Speech Idiosyncrasies are the Nemesis of Speech Recognition Software

Theresa L. Ford

University of Maryland University College
COMM 380: Language in the Social Contexts
Summer 2004
Section 6980
Dr. Lucinda Hart-González

July 18, 2004

Speech Idiosyncrasies are the Nemesis of Speech Recognition Software

By Theresa L. Ford

"Scotty is faced with a 20th century computer.
Scotty: Hello, Computer.
(McCoy hands him a mouse.)
Scotty: *(speaking into the mouse)* Hello, Computer.
Scientist: Just use the keyboard.
Scotty: Oh, a keyboard. How quaint."
(Star Trek IV: The Voyage Home, 1986)

The computer had problems understanding Scotty's Scottish accent, not the least of which was the fact that the computer did not have any speech recognition hardware or software. People speaking English not only pronounce words many different ways, but also use and arrange them differently. Making computers transcribe human speech is an interesting exercise in understanding how humans interpret speech with all its idiosyncrasies. Sounds must be identified, as must neighboring sounds, intended morphemes chosen, words formed, and context analyzed.

SPEECH IDIOSYNCRACIES

Speech varies from person to person due to physical attributes and dialect.

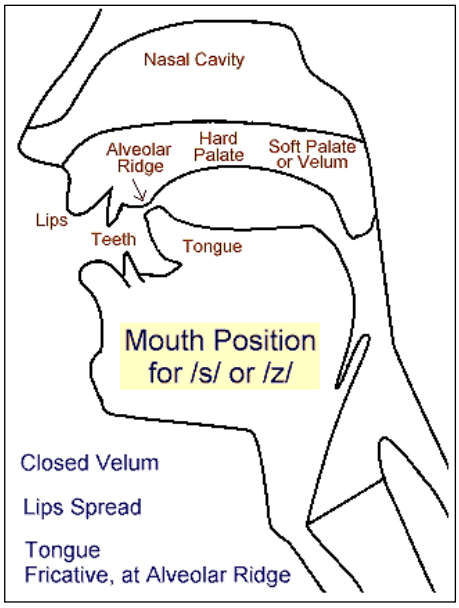


Figure 1. Mouth Position for /s/or /z/.

PHONEMES

Vocal sounds (phones or phonemes) are made by five different physical actions - vocal cord vibration, oral and nasal cavity use, position of lips, manner of articulation, and place of articulation (which may or may not include the tongue). /s/ and /z/ are pronounced with spread lips and a fricative with the tongue in alveolar placement (Figure 1). /s/ is voiceless while /z/ is voiced. (Hall, 2004)

Vowels are created by sending air over the vocal cords, past the

jaw, tongue, and lips. We have nasal sounds, fricative sounds, stops, glides, and liquids. Adding to the complexity, sounds vary from person to person so any software trying to recognize a sound would have to work within a range of frequencies. Li Deng, a researcher for Microsoft, notes that if physical sound actions and features were taken into account, speech recognition software could be more adaptable from person to person because these affect resonant frequencies. (Brooks, 2003)

We can visualize how computers perceive sound by looking at the acoustics of phonemes as represented by waveforms and spectrograms. In a waveform like Figure 2, the flat horizontal line in the middle is the atmospheric pressure at the time of the recording. Anything above that represents higher pressure while anything below represents lower pressure. Sound vibrates the air pressure, so samples of the air pressure difference at a given time are shown. Phonemes are impossible to recognize from waveforms. (Carmell, et. al., 1997)

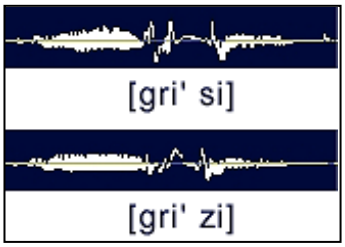


Figure 2. Waveforms for "greasy".

