Language and Computers (Ling 384)

Topic 1: Text and Speech Encoding

Adriane Boyd*

Winter 2006

Contents

*This course was created by Markus Dickinson, Detmar Meurers and Chris Brew.

1 Introduction

Language and Computers – where to start?

- If we want to do anything with language, we need a way to represent language.
- We can interact with the computer in several ways:
  - write or read text
  - speak or listen to speech
- Computer has to have some way to represent
  - text
  - speech

2 Writing systems

Writing systems used for human languages

What is writing?

> "a system of more or less permanent marks used to represent an utterance in such a way that it can be recovered more or less exactly without the intervention of the utterer." (Peter T. Daniels, *The World's Writing Systems*)

> "Words that stay." (-Jen (Jim Henson), *The Dark Crystal*)

Different types of writing systems are used:

- Alphabetic
- Syllabic
- Logographic

Much of the information on writing systems and the graphics used are taken from the amazing site http://www.omniglot.com.

2.1 Alphabetic

Alphabetic systems

Alphabets (phonemic alphabets)

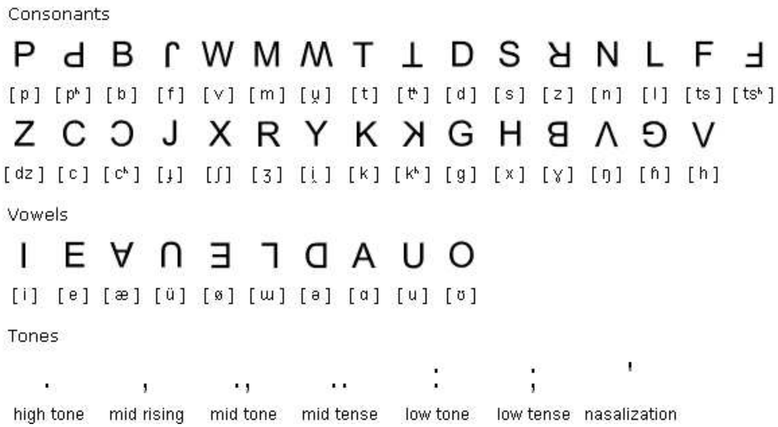
- represent all sounds, i.e., consonants and vowels
- Examples: Etruscan, Latin, Korean, Cyrillic, Runic, International Phonetic Alphabet

Abjads (consonant alphabets)

- represent consonants only (sometimes plus selected vowels; vowel diacritics generally available)
- Examples: Arabic, Aramaic, Hebrew

Alphabet example: Fraser

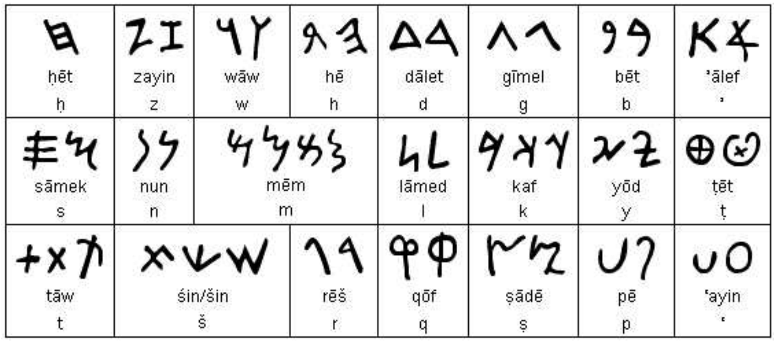
An alphabet used to write Lisu, a Tibeto-Burman language spoken by about 657,000 people in Myanmar, India, Thailand and in the Chinese provinces of Yunnan and Sichuan.



(from: http://www.omniglot.com/writing/fraser.htm)

Abjad example: Phoenician

An alphabet used to write Phoenician, created between the 18th and 17th centuries BC; assumed to be the forerunner of the Greek and Hebrew alphabet.



