# Data Driven Methods in Speech Synthesis

*Geir Gunnarsson*

University of Iceland
geirgu@hi.is
16th January 2005

## *Abstract*

Man has wanted machines to speak for him since the dawn of the computer age. In the beginning people were very optimistic and thought that the computers could easily be programmed to read text and utter speech sounds. The reality has proven to be different. Yet, these daring attempts have given us systems that are quite usable and already taking part in servicing people in their every-day life. These attempts can be split into two categories technologically. On one hand we have solutions that try to mimic the human vocal tract and articulatory system with a digital model. On the other hand we have solutions that use pre-recorded voice examples from a certain language. These recordings are then used to generate a database of sounds needed for building arbitrary sentences in the same language. This paper focuses on the latter, namely the concatenative approach, often referred to as data driven speech synthesis.

## *1. Introduction*

This paper is meant to be a basic overview of the field of speech synthesis, emphasizing the data-driven approach over the formant-synthesis approach. The intended audience is people who do not have extended knowledge on speech synthesis but would like a brief overview of what has been done so far, what is being done know and then perhaps get some ideas of where speech synthesisers could be headed in the future.

Section 2 talks about the first attempts and how the attempts so far have mainly be split into two different approaches: the data-driven approach and the formant-synthesis approach.

Section 3 takes a closer look at data-driven speech synthesis. The main ingredients of the database and the components of a data-driven speech synthesiser are explained.

Section 4 gives an overview of the status and availability of data-driven speech synthesisers today.

Section 5 takes to the skies and depicts some dreams that may or may not come true in the future of data-driven speech synthesis.

This paper is mostly based on my studies and readings in Holmes & Holmes (2001) [1], Dutoit (1997) [2] and Huong, Acero & Hon (2001) [9].

## 2. Overview of Speech Synthesis

Since the 1940's people have been developing and building systems that can speak. In the beginning, work was focused on simulating the vocal system of humans with special made hardware. In the 1970's the vocal tract was simulated digitally and the sounds were generated on the computer's sound card.

These systems were based on measuring the frequencies for the formants in each sound-unit and then generating sound accordingly. These systems are often referred to as *formant synthesis systems*, but also known as *rule-based synthesis systems* and *vocal tract models*.

The main benefit of rule-based systems is that they are universal just as the phonemes of the languages are. I.e., they are independent of the target language. The main faults with these systems is that the produced speech sounds unnatural and various cases that divert from the rules result in errors that disturb the listener and even affect his understanding of what is being said.

A later approach was to use recorded speech, cut it up into units and then glue them together in a fitting order according to the word or sentence being built. The size of the unit varies from being a single phoneme, up through, diphones, triphones, syllables, words, and to entire sentence parts or sentences.

The first approaches of pasting sound-clips together resulted in staggered speech with strange clicks and noises. Intonation and accents were strange and unnatural.

Quality improved with more complicated systems storing data on intonation, pitch and accent along with the different sounds for each different environment of the sound. This gives the system designer more control over the prosody of the generated speech.

## 3. Data Driven Speech Synthesis

Simulating the vocal tract and auditory articulation of humans is complicated. A much simpler approach, at least conceptually, is to record a voice, cut it up into units and then concatenate those units together to form the wanted utterances of speech. This was the idea in the beginning and it is pretty much the same in today's state-of-the-art speech synthesizers but in order to increase their quality the complexity keeps growing. So when the dust settles it is not likely that the data-driven approach is simpler than the formant-synthesis. But, it is hard to compare such fundamentally different approaches. The only real evaluation of a speech synthesizer is how listeners perceive the produced speech.

So, what are the basic components of a general data-driven speech synthesizer? In section 3.1 is an overview of the various data and components needed for a data-driven speech synthesizer.

### 3.1 A Data Driven Speech Synthesizer

The initial phase of building a data-driven speech synthesizer is the analysis stage. A speech corpus is processed in order to find, preferably, all the different sound units of the target language. The designer must decide on the size of the units but can use a combination of several unit sizes. Will the units of the synthesizer be individual words, syllables, half-syllables (demi-syllables), triphones, diphones, etc.?