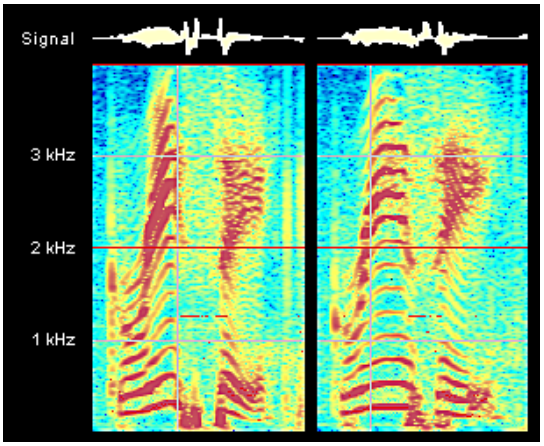


Figure 3. Spectrograms for "greasy". Left [gri' si]. Right [gri' zi].

A spectrogram, in contrast, displays the frequency and amplitude of the speech sound samples. In Figure 3, the height is the frequency from 1000 Hz to 4000 Hz, and the red indicates the higher amplitude. Each phoneme has a distinct image, like a fingerprint, so spectrograms can be read like text letters if one knows how to recognize the patterns. (Carmell, et. al., 1997)

Once the individual phonemes are identified, the morphemes need to be recognized. This can be even more complicated than deciding on individual phones. Morphemes are the smallest sound that has meaning. Morphemes can be words or can be combined to create words. Affixes,

suffixes, prefixes, and infixes are examples of dependent morphemes that have meaning yet must be attached to another morpheme to make a word. (Dictionary.com, 2004)

Depending on the context, morphemes can be interpreted in many different ways. Homonyms have the same phonemes, but are different words. Allophones are "acoustically different forms of the same phoneme". (Dictionary.com, 2004) Phoneme pronunciation, and thus morpheme identification, changes depending on what precedes and follows the phoneme (co-articulation), by which phoneme is emphasized, and where the word appears within the sentence. (Davis, 2001)

Interpretation is also affected by tone. "Emotion changes speed and tone," says Ekaterina Borisova-Maier, "My cat knows what I mean based on my tone. Animals do not speak Russian, English, or German." They hear rising and falling tones and speed. (Borisova-Maier, 2004) Dr. Cheryl Davis said, "Spoken words are like snowflakes. No one person ever says a given word exactly the same way twice." (Davis, 2001)

Luckily, morphemes follow patterns that help in identifying them. Some phonemes only appear at the beginning or end of a morpheme. Some morphemes appear only at the beginning or end of a word, or always occur alone. Morphemes tend to be consonant to vowel sound to consonant again. Human speakers and listeners do not think about how they recognize words, but computers must be programmed to interpret them correctly, and thus phonemes and morphemes need to be carefully analyzed to handle speech idiosyncrasies.

In the next section, we'll look at the history of computer speech recognition and then how computers analyze speech sounds to correctly interpret them.

HISTORY OF COMPUTER SPEECH RECOGNITION

Voice was combined with computers in 1936 when AT&T's Bell Labs created Voder, a speech synthesizer. This was an important first step toward speech recognition as computers began to work with sounds. (Speech Recognition Software and Medical Transcription History, 2003, and Valle, 1997) A recording of Voder being demonstrated at the New York World's Fair in 1939 can be heard at

http://www.cs.indiana.edu/rhythmsp/ASA/highlights.html. Listening to Voder, one can hear that the rhythm is not exactly right and the sound is obviously synthesized. (Klatt, 1987)

Having software pronounce written words from a static set of sounds, however, is quite different than software trying to recognize the large variety of spoken sounds and matching those to the correct word. For instance, every time a computer encounters the letters "greasy", it can send the same sounds to the computer speakers. The opposite, though, how a computer matches both [gri' si] and [gri' zi] to "greasy" is another task. This can be determined using Hidden Markov Modeling which was designed by Lenny Baum in the early 1970's. Hidden Markov Modeling defines how to match multiple speech patterns to their correct word. (Speech Recognition Software and Medical Transcription History, 2003, and Valle, 1997)

In 1971, Defense Advanced Research Projects Agency (DARPA) began the Speech Understanding Research program to create a computer that could recognize continuous speech. In 1984, SpeechWorks started "over-the-telephone automated speech recognition". This began speech recognition for specialized tasks with limited vocabularies and a small number of possible word arrangements. (Speech Recognition Software and Medical Transcription History, 2003, and Valle, 1997)