(from: http://www.omniglot.com/writing/phoenician.htm)

A note on the letter-sound correspondence

- Alphabets use letters to encode sounds (consonants, vowels).
- But the correspondence between spelling and pronounciation in many languages is quite complex, i.e., not a simple one-to-one correspondence.
- Example: English
  - same spelling – different sounds: *ough*: *ought*, *cough*, *tough*, *through*, *though*, *hiccough*
  - silent letters: *knee*, *knight*, *knife*, *debt*, *psychology*, *mortgage*
  - one letter – multiple sounds: *exit*, *use*
  - multiple letters – one sound: *the*, *revolution*
  - alternate spellings: *jail* or *gaol*; but not possible *seagh* for *chef* (despite *sure*, *dead*, *laugh*)

More examples for non-transparent letter-sound correspondences

French

(1) a. *tailles* → [taj]
b. *étais*, *était*, *étaient* → [etɛ]

Irish

(2) a. *Baile A'tha Cliath* (Dublin) → [bl'aː kli uh]
b. *samhradh* (summer) → [sauruh]
c. *scri'obhaim* (I write) → [ʃgriːm]

What is the notation used within the [ ]?

The International Phonetic Alphabet (IPA)

- Several special alphabets for representing sounds have been developed, the best known being the International Phonetic Alphabet (IPA).
- The phonetic symbols are unambiguous:
  - designed so that each speech sound gets its own symbol,
  - eliminating the need for
    - multiple symbols used to represent simple sounds
    - one symbol being used for multiple sounds.
- Interactive example chart: http://web.uvic.ca/ling/resources/ipa/charts/IPAlab/IPAlab.htm

2.2 Syllabic

Syllabic systems

Syllabic alphabets (Alphasyllabaries)

- writing systems with symbols that represent a consonant with a vowel, but the vowel can be changed by adding a **diacritic** (= a symbol added to the letter).
- Examples: Balinese, Javanese, Tibetan, Tamil, Thai, Tagalog*

(cf. also: http://www.omniglot.com/writing/syllabic.htm)

Syllabaries

- writing systems with separate symbols for each syllable of a language
- Examples: Cherokee, Ethiopic*, Cypriot, Ojibwe, Hiragana (Japanese)

(cf. also: http://www.omniglot.com/writing/syllabaries.htm#syll)

Syllabary example: Cypriot*

The Cypriot syllabary or Cypro-Minoan writing is thought to have developed from the Linear A, or possibly the Linear B script of Crete, though its exact origins are not known. It was used from about 800 to 200 BC.

(from: http://www.omniglot.com/writing/cypriot.htm)

2.3 Logographic

Logographic writing systems

- Logographs (also called Logograms):
  - **Pictographs (Pictograms)**: originally pictures of things, now stylized and simplified.

    Example: development of Chinese character *horse*:

  - **Ideographs (Ideograms)**: representations of abstract ideas
  - **Compounds:** combinations of two or more ideographs or ideograms.
  - **Semantic-phonetic compounds:** symbols with a meaning element (hints at meaning) and a phonetic element (hints at pronunciation).

- Examples: Chinese (Zhōngwén), Japanese (Nihongo), Mayan*, Vietnamese 'chu nom'*, Ancient Egyptian

Logograph writing system example: Chinese

Pictographs

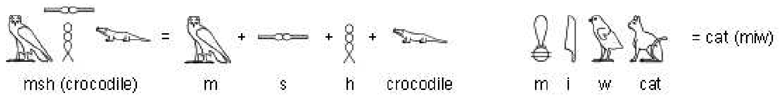
Ideographs

Compounds of Pictographs/Ideographs

(from: http://www.omniglot.com/writing/chinese_types.htm)

Semantic-phonetic compounds

An example from Ancient Egyptian



(from: http://www.omniglot.com/writing/egyptian.htm)

