooks [4]. New voices have also been created using voice adaptation techniques [5] or voice morphing [6]. All of these approaches aim to bypass the data sparsity problem and offer an inexpensive way of building new text-to-speech voices.

Our goal was to examine and compare how training datasets of different quality and composition compare when used with the same engine. In effect, it would be the reverse of The Blizzard Challenge [7] [8], a well-established annual experiment where different speech synthesis systems compare their quality when trained on the same corpus. We start by comparing the resulting speech quality for voices trained on high-quality multispeaker data and lower-quality multi-speaker datasets. We then evaluate the performance of voices trained on various subsets of single-speaker high-quality corpus. We design and engineer this corpus to make it balanced with respect to a larger set of features and more relevant to modern TTS usage scenarios. Lastly, we compare the performance of this corpus with corpora of varying sizes developed using other methods.

2. Comparing crowdsourced and high-quality speech corpora

Crowdsourcing has recently been a popular method of collecting speech data that could be used for text-to-speech. The method is particularly useful for low-resourced languages, where high-quality speech corpora are difficult to find or build [9]. It is also an attractive solution for developing new voices when we do not want to invest in professional recording studios or voice talents. It is presumed that single speaker high quality speech corpus will yield superior synthesis quality to a multi-speaker crowdsourced corpus. However, we wanted to conduct an experiment that would quantify the quality difference, which would allow developers to make a more informed decision when selecting their data colletion method.

2.1. Methodology

As a basis for this experiment, we used an existing Google corpus of speech crowdsourced from multiple ordinary speakers of Afrikaans. The corpus contained 2926 sentences (28957 words) recorded in carefully arranged quiet conditions by 10 speakers of the same gender. We then built a second corpus using the same 2926 sentences but recorded in professional studio conditions with a single professional voice actor. The studio recordings took 4 days. We did not put any effort into chosing a particular voice actor aside from a quick technical assessment of their voice in studio conditions to ensure it was suitable for TTS recordings. From these corpora, we built two voices using Googles LSTM speech synthesis system [10]. We did not modify any data inputs other than the audio. For example, the content of the sentences, the pronunciation lexicon, and any text normalization remained identical for both builds. With each voice, we synthesised a test set of 250 sentences. We then conducted an A/B comparison test and a Mean Opinion Score listening test.

2.2. Results

Table 1 below presents the results of the A/B listening test between voices built on a crowdsourced dataset (System A) and a high-quality studio-recorded dataset (System B).

Table 2 below presents the respective Mean Opinion

10.21437/Interspeech.2018-2400

Table 1: *AB preference test results for crowdsourced (System A) and high-quality (System B) voices.*

System A	Neutral	System B
1.6%	16.9%	81.5%

Scores:

Table 2: *MOS scores for voices trained on crowdsourced (System A) and high-quality (System B) datasets*

System A	System B
2.497	3.610

The AB test results unsurprisingly show a strong preference for the voice trained on the single-speaker high-quality corpus. The MOS listening test establishes the precise delta between the scores, with System B scoring 1.14 higher than System A. The score for the voice trained on high-quality corpus also remains relatively high considering the overall size of this dataset (fewer than 3,000 utterances, or just over 2 hours of speech), and the lack of a principled approach to selecting the text for recording.

3. Designing the optimal corpus for statistical text-to-speech

Our next experiment attempted to design a training corpus for text-to-speech that would maximize synthesis quality while maintaining a balanced dataset size. We use a new sentence selection method employing a new set of features to build the initial text corpus. We then perform a series of listening tests using different subsets of this corpus as training data for text-to-speech voices.

3.1. Building the candidate corpus

Typically, when building a text-to-speech corpus, phonological features are used to assess whether the corpus is balanced. For example, the canonical Arctic database [11] ensures that every phoneme pair (diphone) combination possible in English exists in the corpus. Moreso, it ensures that the frequency distribution of specific diphones in the corpus approximates their frequency distribution in natural language. However, modern commercial speech synthesis systems use datasets much larger than Arctic, often containing 40,000 sentences or more [12][13]. In those, simple phonological types like diphones or even triphones are represented adequately. But as the datasets are growing exponentially in size, the quality gains do not follow at the same pace. To build our optimal training corpus, we first conducted a detailed analysis of TTS usage, classifying the requests into one of over 20 domain types (for example, actions, dialog, entertainment, navigation). We then composed and annotated a huge candidate text corpus of over a million sentences that had a similar frequency distribution of these domain types as real TTS usage. This candidate corpus served as input to Skripto, our sentence selection algorithm, which automatically selected specific sentences that would form the recording script.