In 1995, Dragon Systems sold the first commercially available voice recognition software, DragonDictate, which required a slight pause (1/10th of a second) between words. Dragon System's NaturallySpeaking followed in 1997 and allowed more continuous speech. This began the design of dictation systems capable of recognizing a huge number of varied words in any order. (Speech Recognition Software and Medical Transcription History, 2003, and Valle, 1997)

In 1996, Charles Schwab began voice recognition IVR (Interactive Voice Response), in which human response navigates through voice telephone menus for information, demonstrating how speech recognition can be applied and useful commercially. (Speech Recognition Software and Medical Transcription History, 2003, and Valle, 1997)

Speech recognition software has progressed substantially and is accurate for automated telephone menus,

but still needs higher accuracy for mainstream usage.

MECHANICS OF COMPUTER SPEECH RECOGNITION

In How Computers Work by Ron White, the process a computer goes through to recognize spoken words is broken into eight simple steps. First, the software is configured to the individual's phoneme pronunciation. The person reads predesignated words and sentences and the program maps pronunciations to the correct word, notes background noise and microphone acoustics. (White, 2004, p. 222)

Next, the person speaks into his microphone. The type of microphone and the computer hardware make a noticeable difference in how the sound is registered. Microphones should have noise-cancellation to remove background noise, and the sound card and computer processor should be fast enough to allow a high sampling rate. (White, 2004, p. 222)

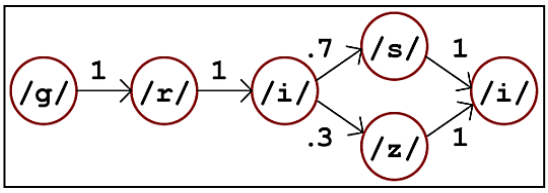
In the third step, the sound, now an analog signal from the microphone, is "sampled by an analog-to-digital converter (ADC)" (White, 2004, p. 222). The ADC translates the signal into series of 1's and 0's that represent "pitch, volume, frequency, length of phonemes, and silences". The 1's and 0's are then compressed into a format the recognition software expects. The compressed data is passed to the speech recognition software. (White, 2004, p. 222)

In the fourth step, the software adjusts the data to account for the background noise and microphone acoustics noted in the first step. The measurements are adjusted for individual pronunciation. (White, 2004, p. 222)

Step five takes each phoneme and finds the best match stored in its collection of phonemes recorded from thousands of people. It finds these matches based on the measurements from step three - pitch, volume, frequency, etc. Once the phonemes are matched, the words need to be identified. (White, 2004, p. 222)

In step six, then, the lexicon database is searched for the same series of phonemes. Because there are so many words in a lexicon, sometimes the databases are customized to a specialized task, for instance, a medical transcription dictionary. (White, 2004, p. 222-223)

Step seven recognizes homonyms and tries to distinguish them by comparing groups of words to common phrases,



applying grammatical rules, and aiming for natural spoken language. The computer determined words appear on the screen. (White, 2004, p. 222-223) This tries to emulate how people identify words. As Michael Brooks writes, "Humans, for instance, use a knowledge of grammar, syntax and semantics to narrow down the range of possible words they are listening to, and linguists have spent decades studying how this works." It is common for phonemes to be changed, modified, or dropped by different speakers. (Brooks, 2003)

If the computer cannot determine a best match for the sounds, the software prompts the speaker for what was meant as step eight. Simply put, the sounds (phones) are acquired by the computer, matched against a set of phonemes, groups of phonemes looked up in a dictionary, checked for homonyms, and displayed. (White, 2004, p. 222-223)

HIDDEN MARKOV MODEL

How does a computer recognize words from series of phonemes? The Markov Model and Hidden Markov Model are key software logic models used in speech prediction and recognition. They are named after Andrei A. Markov, who wrote a paper in 1913 containing the mathematics that would shape how computers would determine words from sounds. Mr. Markov used mathematical formulas to predict vowels and consonants in the Russian poem, "Eugene Onegin". From these formulas, a flowchart-like style was created to find matches and is named the Markov Model. (Fry, 2003)

The Markov Model maps sequences of events and is what computer scientists refer to as a finite state machine. Each event or state, a phoneme, for instance, is connected to another state. Probabilities are used to determine the next state. The chance that a particular path is followed through the Markov model is the product of the probability each step takes. (Fry, 2003)