Another necessary ingredient in such a database is information about duration of each sound in all contexts along with information about pitch and intonation of each context – the prosody of the speech.

Once the database is prepared the actual synthesizer can be formed. The text has to be analysed and mapped to a sequence of sound-units or segments in the database. The prosody has to be mapped on top of that sound sequence and finally the speech waveform can be generated.

### 3.1.1 Tokenising

First thing is to tokenise the text, i.e. break it up into words and punctuation marks. Numbers, acronyms and abbreviations have to be handled specifically. Ambiguity is the biggest problem of everything concerning natural languages. Here, the system must try to resolve the ambiguity of words that are spelled the same but have different meanings. But, luckily the synthesiser only cares if the difference in meaning leads to differences in pronunciation. The double-l words in Icelandic are examples of words where the pronunciation conveys between two totally different meanings of identical words. *Galli* can either be some kind of clothing [gal:I] or a fault [gadl0I].

### 3.1.2 Dictionary lookup

After the text has been broken up into tokens or words, these words are looked up in a dictionary containing phonetic transcription and stress markers.

The lexicon contains the roots of all words that the synthesizer must know. It must also contain all possible suffixes that those roots can take. Having such a dictionary with the pronunciation key for each entry is valuable for the speech synthesizer. During the morphological analysis (see next section), the word in question can be looked up, first as a whole. If the whole word is not found the word is broken up into morphs. Then the system can look for the root and all morphs in the dictionary. Identifying the morphs helps with correct pronunciation of consonant clusters that cross morph-boundaries.

### 3.1.3 Breaking tokens up into morphs

The words must be broken up into morphemes. The root of a word is then used to look it up. The root of the word and the ending of the word will help in determining the stress of each syllable, and the pronunciation.

With existing rules for morphological analysis the system should be able to guess the decomposition of most words into their individual morphs. But those words that do not comply with these rules and can therefore not be de-composed will have to be stored with their suffixes in the dictionary.

### 3.1.4 Analysing the syntax

It is beneficial for dissolving ambiguities to be able to assign part-of-speech tags to the words being synthesized. That can be done with statistical taggers that are trained on a correctly tagged corpus. This analysis of the sentence structure will also help in deciding the prosody of the sentence.

Tagging is expensive but for the purpose of the synthesizer it can be very superficial with only the word category and even only for the content words like the nouns and verbs.

It is essential for trying to guess the correct prosody to decide if the sentence is a question or a statement of fact for instance. Therefore some basic syntax analysis is necessary.

### 3.1.5 Units

On one hand we have the dictionary storing the textual representation of the words that the system must know. On the other hand we have an inventory of sounds that cover all the sounds needed to speak out the words in the dictionary and all sentences containing those words in fluent speech. I.e. the sound inventory must also include sounds that may only exist on word boundaries.

A more primitive approach would be to record all the words needed for such a system and then glue them together to form the sentences. The words have to be carefully recorded by a good speaker speaking clearly. Also there has to be at least two forms of each word, one with an intonation that fits within a sentence or at the start of the sentence and another form with an intonation that fits at the end of the sentence. But still the interplay between words is being ignored. The same word will sound different depending on the words that are next to it in a sentence.

A more sophisticated approach is therefore to build the system with smaller units of sound. We can deduct from phonetics that the letters of the alphabet do not give us a good representation of the sounds in the language. Therefore we must consider the phonemes. But the same applies with the phonemes as with the words in the sense that the environment they are in affects how they sound. A popular solution to that is to build a database of all possible diphones in the target language, like so:

- Make a piece of sound by concatenating the first half of a phoneme to the latter half of every possible phoneme that can stand in front of it.
- Do that for all phonemes in the language.
- Add all first halves of phonemes that can stand after a silence.
- Finally add the latter halves of all phonemes that can stand in front of a silence.