3.2. Sentence selection algorithm

Sentence selection and the economy of training datasets for text-to-speech has been approached by a number of authors [14] [15] [16] [17].

Our *legacy* sentence selection algorithm filled an initial script with the first sentences of the candidate corpus. It then iteratively went through each remaining sentence of the candidate corpus and replaced the lowest scoring sentence from the script with the current sentence of the candidate corpus, if better. The *legacy* optimization goal was to match the target distribution curves of various phonological features such as: word identifiers, stressed diphones, sonorants, sentence-final syllables. The target distribution curves followed a power law based on the candidate corpus distributions, but it moved some of the mass from the head into the tail for a more balanced resulting script. An internal review of this *legacy* algorithm suggested it is similar to selecting sentences at random from the candidate corpus and, as we will show in section 3.5.2, it is outperformed by the submodular algorithm we describe next.

In this paper, we propose a novel way of selecting a recording script using submodular maximisation [18]. Described in the submodularity framework, the problem becomes a bipartite graph where on the left side we have sentence nodes and on the right side we have nodes representing features of the sentences; edges weigh how much a sentence values the connected feature node [19]. This graph represents our corpus. Given a corpus graph we run the lazy greedy algorithm (with Knapsack constraints) [20] to efficiently find a near-optimal recording script that covers the feature nodes of the original corpus as well as possible.

3.2.1. Features

Previous attempts at solving sentence selection through submodular optimization focused on using simple objective functions: usually using phoneme-only or triphone-only objective functions and only for speech recognition [21] [22] [23].

For defining our objective function, we use the following features, composed of multiple linguistic properties:

V/C + stress Vowels/Consonants with stress. Whether a sound is a vowel or a consonant together with its stress (e.g. unstressed = 0, primary stress = 1). The consonants take their stress from the vowel of the syllable.
Examples: *c0, c1, v0, v1*.

Phonemes A line's individual phonemes.
Examples: *k, a, t*.

Triphones List of three consecutive phonemes.
Examples: *sil-k-a, k-a-t, a-t-sil*.

Word identifiers A line's word identifiers.
Examples: *I, read_past, a, book*.

Trigrams Three concatenated word identifiers.
Examples: *sil-this-is, this-is-a, is-a-cat, a-cat-sil*.

Prosodic types Set of prosodic intonation types for the line.
Examples: *WH_QUESTION, COMMAND*.

In Table 3, we provide a description of how each feature is used as part of our objective function.

3.2.2. Objective function

Given a feature, we define its feature set F as all the distinct values ($items$) it can take. For example, the $phonemes$ feature set (for English) is the set of all 44 English phonemes.

We define $F(l)$ as the set of $items$ from feature set F present in line l, where l is a line from a corpus C.

Table 3: *Features of the objective functions. Each feature item is scored up to "max items" ($count_{max}(F)$ in (5)), a feature is penalized for long sentences if "Penalize" is Yes and a feature ignores <sil> tokens if "Ignore <sil>" is Yes.*

Name	Max items	Penalize	Ignore <sil>
V/C + stress	3000	No	N/A
Phonemes	500	No	Yes
Triphones	1	No	No
Word identifiers	1	Yes	Yes
Trigrams	5	Yes	No
Prosodic types	100	No	N/A

We then define the $counts$ of a line l with regards to feature set F as follows:

$$counts(l \mid F) := \{(item_i, count_i)\} \tag{1}$$

where $item_i \in F(l)$ and $count_i$ represents how many times $item_i$ appears in line l. Note that it is straightforward to extend F and $counts$ to a set of lines S.

We define the discrete derivative of a set function $f : 2^C \rightarrow \mathbb{R}$ with respect to l [19]:

$$\Delta_f(l \mid S) := f(S \cup \{l\}) - f(S), \forall S \subseteq C \text{ and } \forall l \in C \tag{2}$$

where C represents a set of lines (the corpus) and 2^C represents the power set of C. Intuitively, the discrete derivative measures how much value line l would add to the current script S.