For example, using the two pronunciations of "greasy" from the six geographic locations in the COMM 380 Dialect Geography Exercise, [gri' si] and [gri' zi], the phoneme path is /g/, /r/, /i/, /s/ or /z/, /i/. /s/ occurs 4 out of 6 times (.7 probability). /z/ occurs 2 out of 6 times (.3

probability). As there is a 100% chance the sounds will progress from /g/ to /r/ and so on, the end result of [gri' si] is 1*1*1*.7*1, which is .7, or a 70% probability. As words gain multiple syllables and different pronunciations of those syllables (like the word "tomato"), predicting the end probability of each variance is easy using this method. The result from the Markov Model is the phoneme path taken; in the case of "greasy", it would be the representation for [gri' si] or [gri' zi]. (Fry, 2003)

Figure 4. Markov Model for "greasy".

A Hidden Markov Model is different from the Markov Model because it does not keep track of the path taken, merely gives the resulting word choice (Figure 4 would return "greasy", not [gri' si]). The path is hidden. Hidden Markov Models are also more complex; multiple sequences potentially produce the same results, and transition probabilities can be adjusted based on use. (Kadous, 1995)

SIMPLE CASE STUDY - AIR FLIGHT SIMULATOR COMMANDS

Drew Kirkpatrick, who worked on configuring speech recognition software for specific commands for a Navy application said that for his command set, specific phonemes must always appear in a specific order. Mr. Kirkpatrick said that his air flight simulator's software did not have a problem with faster speakers because the sound card had a high enough sampling rate that the computer could sill recognize phonemes. Sampling rate is the speed at which the computer captures slices of sound. (Kirkpatrick, 2004)

He configured his system by recording himself carefully pronouncing each command word in a variety of ways. Any born and raised American who tested his software was recognized; however, when tested by foreign speakers, his software did not recognize the words. Several Russians, who spoke very good English according to American listeners, had slightly different ways of pronouncing words and that variation was not programmed into the software. (Kirkpatrick, 2004)

Mr. Kirkpatrick's software also required that words be spoken in a specific order. He said that if words were presented out of order, the system got confused. He commented that more paths (routes) could be added to handle more variations in word order, but project deadlines limited

the paths allowed. This is the difference between saying, "Open door.", "Open the door.", and "Door, open.". These mean the same thing and the same results are expected but the words have different paths that would have to be recognized by the software, open->door, open->the->door, or door->open. Speakers for the air flight simulator had to be trained how to communicate with the recognition software, that is, to use the expected word order in their commands. (Kirkpatrick, 2004)

In this example, the sampling rate determines how fast the speakers are allowed to speak. The software is configured to recognize specific commands. Individual words have a specific required order and the phonemes within each word have to be in a specific order.

CURRENT SOFTWARE ACCURACY AND USABILITY

Typewell.com, in its evaluation of speech recognition software in the classroom states that current software claims of 98% accuracy are exaggerated and generated under ideal testing conditions with minimal background noise and microphone movement and with trained dictators. It suggests that even a 1% error rate in word recognition produces a 5% error rate in sentence meaning, which is critical in a classroom environment where the deaf would receive the wrong message. For example, "senior years" could be recognized as "seen your ears". (Typewell.com, 2004)

Janet Rae-Dupree notes that the National Business Education Association "recommends that all students from fourth through 12th grade learn how to use both speech and handwriting recognition software (as well as learn how to type)." Speech recognition software that accepts free-form dictation is being used currently. Once the human and computer are trained, it is faster and more accurate than typing. Ms. Dupree cites that Karl Barkside dictated 30 of his 43 books going from a typing speed of 70 words per minute to a talking speed of 140 words per minute. (Rae-Dupree, 2003)

Which of these two views really represents the current state of speech recognition by computers? That people should concentrate on typing and human transcription because word recognition is not accurate enough to be useful, or that everyone should learn to use dictation software because it is going to replace typing in the near future? While

Typewell.com is obviously a commercial venture employing human transcriptionists, it brings up an important point for using speech recognition software - how critical is immediate accuracy? Karl Barkside dictating his 30 books would have had time to edit and had editors reviewing what he had dictated, whereas a teacher presenting a lecture to deaf and blind students reading Braille would want a high degree of immediate accuracy.

