blink

By

Quba Michalski

The story so far

Like so many other disciplines, **contemporary text-to-speech (TTS) voice synthesis** is moving toward machine learning. And with enough training data, these systems can learn to imitate not only the sound of a person’s voice, but also their accent, cadence, and even specific mannerisms. All you need to get started is a transcribed recording of someone speaking — the longer, the better — and some processing time on a powerful computer.

Early experiments with deep learning speech synthesis have produced impressive results. However, the process of teaching and using artificial intelligence (AI) still feels hacked together and not quite ready for the end-user. So, until AI is ready for everyday use, we’ll continue to rely on more classic TTS methods, including fully synthesized speech and sentences stitched together from separate words recorded with a human voice.

You've been there

Does this sound familiar? You’re at an airport waiting for an announcement about your delayed flight. Then, a robotic, monotone voice begins spitting out a long string of numbers over the PA system. One. After. Another. No pause. No change in intonation.

By the time you hear the fifth digit, **you've already forgotten the first**.

You fumble for a piece of paper and a pen or try to open a notepad app on your mobile device, and then you brace yourself as you wait for the voice to repeat the sequence. The voice begins listing digits again. **Same speed, same monotony.**

If you're lucky, you manage to write down the list of numbers. If you fumble or hesitate, you'll find yourself heading straight for a help desk to ask the gate agent to repeat the information.

Why does this happen?

In a typical automated PA system, the voice talent provides a sample for each word of the system's phrase. In this case, they would be asked to record numbers "one," "two," "three," all the way to "zero." These recordings (aka samples) are then stored on the PA computer as separate sound bits and reassembled according to the announcement needs. It sounds simple enough, if not for one caveat.

We don't talk like this

When reading a long string of numbers aloud, we follow the same patterns that we use when we say regular sentences.

Let's look at a US phone number, for example. If the number is 5357840213, we first break up the long string into shorter sets (typically 2-5 digits long), such as 535-784-0213.

This process, known as *chunking*, helps us improve short-term retention, which allows the working memory to be more efficient.

Breaking up a long string of numbers into shorter sets, such as the example above, is similar to breaking up a sentence into words. If the long string of numbers is the sentence, then each short set is a word, and the individual digits in each set are the syllables.

Because we treat each digit in the set like a syllable in a word, we pronounce it differently depending on its position in the set. Typically, our voice rises on the first digit and falls on the last while maintaining a relatively flat intonation for the digits in the middle. We also apply a different tone depending on whether the current set of numbers is preceded or followed by another set.

On the other hand, traditional TTS methods record just one audio sample per digit. As a result, each digit in a string of numbers is pronounced in monotone, no matter its position or surroundings.