Finally, instead of defining our objective function explicitly, we construct the following discretive derivative for our objective function:

$$\Delta_f(l \mid S) := \sum_{F \in features} \frac{1}{count_{l,F}} \Delta_F(l \mid S), \quad \forall S \subseteq C \text{ and } \forall l \in C \tag{3}$$

where:

$$count_{l,F} = \sum_{item_i \in F(l)} count_{l,item_i} \tag{4}$$

$$\Delta_F(l \mid S) := \begin{cases} 0, count_{S,item_i} >= count_{max}(F) \\ \sum_{item_i \in F(l)} \Delta_{item_i}(l \mid S), \text{otherwise} \end{cases} \tag{5}$$

$$\Delta_{item_i}(l \mid S) := \frac{count_{l,item_i}}{count_{l,item_i} + count_{S,item_i}} \tag{6}$$

$$count_{l,item_i} = count_i, (item_i, count_i) \in counts(l \mid F) \tag{7}$$

Similarly, for $count_{S,item_i}$ replace l with S in (7).

It is easy to show that the underlying (unknown) objective function f is monotonically non-decreasing and submodular because its discrete derivative Δ_f is always non-negative and has diminishing returns (the value of a line l decreases as the script S increases).

3.2.3. Lazy greedy algorithm

In order to maximize our (monotone) submodular objective function with Knapsack constraint on the number of words (budget):

$$\max_S f(S) s.t. \sum_{l \in S} c_w(l) \leq B_w \tag{8}$$

where $c_w(l)$ represents the "word count" of line l (i.e. $count_{l,words}$ in (4)) and B_w represents the "word budget", we adapt the lazy greedy algorithm with Knapsack constraints [20] [19] (as we already normalize the discrete derivative in (3)):

$$S_i^{uc} = S_{i-1}^{uc} \cup \left\{ \arg\max_{l \in C \setminus S_{i-1}^{uc}} c_w(l) \Delta(l \mid S_{i-1}^{uc}) \right\} \tag{9}$$

$$S_i^{cb} = S_{i-1}^{cb} \cup \left\{ \arg\max_{l \in C \setminus S_{i-1}^{cb}; c_w(S_i) + c_w(l) \leq B_w} \Delta(l \mid S_{i-1}^{cb}) \right\} \tag{10}$$

$$S_f = \arg\max_{S \in \{S^{uc}, S^{cb}\}} f(S) \tag{11}$$

where S^{uc} is the *uniform-cost* solution, S^{cb} is the *cost-benefit* solution and S_f is the *final* recording script.

3.3. Acoustic data collection

The text corpus selected by Skripto was used to collect speech data. Given the quality advantage of high-quality acoustic data we describe in section 2, the corpus was recorded by a single voice talent in carefully controlled studio conditions. Througout the sessions, we maintained a signal-to-noise ratio of around 50dB and reverberation time below 40 ms. To control for individual speaker effects, we recorded two corpora: one with a female and one with a male voice actor. We wanted to depart from the highly unnatural and controlled delivery typical of earlier generation corpora used for unit selection. We instructed the voice actor to avoid hyperarticultion and to read naturally, using conversational or expressive style as needed. We did not attempt to control the consistency of the recordings, but instead encouraged wide prosodic variation in the reading style of the actor. The actor was allowed to read without interruptions, further contributing to the naturalness of the recordings.

3.4. Dataset size

We created a modular corpus of 100,000 words (90,860 sentences) selected by the Skripto sentence selection algorithm described in section 3.2. Each 20,000-word subpart of the corpus was balanced with respect to the coverage types used for sentence analysis. We then built five voices for each of the two speakers using 20%, 40%, 60%, 80% and 100% of the initial 100,000-word corpus.

3.5. Evaluation

A total of 10 voices were evaluated by means of subjective listening tests collecting Mean Opinion Scores (MOS) for each voice. We used the same set of sentences for all evaluations. The test set contained 1000 sentences and covered a variety of domains requiring a range of prosodic styles: conversational dialog, news sentences, questions, driving directions, device navigation commands, among others. The raters listened to speech samples in quiet conditions using headphones.