Speech recognition software is useful and accurate for limited and specialized sets of words already. Ekaterin Borisova-Maier, a Russian linguist who is now married to an American and living in the United States, said she does not have problems navigating voice-response automated menus over the telephone. She says she makes a point to enunciate clearly and to speak louder, but notes those systems "only ask for specific responses, like yes or no, or account number". (Borisova-Maier, 2004)

Janet Rae-Dupree, when first trying to use speech recognition software, comments that "...reciting fully formed sentences doesn't come naturally..." (Rae-Dupree, 2003). A person dictating to a human does not have to modify his language or speaking habits; the transcriptionist edits out "ahs", "ums", yawns and interruptions, and also might add implied words, correct grammar, and will have appropriate homonym choice, and punctuation. Bill Meisel of TMA Associates is quoted by Ms. Dupree, saying, "It's a learned skill to say something the way you want it to appear on paper. Most people have no idea how to dictate." (Rae-Dupree, 2003) Speech recognition software needs to be able to handle these vocalizations or the speakers need to learn not to include them when dictating.

Even if a computer accurately recognized and transcribed speech, the software would have to be updated and maintained. Ekatarin Borisova-Maier points out that "Any language recognition software would grow out of date because language changes over time - so 50 years from now, the software would not work, not that software is used that long." Software would have to be dynamic enough to learn the new pronunciations and rules as time progresses or it would quickly become outdated. The dynamics of language are readily apparent when we discourse with people with a different accent. We adopt new pronunciations from hearing words pronounced differently, sometimes merely from finding the new pronunciation amusing. (Borisova-Maier, 2004)

SUMMARY

The smaller the set of words to be recognized and the less deviation from a set order of words, the more accurate speech recognition software is when dealing with speech idiosyncrasies. Phoneme variation must fall within defined sound ranges to be recognized. Morpheme variation within words where phonemes may be pronounced differently or dropped entirely has to be included as a path within each word's Markov Model to be identified. Accurate prediction of homonyms, word boundaries, and punctuation rely on complex grammar rules.

Smaller scale voice recognition systems, like Interactive Voice Response over the telephone or specialized application commands, are able to handle speech idiosyncrasies better than dictation software that require individual software training (configuration), robust homonym and grammar interpretation, and human dictation skills.

FIGURES

Figure 1. Mouth Position for /s/or /z/. (Hall, 2004, and Glenn, Glenn, and Forman, 1984). Mouth drawing from Hall's Sammy Website with labels added.

Figure 2. Waveforms for "greasy". Same speaker pronouncing greasy as [gri' si] and [gri' zi] recorded as 8.000 kHz, 16 Bit, Mono WAV files. Waveform images generated by ReZound.

Figure 3. Spectrograms for [gri' si] and [gri' zi]. These are the same WAV files as Figure 2, shown using Spectrogram version 10.3 by Visualization Software LLC, using a linear frequency scale with a frequency resolution of 16.8 and pitch detection.

Figure 4. Markov Model for "greasy". Developed using John Fry's examples in his presentation on Hidden Markov Models.

NOTES

1. You can read more about spectrograms and view American English Phonemes as spectrograms at http://cslu.cse.ogi.edu/tutordemos/SpectrogramReading/ipa/ipahome.html.

REFERENCES

Bennett, H., Krikes, P., Meerson, S., & Meyer, N. (1986). *Star Trek IV: The Voyage Home.* Retrieved July 3, 2004, from http://robert.walkertribe.com/interests/trek/quotes/st4.asp

Borisova-Maier, E. (2004). Interview July 4, 2004.

Brooks, M. (2003). *No one understands me as well as my PC.* New Scientist, v. 180 (Nov. 1 2003), p. 28-31. Retrieved June 19, 2004, from WilsonSelectPlus database.

Carmell, T., Cronk, A., Kaiser, E., Wesson, R., Wouters, J., and Wu, X. (1997). *Spectrogram Reading.* Retrieved July 15, 2004, from http://cslu.cse.ogi.edu/tutordemos/SpectrogramReading/spectrogram_reading.html

Cohen, M. F. (1998). *Small tawk.* The New Republic v. 219, No. 14 (Oct. 5 1998), p. 11-12. Retrieved June 19, 2004, from WilsonSelectPlus database.