Then you have all the diphones for a language. It follows that the number of possible diphones can at most be the square of the number of phonemes in a language.

Triphones have the first half of a diphone as its beginning and the second half as its end but in between is a whole phoneme. A triphone may be needed for sounds that are affected differently when enclosed by a certain pair of sounds than they would otherwise.

Another unit that may be considered is the demisyllable. A demisyllabe is made by splitting syllables at the centre of the vowels. This is especially good for languages that have a lot of consonant-clusters like Icelandic has [3]. A consonant may behave very differently in the company of other consonants than if it stands alone. Demisyllables are more numerous than diphones so the amount of data will increase but that has less effect as computer-power escalades.

Modern-day speech synthesizers often use a combination of diphones, triphones and demisyllables as their inventory of segments or units.

### 3.1.6 Prosody

In addition to the units, a good database for speech synthesis needs information on duration and timing of sounds, intonation and pitch. This has a collective term as the

prosody of speech. The quality of the prosody of the speech synthesizer has the biggest effect on how users perceive the speech produced by the synthesizer. A wrong prosody may completely alter the meaning of a sentence, e.g. from being a statement to being a question, etc.

The speech in the speech corpus has to be analysed and the various sounds have to be timed in their various environments. These timing figures are then stored and used later in the final stages of the speech-synthesis. A uniform lengthening or shortening can then alter the waveform for the sound in question.

The pitch of a sound is analysed, stored and then used to alter the final waveform in a similar way as the timing. The pitch has the most effect of all the prosody attributes. A rising pitch at the end of a sentence may alter the meaning of a sentence from being a statement of fact to being a question; while, usually speakers will indicate the end of their sentences with a falling pitch.

The prosody of the speech corpus is lost and broken when we have cut the corpus into units and pasted them together into the wanted speech sound. This results in unnatural prosody of the produced sentence. The wanted prosody for the produced speech sound must be guessed and the waveform altered to include the guessed prosody. The problem of guessing the prosody is hard. The various ambiguities of the sentence can cause endless headaches in guessing. This problem falls back to the problem of identifying the meaning of the sentence being produced.

Once the sentence has been analysed it can be converted into waveform. During that conversion prosody information is read from the database along with rules for prosody mappings and applied to the waveform.

### 3.1.7 Speaking

Finally the word boundaries within the waveform have to be glued together by adjusting each word at either end to the word's neighbours so that the transitions from one word to another will be smooth. Now the waveform is ready to be played and hence the synthesizer speaks!

## *4. Data Driven Speech Synthesis Systems of Today*

Almost all state-of-the-art TTS systems available today are concatenation systems. TMA Associates [4] has a web containing an overview of the most common speech synthesizers in use today. One of these products, Softvoice [8], uses format synthesis and you can clearly hear the difference. No clicks in between but a bit harder to understand.

The Austrian Research Institute for Artificial Intelligence [5] is hosting a lot of projects in Natural Language Processing. An interesting example to look at there is a project on Emotional Speech Synthesis [6]. The examples are in German but yet it is easy for a non-German speaker to get the temperament of the speaker in most of the examples.

Mark Huckwale at Department of Phonetics and Linguistics in University College London has written a very interesting paper [10] on data driven speech synthesis.

Huckwale argues that we are far away from building a system that successfully *replicates* human speech because these systems have no sense of intention or the need to communicate. He further argues that no components of the text-to-speech systems operate like a *cognitive model*. His conclusion is thus that what we are left

with is to *simulate* human speech, i.e. to "fake speech that fools more of us more of the time".

The method of the speech production does not really matter. What matters is the quality of the listeners' perception.

## 5. Thoughts for the future

How far can the increased computing-power carry us?

Let's assume we have infinite processing power, memory and disk-space. Would it then be viable to build a speech synthesiser that would build on existing sentences? If we had a recording of every sentence ever uttered within one language, could we then parse the meaning of what the system is supposed to say and then look for a sentence that contains the exact same meaning? Then we would not care about the exact wording of the sentence, but would have to make sure that the meaning fits what we want the system to say. That way the system would have its own style and choice of 'words' (actually of entire sentences) and would sound intelligible.