3.5.1. MOS Results

Table 4 below presents MOS results for voices trained on balanced, fractional datasets at 20,000-word intervals using Google's LSTM-based speech synthesis system. Voice 1 is the male voice, Voice 2 is the female voice.

We can observe that the male voice scores generally higher than the female voice. We also observe that even the voices with the smallest training dataset achieve good results, and that a five-fold increase in the training data yields only about 0.5 increase in the MOS.

Table 4: *MOS scores for voices built with fractional datasets*

Corpus size in words	Voice 1	Voice 2
$20,000$	3.57	3.28
$40,000$	3.69	3.52
$60,000$	3.72	3.56
$80,000$	3.78	3.63
$100,000$	3.83	3.71

3.5.2. Comparison with other corpora

Table 5 below presents MOS results for voices trained on various legacy American English corpora. These datasets are single speaker corpora recorded by professional voice actors, but developed without the sentence selection algorithm presented in this paper. We quote the corpus size in hours and the MOS achieved by the voices on the same test set. We compare it to a fractional corpus from our experiments described in section 3.5.1 that is closest to the legacy corpus in either size or the MOS result it yields.

Table 5: *Quality comparison between random and carefully designed corpora*

Corpus size in hours and minutes	MOS score	Reference fractional build
$65h\ 37min$	3.92	100% | $3.83\ MOS$
$23h\ 5min$	3.83	100% | $3.83\ MOS$
$4h\ 50min$	3.34	40% | $3.69\ MOS$
$11h\ 25min$	3.41	100% | $3.83\ MOS$
$9h\ 47min$	3.59	100% | $3.83\ MOS$
$10h\ 21min$	3.69	20% | $3.57\ MOS$
$9h\ 12min$	3.32	100% | $3.83\ MOS$
$4h\ 29min$	3.29	40% | $3.69\ MOS$

We see how a 10-hour dataset developed with our method yields similar synthesis quality as datasets 13 hours larger in size, and only slightly lower than a 65-hour dataset. We also see that corpora of similar size yield MOS results betweem 0.3 and 0.7 higher when using our corpus development methodology. In the range of scores quoted here, this distance represents a significant quality difference for the user. We also note that we were able to achieve the same MOS result as a voice trained on a 10-hour legacy corpus using only a fifth of the training data.

3.5.3. Quality with WaveNet

We wanted to investigate whether the quality gains from the proposed corpus design would also apply to voices built using WaveNet [24]. We selected one of the fractional builds for this evaluation, the male voice built with the 80% subset of the corpus (80,000 words or about 7 hours of speech). We then conducted another MOS listening test and compared it with a number of MOS results achieved by WaveNet-backed voices trained on legacy corpora. The legacy corpora are single-speaker databases recorded in studio conditions by carefully vetted professional voice talents, but did not utilize the sentence selection methodology described in section 3.2. Table 6 below shows a comarison of MOS results between WaveNet voices trained on a variety of corpora:

We observe how with only 7 hours of speech data we achieved similar quality on subjective listening tests as with corpora multiple times larger.

Table 6: *MOS result for WaveNet voices and the corresponding corpus sizes*

Training corpus	MOS
$65hours/legacy$	4.21
$23hours/legacy$	4.16
$7hours/Skripto$	4.18

3.6. Discussion

In the results presented above we see how with a more principled recording script design we are able to train voices on much smaller datasets while maintaining similar synthesis quality. While we observe quality increase when comparing datasets of small or medium size, the results are most pronounced when compared with really large datasets of over 20, or even 65 hours of speech. We attribute these results to a wider variety of textual material included in our corpus. We achieved this by utilizing a relatively wide set of domain tags that correlate with prosodic styles, and a more complex method of scoring linguistic features represented by candidate script sentences. In addition, the selected sentenced were recorded in a highly natural, expressive and, when appropriate, conversational style. This, we believe, allowed us to introduce a greater level of variation into the training data. Unlike with unit selection systems, it appears that statistical speech synthesis benefits from this variation in the data, even if observations characterized by a particular set of feature values are only seen a few times in the training set. It is also interesting to note how training on progressively larger datasets did not necessarily yield proportionally better synthesis results. The quality gains were most pronounced with the initial increase from 20,000 to 40,000 recorded words. We also observe how training on very large but random datasets of up to 65 hours does not benefit the quality as compared to smaller but carefully designed datasets. This suggests that developing a smaller corpus using our methodology, depending on resources, between 4 and 10 hours of speech, is a more efficient way of building text-to-speech voices that balances the data effort with synthesis quality.

