

Speech to text

Description

default watermark

A city thatâ??s legible is easy to understand and to navigate, i.e. to read. You can read a cityâ??s people, moods, signs, and what it denotes and connotes. In a previous post I explored the prospect that you might *write* a city, as well as *read* it. According to this theory, a city participates in basic linguistic and literary processes of reading and writing: involving meaning, alliteration, metaphor, grammar, syntax, translation, obfuscation, truth, falsehood, ellipsis, etc.

Extending the writing trope further, the city also provides a medium for *hidden* writing. Hence my willingness to associate reading and writing the city with *cryptography*.

But thereâ??s more. If you can write a city, then you can also sing it, or recite it (as a poem), or simply speak it. I follow Jacques Derrida as a language oriented philosopher in acknowledging the intimate connection between reading, writing and speaking.

Previously, I explored how scholars associate the city with the technologies of writing and print, e.g. the printing press as enabler of architectural knowledge, and thoughts about form, pattern, arrangement and re-arrangement. Iâ??d like to extend the scope of this investigation to the digital apparatuses of speech-to-text translation. Smartphones and laptops convert spoken words into written text automatically. Does speech-to-text technology impact the architecture of the city?

Automated writing

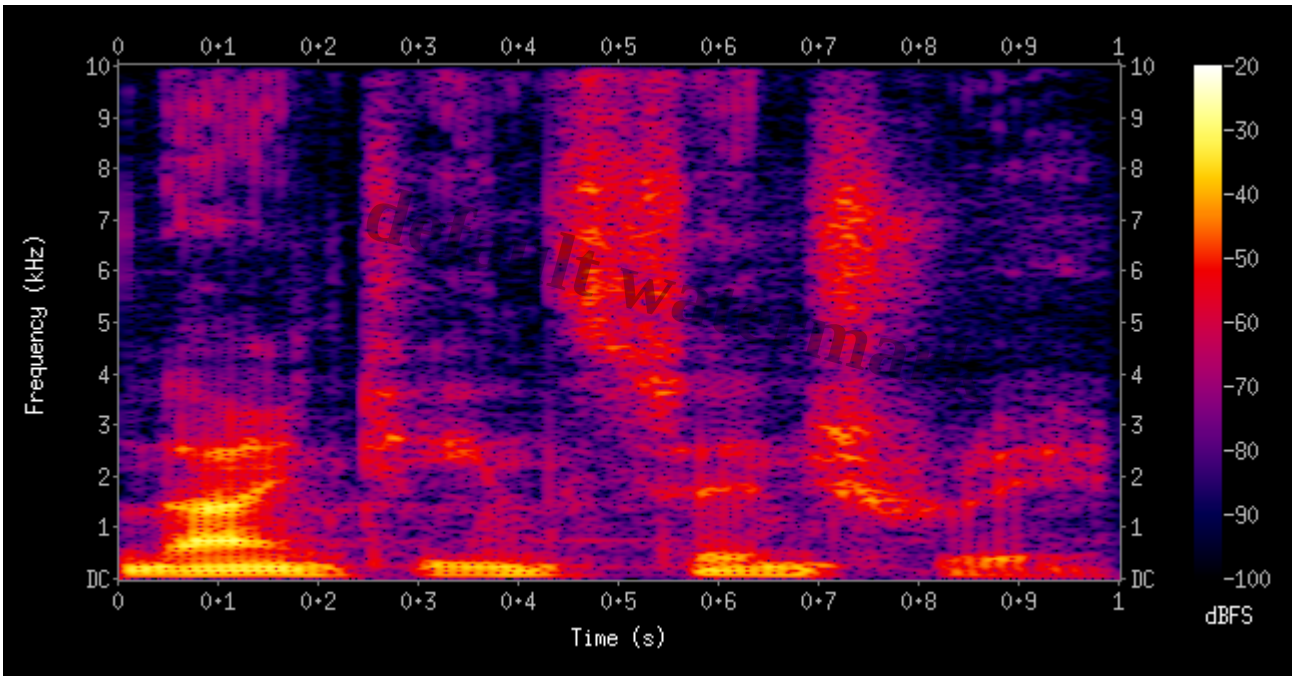
In my previous post I explored how signal processing in tandem with hidden Markov model (HMM) procedures extracts chord structures from audio music clips. Similar techniques apply to automated translation of speech audio into printed text, as in the case of automated dictation, voice activated commands, automated transcriptions, as well as automated closed captions, subtitles and translations between languages. The field is referred to as ASR: automated speech recognition.