Davis, C. (2001). *Automatic Speech Recognition and Access: 20 years, 20 months, or tomorrow?* Hearing Loss, 22(4), p. 11-14. Retrieved on July 5, 2004, from http://www.wou.edu/education/sped/nwoc/asr.htm

Dictionary.com. (2004). *Allophone.* Retrieved on July 5, 2004, from http://dictionary.reference.com/search?q=allophone

Fry, J. (2003). *Hidden Markov Models.* Retrieved June 19, 2004, from http://www.sjsu.edu/~jfry/124/hmm.pdf

Glenn, E. C., Glenn, P. J., & Forman, S. H. (1984). *Your Voice and Articulation.* United States of America: Macmillan Publishing Company.

Hart-González, L. (2004). Editing comments.

Hartley, J. (2003). *Speaking versus typing: a case-study of the effects of using voice-recognition software on academic correspondence.* British Journal of Educational Technology v. 34, No. 1 (Jan. 2003), p. 5-16. Retrieved

June 19, 2004, from WilsonSelectPlus database.

Hall, D. C. *Interactive Sagittal Section. Sammy.* Retrieved July 5, 2004, from http://www.chass.utoronto.ca/~danhall/phonetics/sammy.html

Kadous, M. W. (1995). *Hidden Markov Models from Recognition of Australian Sign Language Using Instrumented Gloves.* Retrieved July 5, 2004, from http://www.cse.unsw.edu.au/~waleed/thesis/node39.html

Kirkpatrick, D. (2004). Interview June 24, 2004.

Klatt, D. (1987). *History of Speech Synthesis.* Cited by Kitahara, M. & Port, R. F. Retrieved on July 13, 2004, from http://www.cs.indiana.edu/rhythmsp/ASA/highlights.html

Ledger, G. R. (2004). *Pushkin's Poems.* Retrieved July 5, 2004, from http://www.pushkins-poems.com/

Rae-Dupree, J. (2003). *Let's Talk.* U.S. News & World Report v. 134, No. 16 (May 12 2003), p. 58-9. Retrieved June 19, 2004, from WilsonSelectPlus database.

ReZound. (2004). Retrieved from June 18, 2004, from http://rezound.sourceforge.net/

Spectrogram. (2004). Retrieved July 16, 2004, from http://www.visualizationsoftware.com/gram.html

Speech Recognition Software and Medical Transcription History, A TIMELINE OF SPEECH RECOGNITION. (2003). Retrieved July 3, 2004, from http://www.dragon-medical-transcription.com/historyspeechrecognitiontimeline.html

Talbot, D. (2002). *Prosody: Computers will soon hear your pain.* Technology Review (Cambridge, Mass.: 1998) v. 105 no6 (July/Aug. 2002), p. 27. Retrieved June 19, 2004, from WilsonSelectPlus database.

Tremlett, C. (2004). Interview July 13, 2004.

Typewell.com. *Speech Recognition in the Classroom.* (2004). Retrieved July 5, 2004, from http://www.typewell.com/speechrecog.html

University of Maryland University College, COMM 380, Language in the Social Contexts. (2004). *Dialect Geography Exercise.*

Valle, G. (1997). *DragonDictate and NaturallySpeaking Compared.* Software Maintenance, Inc. Retrieved July 3, 2004, from http://www.ddwin.com/overview.htm

Wegman, E. J., Symanzik, J., Vandersluis, J. P., Luo, Q., Camelli, F., Dzubay, A., et al. (1999). *The MiniCAVE - A Voice-Controlled IPT Environment.* Retrieved June 19, 2004, from http://www.math.usu.edu/~symanzik/papers/1999_ipt.pdf

Wheeldon, L. & Waksler, R. (2004). *Phonological underspecification and mapping mechanisms in the speech recognition lexicon.* Brain and Language, Volume 90, Issues 1-3, July-September 2004, p. 401-412. Retrieved June 19, 2004, from ScienceDirect database.

White, R. (2004). *How Computers Work.* United States of America: Que Publishing.