2.4 Systems with unusual realization

Two writing systems with unusual realization

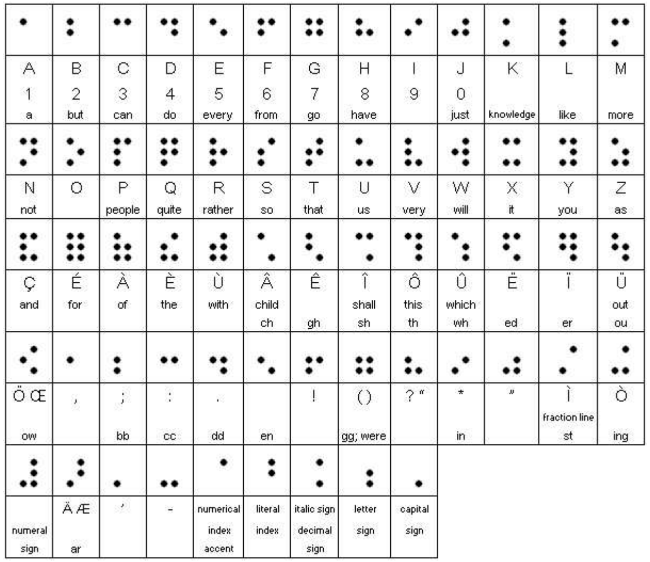
Tactile

- Braille is a writing system that makes it possible to read and write through touch; primarily used by the (partially) blind.
- It uses patterns of raised dots arranged in cells of up to six dots in a 3 x 2 configuration.
- Each pattern represents a character, but some frequent words and letter combinations have their own pattern.

Chromatographic The Benin and Edo people in southern Nigeria have developed a system of writing based on different color combinations and symbols.

(cf. http://www.library.cornell.edu/africana/Writing_Systems/Chroma.html)

Braille alphabet



Chromatographic system

2.5 Relation to language

Relating writing systems to languages

- There is not a simple correspondence between a writing system and a language.
- For example, English uses the Roman alphabet, but Arabic numerals (e.g., 2 instead of the Roman II).
- We'll look at three other examples:
  - Japanese
  - Korean
  - Azeri

Japanese

Japanese: logographic system *kanji*, syllabary *katakana*, syllabary *hiragana*

- kanji: 5,000-10,000 borrowed Chinese characters
- katakana
  - Used mainly for non-Chinese loan words, onomatopoeic words, foreign names, and for emphasis
- hiragana
  - Originally used only by women (10th century), but codified in 1946 with 48 syllables
  - used mainly for word endings, kids' books, and for words with obscure kanji symbols
- Romaji: Roman characters

Japanese example

カプセルホテル

各室がカプセル形の簡易ホテル。終電に乗り遅れたサラリーマンなどが高いタクシー代を払って帰宅するより安く済むことから、手軽に利用している。

kanji (red), hiragana (black), katakana (blue)

Translation:

> Capsule Hotel A simple hotel where each room is capsule-shaped. When businessmen miss the last train home, they can stay overnight very cheaply instead of paying a lot of money to go home by taxi.

(from: http://www.omniglot.com/writing/japanese.htm#origin)

Korean

"Korean writing is an alphabet, a syllabary and logography all at once." (http://home.vicnet.net.au/~ozideas/writkor.htm)

- The *hangul* system was developed in 1444 during King Sejong's reign.
  - There are 24 letters: 14 consonants and 10 vowels
  - But the letters are grouped into syllables, i.e. the letters in a syllable are not written separately as in the English system, but together form a single character.

    E.g., "Hangeul" (from: http://www.omniglot.com/writing/korean.htm):

    한 (han) ㅎ(h) + ㅏ(a) + ㄴ(n) 글 (geul) ㄱ(g) + ㅡ(eu) + ㄹ(l)
- In South Korea, *hanja* (logographic Chinese characters) are also used.

Azeri

A Turkish language with speakers in Azerbaijan, northwest Iran, and (former Soviet) Georgia

- 7th century until 1920s: Arabic scripts. Three different Arabic scripts used
- 1929: Latin alphabet enforced by Soviets to reduce Islamic influence.
- 1939: Cyrillic alphabet enforced by Stalin
- 1991: Back to Latin alphabet, but slightly different than before.