Hereâ??s my understanding of ASR â?? much simplified. Signal processing software turns a speech audio clip into spectral information, i.e. a sequence of spectral signatures indicating frequency distributions. A human being or machine inspecting the changes in spectral distribution across a

sentence will notice discrete patterns. These patterns represent discrete *phones.* Linguists have established *phones* as the basic units of spoken language. Early in the translation procedure, text-to-speech software has to identify the sequence of *phones* in a speech audio clip.

From audio spectrogram to phonic units

I downloaded the following spectrogram from wiki-commons. The caption reads: â??Spectrogram of the spoken words â??nineteenth centuryâ??. Frequencies are shown increasing up the vertical axis, and time on the horizontal axis. The legend to the right shows that the color intensity increases with the density.â?•



Thatâ??s a graphical representation of an audio signal. Rather than a graphical image, ASR software will turn an audio signal into numbers and symbols indicating key *features*. The initial challenge of a speech recognition system is to identify discrete parts of this audio sequence that match small linguistic units, *phones*.

Linguists have textual symbols for *phones*, as appear in pronunciation guides in dictionaries, e.g. *nineteenth* is pronounced nÊ?É³nË?tiË•nÎ¸ in British English. In American English itâ??s naÉ³nË?Ë?tinÎ¸, and different again for other accents. The strange symbols represent the *phones*.

From phones to words

Automated speech recognition systems have to cope with noisy signals, variations in accent, muffled sentences, unclear diction, and variations in speed. The sound data from someone speaking is fuzzy and riddled with uncertainties. To put this in a more positive light, such systems have to deal with probabilities. What is the probability that a particular *phone* when articulated by a human speaker will show up as a particular set of spectral pattern features?

ASR systems derive such probabilities from vast numbers of examples, a training set, the results of which are coded into probability tables. Statistically, any *phone* will produce a range of possible audio features and with differing levels of certainty. Such statistical data provides sets of relative probabilities that any *phone* will show up as a particular set of spectrographic features across a range of human speakers and audio conditions.

The other data that helps translate spectrogram features to *phones* relates to likely sequences of *phones*. A /k/ sound is more likely to be followed by a /r/ or perhaps /oo/ than /p/. The different probabilities of such sequences are also derived and pre-stored from many examples of *phone* sequences. I say â??sequenceâ?•, but the data is just the probability that any phone will follow any other phone. The relationships between phones are considered two at a time.

Thatâ??s similar to the way certain music chords have different probabilities in a sequence of chords. Such relationships can be represented on a complicated network diagram as Iâ??ve shown in previous posts. These networks are stored as transition matrices (tables).

A speech-to-text system will use the probabilities in the *phone*-spectrogram relationships and the probabilities in the *phone*-to-*phone* relationships to create possible sequences of *phones* that in all probability match the speech audio clip.