But these are big assumptions, at least regarding the speech corpus – far less regarding the computing power.

But imagine a limited domain like a news program for a radio station. There exist enormous amounts of recordings of all news ever read in almost any given language. We could digitise some of these recordings for our language of choice. We could make up a form for portraying meaning in news items. The reporters could fill in such forms or an AI-system could parse a story and generate such a form. The form containing the meaning and content of the story would then be used to look up a previous news item that would best fit the meaning. The content words that would not fit the story would be replaced by the content words from the new story. Some filters would have to be applied at positions of concatenations. But those places will be few and far between if the database of existing news items is big enough. As they say, there is nothing new under the sun and yesterday's news can very well also be tomorrow's news. The news has been the same as far back as I can recall. N people killed in country X. N billion krona deficit, etc. When the story has been found and perhaps modified it can be broadcast by simply playing the generated sound file.

The voice may be a problem though. Will we settle on having an array of voices reading the same news program? Can we pre-process the corpus and normalize all the different voices to a single artificial (but nice) voice?

To linguists and phoneticians, this may look like a grim fate for speech synthesizers. Simulating the human endeavour of speaking is much more attractive and much more general. But this approach might be more fitting and more economical for some applications.

This kind of a speech-synthesiser would differ from the ones that are most common today because it cannot be categorized as a text-to-speech synthesizer; it is more like a meaning- or thought-to-speech synthesizer. Actually, after thinking about this I found a mentioning of this in Holmes [1] (pp107-108), and I found such a project being worked on at the Austrian Research Institute for Artificial Intelligence [5] called VIECTOS Vienna Concept-to-Speech system [7].

A similar project is documented in an interesting paper [11], written by Jon R.W. Yi and James R. Glass at the Laboratory for Computer Science at MIT. They have built

a concatenative speech synthesizer, called EINVOICE, for a conversational information retrieval system. They “achieve this in constrained application domains by performing Meaning-to-Speech (MTS) synthesis directly, avoiding a potentially lossy intermediate text re-analysis step.”

This direction of speech synthesizers looks intensely interesting to me as a programmer and science enthusiast who likes to fantasize about futuristic technology with artificial intelligence. And, as I have only just discovered this realm, I have a lot to learn and discover.

## References


[1] Holmes, John, Wendy Holmes, 2001, Speech Synthesis and Recognition, 2nd Edition, *Taylor and Francis*

[2] Dutoit, Thierry, 1997, An Introduction to Text-to-speech Synthesis, *Kluwer Academic Publishers*

[3] Rögnvaldsson, Eiríkur, Rögnvaldur Ólafsson, Þorgeir Sigurðsson, 1999, Tungutækni. Skýrsla starfshóps, *Menntamálaráðuneytið, Reykjavík*, pp 53-54

[4] TMA Associates, http://www.tmaa.com/tts/

[5] Austrian Research Institute for Artificial Intelligence, http://www.ai.univie.ac.at/oefai/

[6] VIECTOS Emotional Speech Synthesis, http://www.ai.univie.ac.at/oefai/nlu/nlu-at-projects.html#EmotionalSpeech

[7] VIECTOS Vienna Concept-to-Speech system, http://www.ai.univie.ac.at/oefai/nlu/viectos/synth.html

[8] Softvoice TTS examples, http://www.tmaa.com/tts/Softvoice_profile.htm

[9] Huang, Xuedong, Alex Acero, Hsiao-Wuen Hon, 2001, Spoken Language Processing, *Prentice Hall PTR*

[10] Huckwale, Mark, Speech Synthesis, Speech Simulation and Speech Science, http://www.phon.ucl.ac.uk/home/mark/papers/icslp02synth.pdf

[11] Yi, Jon R.W., James R. Glass, Natural-Sounding Speech Synthesis Using Variable-Length Units, http://www.sls.csail.mit.edu/sls/publications/1998/icslp98-jonyi.pdf