  → Latin typewriters and computer fonts were in great demand in 1991

2.6 Comparison of systems

Comparison of writing systems

What are the pros and cons of each type of system?

- accuracy: Can every word be written down accurately?
- learnability: How long does it take to learn the system?
- cognitive ability: Are some systems unnatural? (e.g. Does dyslexia show that alphabets are unnatural?)
- language-particular differences: English has thousands of possible syllables; Japanese has very few in comparison
- connection to history/culture: Will changing a writing system have social consequences?

3 Encoding written language

Encoding written language

- Information on a computer is stored in **bits**.
- A bit is either on (= 1, yes) or off (= 0, no).
- A list of 8 bits makes up a **byte**, e.g., 01001010
- Just like with the base 10 numbers we're used to, the order of the bits in a byte matters:
  - **Big Endian**: most important bit is leftmost (the standard way of doing things)
    - The positions in a byte thus encode: 128 64 32 16 8 4 2 1
    - "There are 10 kinds of people in the world; those who know binary and those who don't" (from: http://www.wlug.org.nz/LittleEndian)
  - **Little Endian**: most important bit is rightmost (only used on Intel machines)
    - The positions in a byte thus encode: 1 2 4 8 16 32 64 128

Converting binary numbers to decimals
The first 3 bits on a Big Endian machine:

2^2	2^1	2^0	=	4 2 1		
0	0	0	=	0 + 0 + 0	=	0
0	0	1	=	0 + 0 + 1	=	1
0	1	0	=	0 + 2 + 0	=	2
0	1	1	=	0 + 2 + 1	=	3
1	0	0	=	4 + 0 + 0	=	4

For all 8 bits in a byte:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0 =
128	64	32	16	8	4	2	1
1	0	1	1	0	0	1	1 =
128 +	0 +	32 +	16 +	0 +	0 +	2 +	1 = 179

Converting decimal numbers to binary - Tabular Method
Using the first 4 bits, we want to know how to write *10* in bit (or *binary*) notation.

8	4	2	1
?	?	?	?
8 < 10	?	?	?
1	8 + 4 = 12 > 10	?	?
1	0	8 + 2 = 10 = 10	?
1	0	1	0

Converting decimal numbers to binary - Division Method

Decimal	Remainder?	Binary
10/2 = 5	no	*0*
5/2 = 2	yes	*1*0
2/2 = 1	no	*0*10
1/2 = 0	yes	*1*010

Using bytes to store characters
With 8 bits (a single byte), you can represent 256 different characters. Why would we want so many?

- If you look at a keyboard, you will find lots of non-English characters.
- With 256 possible characters, we can store every single letter used in English, plus all the things like commas, periods, space bar, percent sign (%), back space, and so on.

3.1 ASCII

An encoding standard: ASCII

- **ASCII** = the American Standard Code for Information Interchange
- 7-bit code for storing English text
- 7 bits = 128 possible characters.
- The numeric order reflects alphabetic ordering.

The ASCII chart
Codes 1–31 are used for control characters (backspace, line feed, tab, . . .).

32		48	0	65	A	82	R	97	a	114	r
33	!	49	1	66	B	83	S	98	b	115	s
34	"	50	2	67	C	84	T	99	c	116	t
35	#	51	3	68	D	85	U	100	d	117	u
36	$	52	4	69	E	86	V	101	e	118	v
37	%	53	5	70	F	87	W	102	f	119	w
38	&	54	6	71	G	88	X	103	g	120	x
39	'	55	7	72	H	89	Y	104	h	121	y
40	(	56	8	73	I	90	Z	105	i	122	z
41	)	57	9	74	J	91	[	106	j	123	{
42	*	58	:	75	K	92	\	107	k	124	—
43	+	59	;	76	L	93	]	108	l	125	}
44	,	60	<	77	M	94	^	109	m	126	~
45	-	61	=	78	N	95	_	110	n	127	DEL
46	.	62	>	79	O	96	`	111	o		
47	/	63	?	80	P			112	p		
		64	@	81	Q			113	q		

