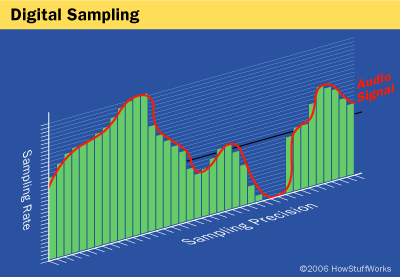
1. Today, when we call most large companies, a person doesn't usually answer the phone. Instead, an automated voice recording answers and instructs you to press buttons to move through option menus. Many companies have moved beyond requiring you to press buttons, though. Often you can just speak certain words (again, as instructed by a recording) to get what you need. The system that makes this possible is a type of speech recognition program -- an automated phone system.
2. You an also use speech recognition software in homes and businesses. A range of software products allows users to dictate to their computer and have their words converted to text in a word processing or [e-mail](https://computer.howstuffworks.com/e-mail-messaging/email.htm) document. You can access function commands, such as opening files and accessing menus, with voice instructions. Some programs are for specific business settings, such as medical or legal transcription.
3. People with disabilities that prevent them from typing have also adopted speech-recognition systems. If a user has lost the use of his hands, or for visually impaired users when it is not possible or convenient to use a Braille keyboard, the systems allow personal expression through dictation as well as control of many computer tasks. Some programs save users' speech data after every session, allowing people with progressive speech deterioriation to continue to dictate to their computers.
4. Current programs fall into two categories:

**Small-vocabulary/many-users**

1. These systems are ideal for automated telephone answering. The users can speak with a great deal of variation in accent and speech patterns, and the system will still understand them most of the time. However, usage is limited to a small number of predetermined commands and inputs, such as basic menu options or numbers.

**Large-vocabulary/limited-users**

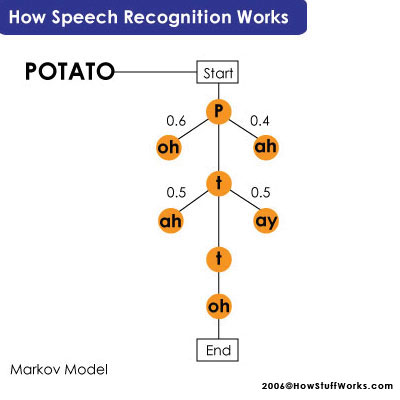
1. These systems work best in a business environment where a small number of users will work with the program. While these systems work with a good degree of accuracy (85 percent or higher with an expert user) and have vocabularies in the tens of thousands, you must train them to work best with a small number of primary users. The accuracy rate will fall drastically with any other user.
2. Speech recognition systems made more than 10 years ago also faced a choice between discrete and continuous speech. It is much easier for the program to understand words when we speak them separately, with a distinct pause between each one. However, most users prefer to speak in a normal, conversational speed. Almost all modern systems are capable of understanding continuous speech.
3. To convert speech to on-screen text or a computer command, a computer has to go through several complex steps. When you speak, you create vibrations in the air. The analog-to-digital converter (ADC) translates this analog wave into digital data that the computer can understand. To do this, it samples, or digitizes, the sound by taking precise measurements of the wave at frequent intervals. The system filters the digitized sound to remove unwanted noise, and sometimes to separate it into different bands of frequency(frequency is the wavelength of the sound waves, heard by humans as differences in pitch). It also normalizes the sound, or adjusts it to a constant volume level. It may also have to be temporally aligned. People don't always speak at the same speed, so the sound must be adjusted to match the speed of the template sound samples already stored in the system's memory.



