Learn About Speech

SYNTHESIS

sas
Specialty Answering Service

WWW.SPECIALTYANSWERINGSERVICE.NET

Contents

1 Introduction

The word 'Synthesis' is defined by the Webster's Dictionary as 'the putting together of parts or elements so as to form a whole'. Speech synthesis generally refers to the artificial generation of human voice – either in the form of speech or in other forms such as a song. The computer system used for speech synthesis is known as a speech synthesizer. There are several types of speech synthesizers (both hardware based and software based) with different underlying technologies. For example, TTS (Text to Speech) system converts normal language text into human speech, while there are other systems that can convert phonetic transcriptions into speech. The basic principle behind speech synthesis is known as the source-filter theory of speech production – that is, the information about the voice source and the vocal tract filter can be separated from each other.

Today, speech synthesizers are a common feature in most operating systems. Speech synthesis applications have also made computing related services more inclusive by allowing access to people with visual impairments or reading disabilities.

1.1 Quality of a Speech Synthesizer

The quality of a speech synthesizer is measured based on two primary factors – its similarity to normal human speech (naturalness) and its intelligibility (ease of understanding by the listener). Ideally, a speech synthesizer should be both natural and intelligible, and speech synthesis systems always attempt to maximize both characteristics.

1.2 The TTS System

A typical text to speech system has two parts – a front end and a back end. The front end is responsible for text normalization (the pre-processing part) and text to phoneme conversion. Text normalization or tokenization is the phase where numbers and abbreviations in the raw text are converted into written words. Text to phoneme conversion or grapheme-to-phoneme conversion is the process of assigning phonetic transcriptions to each word and dividing them into prosodic units such as phrases, clauses, and sentences. The output of the front-end system is the symbolic linguistic representation of the text. It is composed of the phonetic transcriptions along with the prosody information. This output is then passed on to the back-end system or the synthesizer, which converts it into sound. Sophisticated synthesizers would also have the option to compute the target prosody – the pitch contours and phoneme durations – which can then be applied to the output speech to improve the human nature of the speech output.

2 History

Early speech synthesis efforts revolved around attempts to create machines that could generate human speech, much before electronic signal processing was invented. Over the years, an ever-increasing understanding of the mechanism of voice has led to significant developments in the field of speech synthesis.

One of the earliest successful attempts to reproduce human speech was in 1779, when the Danish scientist Christian Kratzenstein, developed models of the human vocal tract that could produce the vowel sounds in the English language 'a, e, i, o, u' using various shaped tubes. The 'acoustic-mechanical speech machine' developed by Wolfgang von Kempelen in 1791 was an improvement over this and managed to produce consonants as well as vowels.

One of the earliest manually operated electrical apparatuses for speech synthesis was constructed by J.Q. Stewart in 1922. In 1924, a similar machine was also demonstrated by Dr. Harvey Flecher of Bell Labs when he managed to produce a limited vocabulary of sounds, including vowels and words such as 'mamma' and 'papa'.

The Voder was the first major attempt to encode speech electronically. Work on the Voder started as early as October 1928 at the AT&T Bell Laboratories. The Voder was based on a source-filter model for speech that included a non-parametric spectral model of the vocal tract produced by the output of a fixed bandpass-filter bank. In 1948, Werner Meyer-Eppler recognized the capability of the Voder machine to generate electronic music, as described in Dudley's patent. In the 1950s, Munson and Montgomery attempted a parametric spectral model for speech synthesis.

Simultaneously Dr. Franklin S. Cooper and his colleagues at Haskins Laboratories developed the Pattern playback device, which converted a spectrogram plotting of the acoustic patterns into artificial speech.

In the 80s and 90s, there was a lot of research on speech synthesis at MIT and Bell Labs. The Bell Labs system was the first language-independent system that made use of natural language processing techniques for speech synthesis.

2.1 Electronic Devices

The earliest computer based speech synthesis systems were created in the late 50s. Noriko Umeda et al. developed the first general purpose English TTS system in 1968 at the Electrotechnical Laboratory, Japan. In 1961, physicist John Larry Kelly, Jr. and colleague Louis Gerstman used an IBM 704 computer to synthesize speech, which was a prominent event in the history of speech related research done at Bell Labs. Despite the success of purely electronic speech synthesis, research is still being conducted into mechanical speech synthesizers.