E-mail issues

- Have you ever had something like the following at the top of an e-mail sent to you?

  ```
  [The following text is in the ''ISO-8859-1'' character set.]
  [Your display is set for the ''US-ASCII'' character set.  ]
  [Some characters may be displayed incorrectly.  ]
  ```

- Mail sent on the internet used to only be able to transfer the 7-bit ASCII messages. But now we can detect the incoming character set and adjust the input.
- Note that this is an example of **meta-information** = information which is printed as part of the regular message, but tells us something about that message.

Multipurpose Internet Mail Extensions (MIME)
MIME provides meta-information on the text, which tells us:

- which version of MIME is being used
- what the charcter set is
- if that character set was altered, how it was altered

```
Mime-Version:  1.0 Content-Type:  text/plain; charset=US-ASCII Content-Transfer-Encoding:  7bit
```


Different coding systems
But wait, didn't we want to be able to encode *all* languages?
There are ways ...

- Extend the ASCII system with various other systems, for example:
  - ISO 8859-1: includes extra letters needed for French, German, Spanish, etc.
  - ISO 8859-7: Greek alphabet
  - ISO 8859-8: Hebrew alphabet
  - JIS X 0208: Japanese characters
- Have one system for everything → **Unicode**

Unicode
Problems with having multiple encoding systems:

- Conflicts: two encodings can use the same number for two different characters and use different numbers for the same character.
- Hassle: have to install many, many systems if you want to be able to deal with various languages

Unicode tries to fix that by having a single representation for every possible character.

> "Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language." (www.unicode.org)

3.2 Unicode

How big is Unicode?
Version 3.2 has codes for 95,221 characters from alphabets, syllabaries and logographic systems.

- Uses 32 bits – meaning we can store $2^{32} = 4,294,967,296$ characters.
- 4 billion possibilities for each character? That takes a lot of space on the computer!

Compact encoding of Unicode characters

- Unicode has three versions
  - UTF-32 (32 bits): direct representation
  - UTF-16 (16 bits): $2^{16} = 65536$
  - UTF-8 (8 bits): $2^8 = 256$
- How is it possible to encode 2^{32} possibilities in 8 bits (UTF-8)?
  - Several bytes are used to represent one character.
  - Use the highest bit as flag:
    - highest bit 0: single character
    - highest bit 1: part of a multi byte character
  - Nice consequence: ASCII text is in a valid UTF-8 encoding.

3.3 Typing it in

How do we type everything in?

- Use a keyboard tailored to your specific language

  e.g. Highly noticeable how much slower your English typing is when using a Danish-designed keyboard.
- Use a processor that allows you to switch between different character systems.

  e.g. Type in Cyrillic characters on your English keyboard.
- Use combinations of characters.

  An *e* followed by an *'* might result in an *é*
- Pick and choose from a table of characters.

So, now we can encode every language, as long as it's written.

4 Spoken language

Unwritten languages
Many languages have never been written down. Of the 6700 spoken, 3000 have never been written down.

- Salar, a Turkic language in China.
- Gugu Badhun, a language in Australia.
- Southeastern Pomo, a language in California

The need for speech

- What if we want to work with an unwritten language?
- What if we want to examine the way someone talks and don't have time to write it down?

Many applications for encoding speech:

- Building spoken dialogue systems, i.e. speak with a computer (and have it speak back).
- Helping people sound like native speakers of a foreign language.
- Helping speech pathologists diagnose problems

4.1 Transcription

What does speech look like?
We can **transcribe** (write down) the speech into a **phonetic alphabet**.

- It is very expensive and time-consuming to have humans do all the transcription.

- To automatically transcribe, we need to know how to relate the audio file to the individual sounds that we hear.