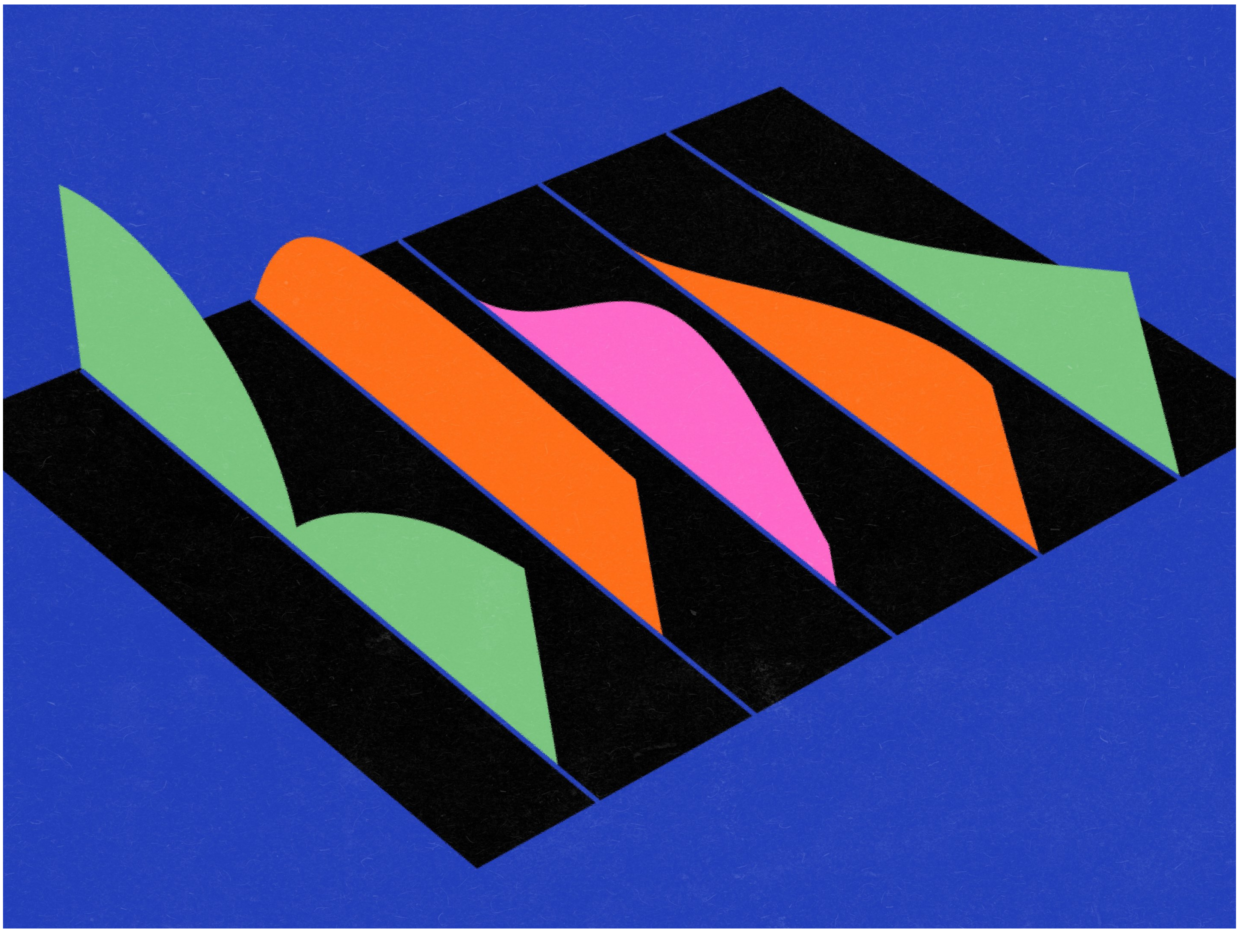

Illustration by Caryn Arredondo.

Instead of reading the phone number aloud with a variety of tones, a traditional PA system would read the text as ten separate words with a full stop after each digit:

Five. Three. Five. Seven. Eight. Four. Zero. Two. One. Three.

The human brain likes patterns and phrases that flow naturally. While each number here is easy to understand on its own, this monotonous fragmentation is what makes PA announcements so difficult to understand.

What is a minor annoyance to some can become an accessibility barrier to others. All users come with different cognitive and physical abilities that may impact their experience with traditional text-to-speech voices. For those who are hard of hearing or who listen to the system in a language other than their native tongue, understanding a tedious, robotic voice can be especially difficult.

When searching for a solution to this problem, we realized that clear communication can support the wide range of our abilities and build a better user experience for all.

The solution:
Natural Number Reader

At Blink, we created a method for recording and playing back numbers that considers both the melody and the cadence of lifelike speech patterns.

Our solution consists of two parts:

- an engine and algorithm that transcribe the text into natural-sounding voice
- a methodology for recording the required audio files using your own voice talent

Throughout the remainder of this article, we'll go over both parts, but first, try the reader below and hear the difference for yourself. Start by using a string of ones to hear the most noticeable difference between standard speech and natural-sounding speech.

[Visit web page at https://blinkux.com/ideas/natural-number-reader to view embedded content]

The difference is quite straightforward, despite the fact that we created this demo without a professional recording session or a trained voice talent. So, while he ended up sounding great, one of our very own Blinkers recorded all samples with a built-in microphone from the comfort of his home office.