Handheld devices featuring speech synthesis emerged in the 70s. Early attempts towards using speech synthesis in devices were guided by the need for inclusive design and the Telesensory Systems Inc. (TSI) Speech+ portable calculator for the blind. Developed in 1976, it was probably the first device to have embedded speech synthesis. Other early uses for embedded speech synthesis included educational toys such as Speak & Spell, produced by Texas Instruments in 1978, or games such as the speaking version of Fidelity's electronic chess computer in 1979. The first video game to feature speech synthesis was the 1980 arcade game, Stratovox from Sun Electronics, followed by the arcade version of Berzerk. The first multi-player electronic game using voice synthesis was Milton from Milton Bradley Company, which was released the same year.

In the early stages, electronic speech synthesizers produced robotic sounds that were hardly intelligible. Today, there has been a lot of improvement in speech synthesis systems, but a discerning listener can still distinguish the output of a speech synthesizer from actual human speech.

As hardware and storage costs become lower and computational power of devices increases, speech synthesizers will play a key role in inclusive design, thus making computing applications accessible to a larger population of people.

3 Synthesizer Technologies

Synthesized speech can be created by joining together speech elements that are stored in a database. These speech elements can be as granular as phones and diaphones, or as huge as entire sentences. The level of output range and clarity of a speech synthesis system are inversely proportional to each other and are dependent on the size of the stored speech elements. Typically, a speech synthesizer that makes use of stored phonetic elements can give a large output range, but with less clarity. Speech synthesizers used for specific domains such as a medical call center or a legal firm, are more likely to use an entire vocabulary of specific words and sentences to make the output quality better. However, the number of sentences that the speech synthesizer could produce would be limited to the extent of its stored vocabulary. On the other end of the spectrum, there are also true voice synthesizers that can create artificial speech from scratch using a model of the vocal tract and reproducing the characteristics of the human voice.

Synthesizer technologies used to generate artificial speech can be broadly classified into two groups – concatenative synthesis and formant synthesis. Both of these technologies have their advantages and disadvantages, and the choice of technology in any speech synthesis system is usually based on the intended end use.

3.1 Waveform/Spectral Coding

Waveform coding refers to the modification of a sound wave by analyzing its characteristics, compressing it, and then restoring it back to its original form. This type of synthesis works only on the acoustic waveform and its spectral characteristics, and is not concerned with the physical speech production system. It can be applied in areas such as telephonic transmission. Human speech is analyzed and compressed into packets that are then transmitted and decoded at the receiving end to synthesize speech for the listener. Other applications of this form of speech synthesis include voice transformation (e.g. alteration of the pitch or duration of the voice) and for creating speech databases for concatenative synthesis. Some popular waveform coding techniques include LPC (linear predictive coding) and PSOLA (pitch synchronous overlap approach).

3.2 Concatenative Synthesis

Concatenative synthesis, as the name suggests, is based on the concatenation (or stringing together) of pre-recorded speech segments. As a result, this can produce the most natural sounding artificial speech. However, the automated technique used for segmenting the waveforms generated can affect the successful reproduction of variations in speech and inflection that is a part of normal speech. This leads to audible glitches in the output. There are three main sub-types of concatenative synthesis.

3.2.1 Unit Selection Synthesis

Unit selection synthesis is done using speech databases, which are large storehouses of recorded speech. The database is created to store speech units – each recorded utterance is segmented into individual phones, diaphones, half-phones, syllables, morphemes, words, phrases, or sentences with the help of a customized speech recognizer. Typically, a manual correction of the segments is performed based on the visual representation of the recorded speech, such as the waveform and spectrogram. Each speech segment also includes other acoustic parameters such as pitch, duration, position in the syllable and the neighboring phones. During the creation phase, the target utterance is recreated using a weighted decision tree algorithm that selects the best possible candidate units from the database and strings them together.