1. Next the signal is divided into small segments as short as a few hundredths of a second, or even thousandths in the case of plosive consonant sounds -- consonant stops produced by obstructing airflow in the vocal tract -- like "p" or "t." The program then matches these segments to known phonemes in the appropriate language. A phoneme is the smallest element of a language -- a representation of the sounds we make and put together to form meaningful expressions. There are roughly 40 phonemes in the English language (different linguists have different opinions on the exact number), while other languages have more or fewer phonemes.
2. The next step seems simple, but it is actually the most difficult to accomplish and is the focus of most speech recognition research. The program examines phonemes in the context of the other phonemes around them. It runs the contextual phoneme plot through a complex statistical model and compares them to a large library of known words, phrases and sentences. The program then determines what the user was probably saying and either outputs it as text or issues a computer command.
3. Early speech recognition systems tried to apply a set of grammatical and syntactical rules to speech. If the words spoken fit into a certain set of rules, the program could determine what the words were. However, human language has numerous exceptions to its own rules, even when it's spoken consistently. Accents, dialects and mannerisms can vastly change the way certain words or phrases are spoken. Imagine someone from Boston saying the word "barn." He wouldn't pronounce the "r" at all, and the word comes out rhyming with "John." Or consider the sentence, "I'm going to see the ocean." Most people don't enunciate their words very carefully. The result might come out as "I'm goin' da see tha ocean." They run several of the words together with no noticeable break, such as "I'm goin'" and "the ocean." Rules-based systems were unsuccessful because they couldn't handle these variations. This also explains why earlier systems could not handle continuous speech -- you had to speak each word separately, with a brief pause in between them.
4. Today's speech recognition systems use powerful and complicated statistical modeling systems. These systems use probability and mathematical functions to determine the most likely outcome. According to John Garofolo, Speech Group Manager at the Information Technology Laboratory of the National Institute of Standards and Technology, the two models that dominate the field today are the Hidden Markov Model and Neural Networks. These methods involve complex mathematical functions, but essentially, they take the information known to the system to figure out the information hidden from it.

**Hidden Markov Model**

1. The Hidden Markov Model is the most common, so we'll take a closer look at that process. In this model, each phoneme is like a link in a chain, and the completed chain is a word. However, the chain branches off in different directions as the program attempts to match the digital sound with the phoneme that's most likely to come next. During this process, the program assigns a probability score to each phoneme, based on its built-in dictionary and user training.



1. This process is even more complicated for phrases and sentences -- the system has to figure out where each word stops and starts. The classic example is the phrase "recognize speech," which sounds a lot like "wreck a nice beach" when you say it very quickly. The program has to analyze the phonemes using the phrase that came before it in order to get it right. Here's a breakdown of the two phrases:

**r  eh k ao g n ay  z       s  p  iy  ch**

**"recognize speech"**

**r  eh  k     ay     n  ay s     b  iy  ch**

**"wreck a nice beach"**

1. Why is this so complicated? If a program has a vocabulary of 60,000 words (common in today's programs), a sequence of three words could be any of 216 trillion possibilities. Obviously, even the most powerful computer can't search through all of them without some help
2. That help comes in the form of program training. According to John Garofolo.
3. These statistical systems need lots of exemplary training data to reach their optimal performance -- sometimes on the order of thousands of hours of human-transcribed speech and hundreds of megabytes of text. These training data are used to create acoustic models of words, word lists, and [...] multi-word probability networks. There is some art into how one selects, compiles and prepares this training data for "digestion" by the system and how the system models are "tuned" to a particular application. These details can make the difference between a well-performing system and a poorly-performing system -- even when using the same basic algorithm.
4. No speech recognition system is 100 percent perfect; several factors can reduce accuracy. Some of these factors are issues that continue to improve as the technology improves. Others can be lessened -- if not completely corrected -- by the user.

**Neural network**

1. We’ll use a [recurrent neural network](https://medium.com/@ageitgey/machine-learning-is-fun-part-2-a26a10b68df3) — that is, a neural network that has a memory that influences future predictions. That’s because each letter it predicts should affect the likelihood of the next letter it will predict too. For example, if we have said “HEL” so far, it’s very likely we will say “LO” next to finish out the word “Hello”. It’s much less likely that we will say something unpronounceable next like “XYZ”. So having that memory of previous predictions helps the neural network make more accurate predictions going forward.