Our process for creating numbers, naturally

After years of testing, we concluded that our Natural Number Reader can achieve high-quality playback results **with just five unique voice recordings for each digit:**

A. Opener: The very first digit of the first set of numbers.

B. Middle: Any digit surrounded by other numbers in a set

C. Coma: Last digit of a set, followed by a brief pause and the next set

D. New Set: First digit in any set after the first set

E. Full Stop: Last digit of the entire number

Using our phone number example from earlier, we would represent the number with three sets of digits, following this pattern: ABC-DBC-DBBE.

A zip code may use either one or two sets: AC DBE, or ABBBE.

To add even more variety to our system, we recorded eight versions of each digit, incorporating four different versions of the Middle sample in addition to the Opener, Coma, New Set, and Full Stop. These extra samples allow us to break down the monotony further and create a natural-sounding voice that's easy to understand and remember.

Recording for everyone

If you're working with professional voice talent, you can trust their skill and experience to drive the recording session. However, we designed Blink's Natural Number Reader to produce equally professional-sounding results **even when recording untrained voices.**

We achieve this by following people's natural speech tendencies. Instead of asking the voice talent to perform specific pronunciations, we have them read a sequence of numbers specifically designed to produce the voice modulation we're looking for.

We ask the voice talent to read aloud the following script as if dictating the numbers over the phone. **Numbers should be read slowly with a short pause between each digit and a slightly longer pause in place of the dashes.**

4629-8015
1746-3024
9062-4180
2135-7963
5928-0536
8571-9847
4203-6471
0814-2609
3697-5312
6350-8295
7489-1758
5108-9264

Once recorded, we discard the first and last lines. These are typically pronounced differently, as people tend to put unique emphasis on the first and last sentence of every story.

The remaining ten lines are somewhat similar to a sudoku puzzle, with each digit appearing in each of eight places exactly once. Additionally, the numbers are scrambled to avoid digit repetition (e.g., 111) or ascending and descending sequences (e.g., 456), which, again, we tend to pronounce uniquely, which breaks the pattern.

Illustration by Caryn Arredondo.

The cutting floor

The resulting recording will be about one minute long. So now, all that we have to do is cut the minute-long recording into individual samples and label each sample for use within the algorithm.

Each sample is named with a letter signifying its position in the sequence — **Opener (A), Middle (B), Coma (C), New Set (D), Full Stop (E)** — followed by the digit it represents. So, for example, we would write out the first line of our recording, 1746-3024, as A1.wav, B7.wav, B4.wav, C6.wav, D3.wav, B0.wav, B2.wav E4.wav. The algorithm contains predefined patterns for reassembling the individual samples into full numbers of any length between one and twenty digits.

Tip: If you are reading this article in a Chrome browser (on a desktop or laptop computer), you can open a console window with *ctrl+shift+i* and observe how Natural Number Reader assembles each pattern in real-time through debugging messages.

Creating a solution is on us

Natural Number Reader was born out of frustration. Blink's Director of Innovation, Quba Michalski was tired of missing information due to monotonous PA announcements and kicked off the effort to create a speech technology that offered a solution.

While we've been using the Natural Number Reader on our own projects for a while now, we'd love to see others use it as well. To accomplish this, we're releasing the reader as an open-source project.

You can download Natural Number Reader from GitHub. Try it, analyze it, and use it in your own project.

The algorithm can be easily implemented in any language. We settled on using JavaScript (with the p5js libraries) for the sake of compatibility. The result is lightweight, easily embeddable on any webpage, and works equally well on macOS, Microsoft Windows, Linux, Android, or iOS.

Contact us today

Credits

Natural Numbers Reader is a Blink production.

Concept, algorithm, and implementation: Quba Michalski
Additional programming: Eric Gomez
Visual design: Christen Dute
Illustrations: Caryn Arredondo
Voice talent: Stewart Maclennan

Chief Innovation Officer: Kelly Franznick