Creating artificial speech using unit selection provides the maximum amount of naturalness to the output since it applies only a minimal amount of digital signal processing (DSP) to the human speech that is recorded. The higher the amount of DSP, the less natural the speech will be. However, most systems will use at least a minimal amount of signal processing, especially to smooth out the waveform at points where the speech units are concatenated to each other. High-end unit-selection systems are so good that the output from these systems is often not any different from real human voices. This is especially true for Text to speech systems, which have been specifically created and fine-tuned to handle pre-defined scenarios. However, the flip side is that in order to produce the best output possible, unit selection systems need huge speech databases often running into gigabytes of recorded data. It is also time consuming to create these databases as they often represent several hours of recorded speech. The other negative aspect of these systems is the robustness of the selection algorithm, which often selects a less than clear unit, even if there are better choices available in the database. Newer systems rely on automated methods to identify segments that are unnatural in the speech synthesis systems so as to introduce a correction factor.

3.2.2 Diaphone Synthesis

This method uses a comparatively smaller database than unit selection systems. The database consists of all sound-to-sound transitions or diaphones in the language being modeled. The number of diaphones and consequently the size of the database varies based on the phonotactics of the language choice – Spanish for example has only around 800 diaphones whereas German has three times the number – around 2500. One sample of each diaphone is stored in the database and the run-time speech generation is done by superimposing the target prosody of the sentence of these diaphone units with the help of DSP techniques such as PSOLA, MBROLA or linear predictive coding.

The negatives of this technique include the sonic glitches caused by concatenation of very small units, and voice output that is generally more robotic than natural sounding. As storage prices drop, the significant advantage of this technique of low database size is no longer a differentiator for commercial applications and hence it is mostly used in free speech synthesis software and not much for commercial purposes.

3.2.3 Domain-Specific Synthesis

Domain specific synthesis is used for specialized domains where the output vocabulary is limited – such as in a medical call center, a weather report, or a railway announcement. Here prerecorded words and phrases are combined together to create specific sentences. This is one of the easiest technologies to implement in the area of speech synthesis and as a result has found several commercial uses including in devices such as calculators and clocks. The biggest advantage of these systems is the high level of naturalness due to the large unit size, the limited vocabulary needed, and the small set of contexts, resulting in easy matching of prosody and intonation.

On the other hand, they cannot be used as a general-purpose speech synthesizer due to the limited vocabulary stored in their databases. The other problem with this technique is the context sensitive pronunciation differences of the same word, which are difficult to reproduce using this technique without additional logic being built-in. For example, in French many final consonants that are usually silent in words are pronounced when they are followed by a word that begins with a vowel. In English as well, the ‘r’ sound is only pronounced when the following word has a vowel as its first letter.

3.3 Formant Synthesis

Formant synthesis is yet another technique of speech synthesis that is not based on a physical model of the vocal folds and vocal tract. The formant synthesizer takes voice source parameters such as the fundamental frequency and amplitude of the voiced sound and attempts to reproduce them. The effects of the vocal tract are specified as formant frequencies and their respective bandwidths. Thus, the formant synthesizer attempts to replicate voiced sound by combining the sound sources and filters and then shapes them into vowels and consonants by applying the vocal tract resonances over the voiced sound.

Unlike concatenative synthesis, formant synthesis techniques do not use human speech units for creating speech. Instead, the speech output is generated using additive synthesis and an acoustic model. Artificial speech waveforms are generated by varying speech parameters such as fundamental frequency, voicing, and noise levels over time based on predefined rules. As a result, this technique is also referred to as rules-based synthesis. Most systems based on this technique generate artificial robotic sounding speech and hence naturalness is very low for this kind of speech synthesis. Despite this, formant synthesis does have several advantages over concatenative systems.

In a formant synthesizer the user can control all acoustic characteristics observed in natural speech such as the pitch, formants etc. by plotting and editing them in a spectrographic display.

The key advantage is that it is more intelligible than concatenative systems especially at high speeds, thus making it useful for applications such as screen readers for visually impaired people, where the purpose is not to replace the human speech. The other big advantage is the limited storage space requirements, as no speech sample database is used. It is an ideal choice for speech synthesis in embedded systems, where the limited memory and microprocessor power have to be optimized. It is also used for experiments to study the perceptual relevance of acoustic properties. These systems are also much more flexible as all aspects of output speech can be controlled using a rule base, thereby allowing various speech outputs. Systems can easily model a wide variety of prosodies and intonations, making it easy to reproduce statements and questions and also to introduce emotions and different voice tones.

In the late 1970s, formant synthesis was used in Speak & Spell, a Texas Instruments gadget where a very high level of intonation control was achieved. It is also used in gaming and entertainment systems such as the Sega arcade machines and in many Atari, Inc. arcade games using the TMS5220 LPC Chips. Formant synthesis is also used in TTS applications such as DecTalk.