4. Conclusions

We have presented results from listening tests which we conducted to evaluate speech synthesis quality of voices trained on a number of different corpora. Our experiments examined the relationship between the size and the quality of training datasets, and the resulting text-to-speech quality. We have compared voice quality achieved on multi-speaker crowdsourced training datasets, as well as on single-speaker datasets recorded in professional studio conditions. We have also presented a principled method of building speech corpora that we have shown yields superior text-to-speech quality for voices trained using both LSTM and WaveNet backends. Lastly, for WaveNet voices, we have shown how our method allows to achieve similar Mean Opinion Score results while using only a fraction of the data compared to corpora structured using other methods. Together, our findings suggest the most optimal data collection methods for rapid development of high-quality text-to-speech voices.

5. References

[1] A. W. Black, "Perfect synthesis for all of the people all of the time," in *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*. IEEE, 2002, pp. 167–170.

[2] A. Gutkin, L. Ha, M. Jansche, O. Kjartansson, K. Pipatsrisawat, and R. Sproat, "Building statistical parametric multi-speaker synthesis for bangladeshi bangla," in *SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages, 09-12 May 2016, Yogyakarta, Indonesia; Procedia Computer Science*, 2016, pp. 194–200, edited by Sakriani Sakti, Mirna Adriani, Ayu Purwarianti, Laurent Besacier, Eric Castelli and Pascal Nocera.

[3] E. Cooper, X. Wang, A. Chang, Y. Levitan, and J. Hirschberg, "Utterance selection for optimizing intelligibility of tts voices trained on asr data," in *Interspeech*, 2017.

[4] I. Jauk, A. Bonafonte, P. Lopez-Otero, and L. Docio-Fernandez, "Creating expressive synthetic voices by unsupervised clustering of audiobooks," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] A. Kain and M. Macon, "Text-to-speech voice adaptation from sparse training data," 1998.

[6] Y. Agiomyrgiannakis and Z. Roupakia, "Voice morphing that improves tts quality using an optimal dynamic frequency warping-and-weighting transform," in *ICASSP*, 2016.

[7] S. King and V. Karaiskos, "The Blizzard Challenge 2013," in *Blizzard Challenge Workshop, Barcelona, Spain*, 2013.

[8] R. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the Blizzard Challenge 2007 listening test results," in *Proceedings of Sixth ISCA Workshop on Speech Synthesis, Bonn, Germany*, 2007.

[9] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat, and R. Sproat, "TTS for low resource languages: A Bangla synthesizer," in *10th edition of the Language Resources and Evaluation Conference, 23-28 May 2016*, Portoro, Slovenia, 2016, pp. 2005–2010.

[10] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[11] J. Kominek, A. W. Black, and V. Ver, "CMU Arctic databases for speech synthesis," Tech. Rep., 2003.

[12] 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.

[13] X. Gonzalvo, S. Tazari, C. an Chan, M. Becker, A. Gutkin, and H. Silen, Eds., *Recent Advances in Google Real-time HMM-driven Unit Selection Synthesizer*, Sep 8-12, San Francisco, USA, 2016.

[14] N. Braunschweiler and S. Buchholz, "Automatic sentence selection from speech corpora including diverse speech for improved hmm-tts synthesis quality," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[15] B. Bozkurt, O. Ozturk, and T. Dutoit, "Text design for tts speech corpus building using a modified greedy selection," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[16] W. Zhu, W. Zhang, Q. Shi, F. Chen, H. Li, X. Ma, and L. Shen, "Corpus building for data-driven tts systems," in *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*. IEEE, 2002, pp. 199–202.

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

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[19] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014, pp. 71–104.

[20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 420–429.

[21] Y. Shinohara, "A submodular optimization approach to sentence set selection," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4112–4115.

[22] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Interspeech*, 2009.

[23] ——, "Optimal selection of limited vocabulary speech corpora," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499