  ⇒ We need to know:

  - some properties of speech
  - how to measure these speech properties
  - how these measurements correspond to sounds we hear

4.2 Why speech is hard to represent

What makes representing speech hard?
Difficulties:

- People have different dialects and different size vocal tracts and thus say things differently
- Sounds run together, and it's hard to tell where one sound ends and another begins.
- What we think of as one sound is not always (usually) said the same: **coarticulation** = sounds affecting the way neighboring sounds are said

  e.g. *k* is said differently depending on if it is followed by *ee* or by *oo*.
- What we think of as two sounds are not always all that different.

  e.g. The *s see* is very acoustically similar to the *sh* in *shoe*

4.3 Articulation

Articulatory properties: How it's produced
We could talk about how sounds are produced in the vocal tract, i.e. **articulatory phonetics**

- *place of articulation* (where): [t] vs. [k]
- *manner of articulation* (how): [t] vs. [s]
- *voicing* (vocal cord vibration): [t] vs. [d]

But unless the computer is modeling a vocal tract, we need to know acoustic properties of speech which we can *quantify*.

4.4 Measuring sound

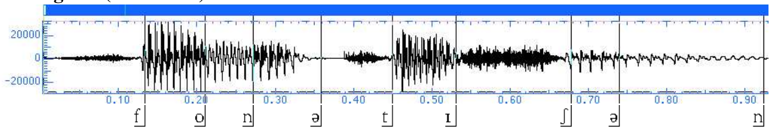
4.5 Acoustics

Acoustic properties: What it sounds like
Sound waves = "small variations in air pressure that occur very rapidly one after another" (Ladefoged, *A Course in Phonetics*)
⇒ Akin to ripples in a pond

- **speech flow** = rate of speaking, number and length of pauses (seconds)

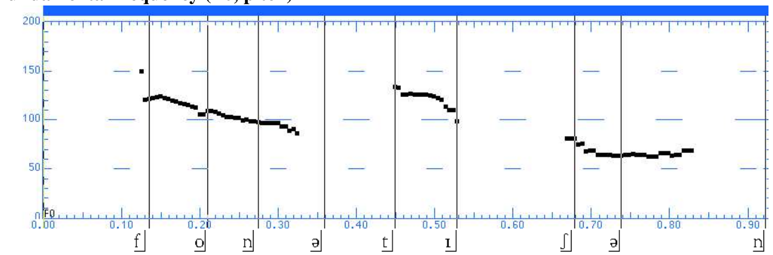
- **loudness** (amplitude) = amount of energy (decibels)
- **frequencies** = how fast the sound waves are repeating (cycles per second, i.e. Hertz)
  - **pitch** = how high or low a sound is
  - In speech, there is a **fundamental frequency**, or pitch, along with higher-frequency **overtones**.
- **intonation** = rise and fall in pitch

Oscillogram (Waveform)

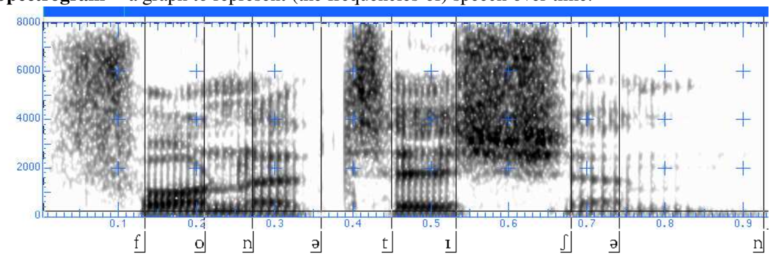


(Check out the *Speech Analysis Tutorial*, of the Deptartment of Linguistics at Lund University, Sweden at http://www.ling.lu.se/research/speechtutorial/tutorial.html, from which the illustrations on this and the following slides are taken.)

Fundamental frequency (F0, pitch)



Spectrograms
Spectrogram = a graph to represent (the frequencies of) speech over time.