3.4 Articulatory Synthesis

Articulatory synthesis systems use computational techniques for synthesizing speech based on models of the human vocal tract and the articulation processes occurring there. Articulatory synthesis systems are mostly used for laboratory experiments. The first such synthesizer known as ASY was developed in the 70s at Haskins Laboratories by Philip Rubin, Tom Baer, and Paul Mermelstein. It was based on vocal tract models developed earlier at Bell Laboratories by Paul Mermelstein, and Cecil Coker.

Although articulatory synthesis models are typically not used in commercial applications, a notable exception is the NeXT-based system. Marketed in the 90s, it provided articulatory synthesis-based text to speech conversion using a transmission line model of the human oral and nasal tracts.

Articulatory synthesis systems are based on an attempt to recreate the physiology of the speech production system – thus the synthesizer gives the user control over the positions, movements and other characteristics of speech producing organs such as the tongue, jaw, lips, vocal chords and even the respiratory system. There are several subsystems under this and they vary drastically in terms of the synthesizer model as well as the practical applications.

Research in this field is also diverse with some researchers focusing on specific subsystems of the speech system – attempting to create complex mathematical models of specific organs such as the tongue or the jaw. In such research, the focus is not on re-creation of speech, but on the mechanical movement patterns of the speech organs. Other researchers focus on creating a simple mathematical model of all the articulators – attempting to understand how articulation affects various acoustic properties. Yet another method of synthesis involves directly specifying the vocal tract shape, avoiding the need to describe the actual articulators.

Theoretically, the complex subsystem models can be combined to form a comprehensive model of the speech production system, which can simulate the articulator movement as well as resulting acoustics. This 'virtual human speaker' would also be a great research tool that could help to test many theories about speech production and understanding.

Researchers believe that articulatory synthesis would someday be able to create the most natural sounding synthetic speech as it can incorporate all the physical aspects of a speech production system. The key roadblock to obtaining natural sounding speech using this technique is the lack of understanding on the timing of the articulators and how they coordinate with each other during real life speech production. As a result of this timing problem, speech produced by the articulatory synthesizers is often unintelligible and unnatural.

3.5 HMM-Based Synthesis

Hidden Markov Models (HMMs) based synthesis use statistical parametric techniques to generate the speech output. The frequency spectrum, the fundamental frequency, and duration of speech are modeled simultaneously by HMMs to capture the characteristics of the vocal tract, the vocal source, and the prosody of human speech. Using a maximum likelihood algorithm, speech waveforms are directly generated from the HMMs.

3.6 Sine Wave Synthesis

In this model, speech synthesis is done by replacing the main energy bands or formants with pure tone whistles.

4 Challenges

There are several challenges faced by speech synthesis techniques at various stages of the process. Some of these are enumerated below:

4.1 Text Normalization Challenges

4.1.1 Homographs

Texts are typically full of numbers, heteronyms, and abbreviations that all need to be expanded into a phonetic representation. At the text normalization stage, a key challenge is posed by words that have the same spelling but different pronunciations based on the context. Generating semantic representations of input texts are computationally ineffective and the results are often unreliable. Hence, most text-to-speech systems rely on heuristic techniques to identify the right pronunciation for homographs based on the neighboring words, with the help of statistics such as frequency of occurrence, or Hidden Markov Models (HMM). HMMs generate 'parts of speech' to help disambiguate homographs and they have been quite successful in many cases with an error rate of less than five percent. For example, they can help to decipher whether 'read' appears in the present tense or past tense in a sentence, and thus vary the pronunciation between 'reed' and 'red'. Hidden Markov Models can be effectively used with most European languages, but most may not have well-defined training corpora.

4.1.2 Numbers and Abbreviations

Converting numbers to speech is another challenge that TTS systems face. Although converting a number to words is a straightforward programming problem, the challenge lies in identifying the context sensitive speech requirements for a number. For example, 1234 can be read as "one two three four" or "twelve thirty four" or "one thousand two hundred and thirty four" depending on the context. Romal numerals also face the same challenge – Edward VII is read as 'Edward the Seventh' while Chapter VII is read as 'Chapter Seven'.