***Speech Recognition: Weaknesses and Flaws***

**Low signal-to-noise ratio**

The program needs to "hear" the words spoken distinctly, and any extra noise introduced into the sound will interfere with this. The noise can come from a number of sources, including loud background noise in an office environment. Users should work in a quiet room with a quality [microphone](https://electronics.howstuffworks.com/gadgets/audio-music/question309.htm) positioned as close to their mouths as possible. Low-quality [sound cards](https://computer.howstuffworks.com/sound-card.htm), which provide the input for the microphone to send the signal to the computer, often do not have enough shielding from the electrical signals produced by other computer components. They can introduce hum or hiss into the signal.

**Overlapping speech**

Current systems have difficulty separating simultaneous speech from multiple users. "If you try to employ recognition technology in conversations or meetings where people frequently interrupt each other or talk over one another, you're likely to get extremely poor results," says John Garofolo.

**Intensive use of computer power**

Running the statistical models needed for speech recognition requires the computer's processor to do a lot of heavy work. One reason for this is the need to remember each stage of the word-recognition search in case the system needs to backtrack to come up with the right word. The fastest personal computers in use today can still have difficulties with complicated commands or phrases, slowing down the response time significantly. The vocabularies needed by the programs also take up a large amount of hard drive space. Fortunately, disk storage and processor speed are areas of rapid advancement -- the computers in use 10 years from now will benefit from an exponential increase in both factors.

**Homonyms**

1. Homonyms are two words that are spelled differently and have different meanings but sound the same. "There" and "their," "air" and "heir," "be" and "bee" are all examples. There is no way for a speech recognition program to tell the difference between these words based on sound alone. However, extensive training of systems and statistical models that take into account word context have greatly improved their performance.

***Text to Speech***

1. In order to reproduce the natural sound of each language, a narrator records a series of texts (poetry, political news, sports results, stock exchange updates, etc.) which contain every possible sound in the chosen language.
2. These recordings are then sliced and organized into an acoustic database.
3. During database creation, all recorded speech is segmented into some or all of the following: diphones, syllables, morphemes, words, phrases, and sentences.
4. To reproduce words from a text, the TTS system begins by carrying out a sophisticated linguistic analysis that transposes written text into phonetic text.
5. A grammatical and syntactic analysis then enables the system to define how to pronounce each word in order to reconstruct the sense. We call this the prosody: it gives the rhythm and intonation of a sentence.
6. Finally, the system produces information associating the phonetic writing with the tone and required length of the pronunciation.  
   The chain of analysis ends here and sound is generated by selecting the best units stocked in the acoustic database.

**System Requirements**

1. Python 3.x preferably anaconda
2. Following modules must be installed in python

* Speech recognition module,
* pyaudio module,
* Gtts,
* Pygame,
* pyttsx module

We can use *pip install* command to install above mentioned module.

1. System should be microphone enabled. It should be capable of recording voice.
2. System should have speakers connected to it.

**Working Model**

1. First of all we set sampling rate approximate to the human voice i.e. 48000 and chunk size of 1024 kb.
2. After checking all the available microphones on the system, we verified and took permission to use specified microphone.
3. The microphone will then record the audio from the microphone source set in step 2.
4. This audio file is passed to Google speech recognition Api which convert speech to text using pre trained recurrent neural network.
5. A *Neural Network* that has a memory that influences future predictions. That’s because each letter it predicts should affect the likelihood of the next letter it will predict too.
6. The string received from Google API is then fed to tokenizer. That will convert the string into a number of tokens.
7. *Operand Identifier:* This is a classifier which will identify the operands among the token generated from step 6.
8. If the number of operand identified in step 7 is less than two, the talking calculator will notify the user about the concerned error.
9. *Operator Classifier:* A class for each operator is created that contain all the abbreviation of that operator. This classifier will classify any abbreviation of operator to its literal meaning.
10. After the identification of operator and operands, we perform asked calculation.
11. The answer is then given to gTTS which use sophisticated linguistic analysis and syntactical analysis to convert answer to speech.
12. The system is capable to remind the result of last operation and thus can be used to calculate a chain of arithmetic calculations.