default watermark

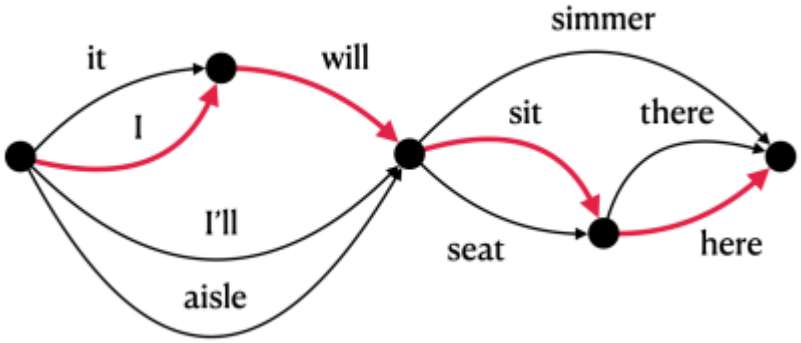
The *phone* sequences then have to match likely words. The process is similar to the spectrogram-to-*phone* translation involving training sets and sequence probabilities.

Word lattices

Once the system via its probability matrices has made a reasonable approximation of some candidate words, then the ASR system applies a similar process to word sequences. What is the probability that any particular word will show up as a particular *phone* sequence? What is the probability that a particular word will be followed by any other word?

ASR systems must derive optimal paths through lattice networks of word sequences. The system will also move between layers and select alternative *phone* sequences if the system runs out of sensible options for word sequences. Therein lies the computational challenge: how to render the massive task of iterating, checking and pruning sequences of phones and words to derive the best possible translation from audio to print and in a very short span of time â?? instantaneously.

Hereâ??s an example network (redrawn) from a technical article by Woojay, et al indicating the problem at the level of the word. An audio clip containing the sentence â??I will sit hereâ?• (in red) could sound like â??aisle seat hereâ?• or if indistinct could even sound like â??Iâ??ll simmerâ?•.



A human listener will ascertain an appropriate interpretation of ambiguous speech audio from context, even the context of the word in the sentence. â??Aisle seatâ?• is more common than â??aisle simmerâ?• and so most listeners would discount the second option. â??Will sitâ?• is more commonly uttered than â??will seat.â?• Auto-correct and predictive text while typing into a smartphone, and real-time closed captions often show this switching between alternatives as the system catches up with what you are typing or saying.

default watermark

Iâ??ve left out other layers, such as translating *phones* to phonemes, and a grammar layer. From itâ??s built in lexicons, an ASR system can tag candidate words according to whether they are nouns, verbs, adjectives, articles, pronouns, etc. An article is likely to be followed by an adjective , which may be followed by another adjective or a noun.

Updating

In automated speech synthesis, numerical probabilities drawing on multiple cues, tables, evidences and calculations at the level of the phone, phoneme, word, and grammar tags are generally sufficient to produce an acceptable translation of speech to text and at real-time speeds.

ASR systems are also designed so that lexicons and probabilities can be improved over successive generations of use, according to the speaker, and with an increased training set. The vast sets of spoken and written content distributed across telephone networks and the Internet provide further data, and in vast quantities, for ASR systems to update and improve.

Access to training data and extremely fast processors, combined with off-device processing in specialised servers, and non-real-time offline gathering and optimisation of speech data contribute to the efficiency of such systems.

The voice-activated city

Does speech-to-text technology impact the architecture of the city? ASR increases the impression that we can talk to non-sentient things. We can talk to the city, and perhaps *speak it* in new ways. ASR fuels AI narratives. The city is increasingly voice-activated. Itâ??s also listening and monitoring. Following Shoshana Zuboffâ??s critique of the digital age, itâ??s also extracting our â??behavioural surplus,â?• or at least our surplus speech acts.



References

- Jeon, Woojay, Maxwell Jordan, and Mahesh Krishnamoorthy. 2020. On Modeling Asr Word Confidence. *Cornell University*, 2 June. Available online: https://arxiv.org/abs/1907.09636v2 (accessed 15 October 2020).
- Zuboff, Shoshana. 2019. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. London: Profile Books

default watermark

Category

1. Voice and text

Tags

1. AI
2. ASR
3. automated speech recognition
4. sound
5. speech
6. text

Date Created
October 17, 2020
Author
rcoyne99