TTS systems often identify how to expand a number based on the neighboring words and numbers as well as the punctuation used in the sentence.

Abbreviations can also be challenging. The abbreviation 'in' for inches has to be distinguished from the word 'in'. Similarly, 'St' can be expanded as 'Street' or 'Saint' depending on the context. Some TTS systems with built-in artificial intelligence make educated guesses about ambiguous abbreviations at the front end, while others create nonsensical outputs that are out of context.

4.2 Text-to-Phoneme Challenges

Text-to-phoneme conversion is the process of arriving at the pronunciation of a word based on its spelling. There are two basic techniques for text-to-phoneme conversion in speech synthesis systems – dictionary -based approach and rule-based approach. In the dictionary-based approach, a large database containing all the words and their pronunciations is stored in the program and the text-to-phoneme conversion is performed by a simple program that looks up each word in the database and replaces the text with the pronunciation. The rule-based approach makes use of phonetic rules, which are applied to words to arrive at their pronunciation. This is similar to the phonetic reading approach used to teach reading to kids.

Both of these approaches have pros and cons. The dictionary-based approach is faster and accurate for words in the vocabulary; however, if a word is not present in the database, it will fail. The other negative is the huge memory size required for the dictionary database. Rule-based systems are generally smaller in size and have the advantage of being able to work with any input. However, the complexity of the rules can make the system programmatically very complex, especially when you have to handle irregular pronunciations and spellings.

Most commercial speech synthesis systems use a combination of the dictionary-based and rule-based approaches to incorporate the best of both worlds. The choice of technique also depends on the type of language being modeled. Languages with a phonemic orthography (where their pronunciations closely match their spellings) render themselves well to rule based method and speech synthesis systems, for such languages use rule-based methods extensively with just an incremental dictionary being used for foreign names and other derived words where the pronunciations may differ significantly from their spellings. English language however is very different with irregular spelling systems and speech synthesis systems. English systems rely more on dictionaries and use rule-based engines for handling exceptions, and for words that are not present in the dictionary database being used.

4.3 Evaluation Challenges

Challenges also exist in evaluating speech synthesizers and it is not always easy to say that one system is better than the other. This is primarily because there are no universally agreed evaluation criteria for speech synthesizer performance. In addition, different applications use largely different vocabularies and the quality of the speech synthesis output depends on the listener. Equipment used to record and reproduce speech will also affect the quality of the output generated. Today however, most speech synthesis systems are evaluated using a common dataset.

5 Speech Synthesis in Operating Systems

5.1 Atari

The first integration of speech synthesis systems with operating systems was done in the 1400XL/1450XL personal computers designed by Atari. These machines used a Finite State Machine to enable text-to-speech synthesis of the English language. However, these were not a commercial success.

5.2 Apple

Apple came out with the first commercially successful operating system with an integrated speech synthesis system – MackInTalk software. The speech synthesis software was licensed from Joseph Katz and Mark Barton and featured in Mac computers since the 80s. By the early 90s, Apple provided system-wide TTS support. As the PC revolution emerged, Apple started including high quality voice sampling and speech recognition into its systems thus providing full capabilities around speech. Although it started as a curiosity, the speech system of Apple Macintosh has today evolved into a complete program in its own right – PlainTalk. Users even have the ability to select from a wide range of voices.

The Apple iOS operating system used on Apple devices such as the iPhone, iPad and iPod provides accessibility enhancement using VoiceOver speech synthesis. Today many third party apps on the iPad and iPhone also provide speech synthesis to enable webpage browsing or translating text from a foreign language.

5.3 AmigaOS

The AmigaOS, introduced in 1985, also had advanced speech synthesis capabilities made available with the help of the Amiga hardware audio chipset. The speech synthesis software for this OS was also licensed to SoftVoice, Inc., the developers of the original MackInTalk text to speech system. It had advanced features such as male and female voices and 'stress' indicator markers. The speech synthesis system was divided into a narrator device and a translator library. The AmigaOS designed speech synthesis as a virtual hardware device, so the user could even redirect console output to it.