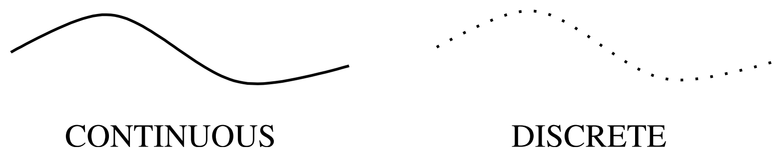


How measurements correspond to sounds we hear

- How dark is the picture? → How loud is the sound?
  We can measure this in decibels.
- Where are the lines the darkest? → Which frequencies are the loudest and most important?
  We can measure this in terms of Hertz, and it tells us what the vowels are.
- How do these dark lines change? → How are the frequencies changing over time?
  Which consonants are we transitioning into?

How did we get these measurements?
sampling rate = how many times in a given second we extract a moment of sound; measured in samples per second

- Sound is **continuous**, but we have to store data in a **discrete** manner.



- We store data at each discrete point, in order to capture the general pattern of the sound

Sampling rate

- The sampling rate is often 8000 or 16,000 samples per second. The rate for CDs is 44,100 samples/second (or **Hertz** (Hz))
- The higher the sampling rate, the better quality the recording ... but the more space it takes.
- Speech needs at least 8000 samples/second, but most likely 16,000 or 22,050 Hz will be used nowadays.

5 Relating written and spoken language

Applications of speech encoding
Mapping sounds to symbols (alphabet), and vice versa, isn't all that easy.

- **Automatic Speech Recognition (ASR)**: sounds to text
- **Text-to-Speech Synthesis (TTS)**: texts to sounds

5.1 From Speech to Text

Automatic Speech Recognition (ASR)
Automatic speech recognition = process by which the computer maps a speech signal to text.
Uses/Applications:

- Dictation
- Telephone conversations
- People with disabilities – e.g. a person hard of hearing could use an ASR system to get the text

Kinds of ASR systems
Different kinds of systems:

- Speaker dependent = work for a single speaker
- Speaker independent = work for any speaker of a given variety of a language, e.g. American English
- Speaker adaptive = start as independent but begin to adapt to a single speaker to improve accuracy

Kinds of ASR systems

- Differing sizes of vocabularies, from tens of words to tens of thousands of words
- **continuous speech** vs. **isolated-word** systems:
  - continuous speech systems = words connected together and not separated by pauses
  - isolated-word systems = single words recognized at a time, requiring pauses to be inserted between words
    → easier to find the endpoints of words

Steps in an ASR system

1. Digital sampling of speech
2. **Acoustic signal processing** = converting the speech samples into particular measurable units
3. Recognition of sounds, groups of sounds, and words

May or may not use more sophisticated analysis of the utterance to help.

5.2 From Text to Speech

Text-to-Speech Synthesis (TTS)
Could just record a voice saying phrases or words and then play back those words in the appropriate order.
Or can break the text down into smaller units

1. Convert input text into phonetic alphabet
2. Synthesize phonetic characters into speech

To synthesize characters into speech, people have tried:

- using formulas which adjust the values of the frequencies, the loudness, etc.
- using a model of the vocal tract and trying to produce sounds based on how a human would speak

It's hard to be natural
When trying to make synthesized speech sound *natural*, we encounter the same problems as what makes speech encoding in general hard:

- The same sound is said differently in different contexts.
- Different sounds are sometimes said nearly the same.
- Different sentences have different intonation patterns.
- Lengths of words vary depending on where in the sentence they are spoken.
  The car crashed into the tree.
  It's my car.
  Cars, trucks, and bikes are vehicles.

Speech to Text to Speech
If we convert speech to text and then back to speech, it should sound the same, right?

- But at the conversion stages, there is **information loss**. To avoid this loss would require a lot of memory and knowledge about what exact information to store.
- The process is thus **irreversible**.

Demos

Text-to-Speech

- AT&T mulitilingual TTS system: http://www.research.att.com/projects/tts/demo.html
- various systems and languages: http://www.ims.uni-stuttgart.de/˜moehler/synthspeech/