5.4 Microsoft Windows

In Windows, speech synthesis and speech recognition are supported by using SAPI 4 and SAPI 5 components. Windows 2000 included Narrator, a text to speech utility for the visually impaired. Windows does not provide system-wide text to speech capabilities. Some programs can use speech synthesis directly while others rely on plug-ins, extensions, or add-ons to read out text

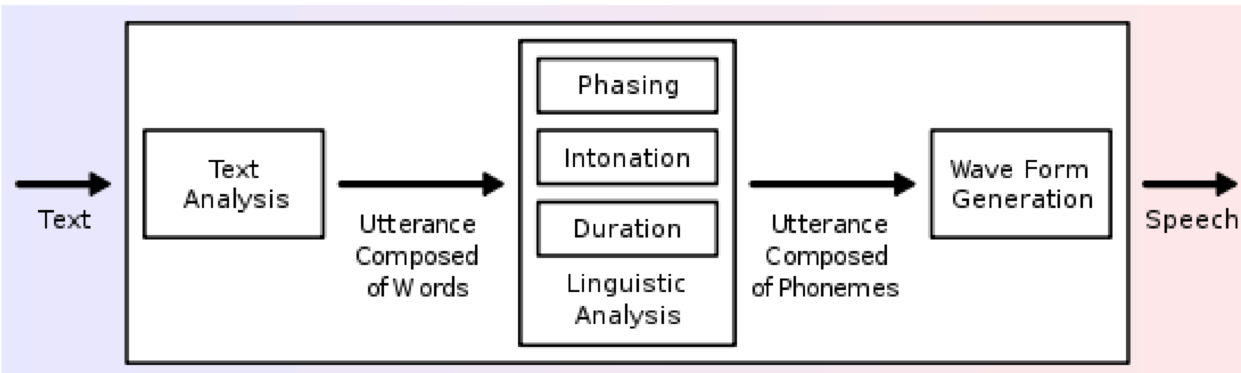
Microsoft Speech Server is a server based voice package with speech synthesis and speech recognition capabilities. It is aimed for network use with web applications and for use in call centers.

Today, there are a number of applications, browser plug-ins and gadgets such as e-book readers (Amazon Kindle, Samsung E6), GPS navigation systems (Garmin, TomTom) and even mobile phones (iPhone, Samsung Galaxy) that incorporate speech synthesis. Online RSS narrators can narrate RSS feeds, enabling users to listen to their RSS feeds while on the move.

Several non-profit projects revolving around accessibility have speech synthesis incorporated within their offerings. The Pediaphon project created in 2006 provides a web-based TTS interface to Wikipedia. Other web-based assistive technology products include Browsealoud and Readspeaker, both of which deliver TTS functionality over the web.

6 Speech Synthesis Markup Languages

Several markup languages can be used for speech synthesis. These XML-compliant formats include Speech Synthesis Markup Language (SSML), the W3C recommendation in 2004, as well as older markup languages such as JSML Java Speech Markup Language and SABLE. Although each of the markup languages was proposed as a standard when they were initially introduced, none of these have found widespread acceptability. Speech synthesis markup languages are different from dialogue markup languages such as VoiceXML, which include tags for speech recognition, dialogue management, and touch tone dialing.



7 Applications

Speech synthesis finds applications in three broad areas – contact centers, assistive technologies for people with disabilities, and gaming.

7.1 Contact Centers

Speech synthesis combined with speech recognition has also improved the interaction between the user and mobile devices with the help of natural language processing interfaces.

7.2 Assistive Technologies

Speech synthesis solutions have played a vital role in assistive technologies by removing environmental barriers for people with different disabilities. One of the most common applications is the use of speech synthesis technology in screen readers for people with visual disabilities. Speech synthesis based tools also find applications in the area of helping people with learning disabilities such as dyslexia. They are also used to engage pre-literate children in multimedia applications. Solutions for speech-impaired people also include a speech synthesis module combined with dedicated voice output hardware. Text-to-speech solutions for disabled people also find applications in public transport systems and in other areas, integrating them into daily life.

7.3 Gaming and Entertainment

Speech synthesis techniques are also extensively used in games and animation movies. Key software that has been developed specifically for the entertainment industry was the software application package created in 2007 by Animo Limited. It is based on their speech synthesis software, FineSpeech. This application package is specifically geared towards use in the entertainment industry with capabilities to generate narration and dialogues based on specific user inputs.

8 References

1. http://en.wikipedia.org/wiki/Speech_synthesis

2. http://sal.shs.arizona.edu/~bstory/synthesis