Read-Along Studio: Practical zero-shot text-speech alignment for Indigenous language audiobooks

Patrick Littell, Eric Joanis, Aidan Pine, Marc Tessier
Digital Technologies Research Centre
National Research Council Canada

David Huggins-Daines

Delasie Torkornoo
Carleton University

What is it?

A text-speech aligner (“forced aligner”)

- Automatically associates units in a text document (e.g. words) with timestamps in an audio recording.
- Intended initially to make interactive read-along audiobooks for Indigenous language literacy education (Luchian & Junker, 2004).

Zero-shot: does not require seeing ANY data in the target language in advance.

- Should work out-of-the-box on most languages.
- For languages with particularly difficult orthographies, you may need to add a G2P mapping.

Comes with a visualization WebComponent for easy embedding on the web

- Also supports a variety of academic & industry formats.

A read-along storybook in Atikamekw, courtesy of https://atikamekw.atlas-ling.ca/lecture-audio/

A sing-along karaoke video in Kitigan Zibi Anishinàbemowin

Indigenous languages spoken in Canada

There are ~70 Indigenous languages spoken in Canada

- Highly diverse; from 10 unrelated language families.
- Due to government attempts at cultural eradication, most of these languages have few fluent first-language speakers remaining (often <500), most of them elderly.

There is significant (and growing!) interest by young people and parents in Indigenous language education.

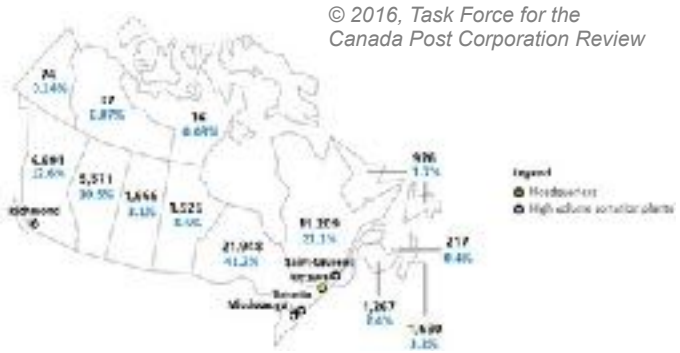
- Educational technologies remain our #1-most-requested, especially those that incorporate speech.
- Teachers have reported being overwhelmed with interest from students, and are interested in technologies to help them serve more students better.

Vastly different amounts of available digitized data

- Some communities have a lot (e.g., a 1.3M-sentence Inuktut/English parallel corpus, Joanis et al. 2018)
- Others have very little, others have some but are reluctant to share because of negative past experiences.

We want to be cautious about unconsciously gravitating towards *only* working on the 1-3 most-resourced languages.

- For those of us who are federal employees, we need to be doubly aware of unconscious biases when taking on projects.
- Part of our efforts to mitigate this bias: concentrating some of our R&D on technologies for the least-resourced languages. What technologies could feasibly be developed for *any* of these 70 languages? (Littell et al., 2018)



It'd be as if Canada Post decided they only want to do high-volume Toronto/Montreal routes.

Yes, that's the majority of postal activity!

But they have to be prepared to deliver mail elsewhere, too.

I’m always on the lookout for zero-shot techs

I want to say “Yes” here!

- But it has to be something that only requires resources your typical language organization (e.g. a school) already possesses, or can straightforwardly make.

Don’t forget the human resources!

- I’m also cautious about pitching technologies that require rare or hybrid expertise to create/maintain.

One more thing:

- It should have some use-case beyond research. E.g., helping a teacher make materials, not just making a prof’s workflow faster.

Text-speech alignment almost hits the trifecta

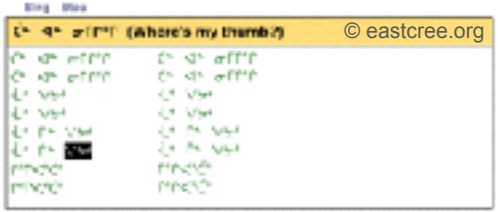
It's not just for research prep, it has a real use case in education:

- A lot of schools/publishers have literacy material in both printed and recorded forms (often, on CD). But kids aren't really checking them out anymore…
- To meet students where they are, we need to get these materials online.

Multimedia content requires timing information to coordinate parallel elements

- The more fine-grained this timing information is, the more value-added elements can be enabled.
- But finer-grained manual annotation is exponentially costly (Schiel et al., 2004). Automation is necessary to scale beyond a few proofs-of-concept.

Displaying closed-captioning at the right time



Highlighting a word when it is spoken in the recording.

© Tehahenteh Frank Miller

Letting a reader click on any word to hear that word in isolation.

© Kitigan Zibi Cultural Education Centre

Synchronizing a bouncing ball for a karaoke sing-along

From a data-prereq standpoint it's ideal!

- It doesn't need *any* pre-existing data in the target language, you can bootstrap it cross-linguistically.
- The basic idea:
  - Find an off-the-shelf alignment system (usually for English)
  - Map each sound in the target language to a rough near-neighbor in English.
  - Render all the words in the document in this pseudo-English, then align that.
- There are several ways to implement something like this.
  - We make a trivial finite-state grammar representing the pseudo-English document,
  - then force-align the audio to that grammar using an English-trained HMM-GMM (based on Huggins-Daines et al. 2006).
- We didn't invent this, it's commonly and quietly done, usually on an ad-hoc basis, on the way to doing other things.

Original document:

- `“Laƛən laχ qʼaqʼuƛʼəʼatʼsi…”`

Map between target-language orthography and English ARPABET

- `l -> L`
- `a -> AA`
- `ƛ -> T L`
- `etc...`

Resulting FSG:

But there's a surprisingly high expertise prereq

It's not that any one part of it is rare or difficult.

- But there are a lot of little things that go wrong, are under-documented, or aren't widely taught outside of specialist departments.
- It can hard to see from the inside just how much you have to know to recover from the little snags.
  - "Oh, the document needs to be Unicode NFC normalized, and also replace U+0315 COMBINING COMMA ABOVE RIGHT with U+0313 COMBINING COMMA ABOVE".
  - "Oh, this acoustic model doesn't use ARPABET 'AX' or 'NX'; use 'AH' and 'N' instead."
  - "Oh, the default beam parameters are too narrow for cross-linguistic work, make this one wider and try again."
- Sometimes these tips and tricks survive only as lore in informal social networks.
  - "Oh, I TA'd for a course where they used this system. The notes for Lecture 4 have a good beginner-level tutorial, here's a link."
  - "Oh, I know the guy who originally wrote that software. He'll know what settings to change."

ReadAlong Studio automates away the “lore”

Bit by bit, we identified the stuff that’s specialist knowledge or tricky when adapting these systems to new languages (30+ and counting).

- If the language has a G2P mapping in the G_i2P_i library (Pine et al., 2022), it uses that…
  - But if not, there’s a language-neutral one, using information from the Unicode table itself, that makes reasonable guesses for most languages.
  - There are reasonable fallbacks for OOV characters, whether they’re English names that use non-target-language letters, variant Unicode characters, etc. You can specify fallback languages, or just let the language-neutral one catch it
  - Tokenization takes into account the character inventory of the language (if known), so that phonetically-meaningful punctuation (e.g. “o:”) doesn’t accidentally split words.
- We automate nearest-neighbor mapping between target language sounds and model vocab through PanPhon (Mortensen et al 2016).
  - This also takes away the users’ need to know ARPABET, and the specific vocab used by the acoustic model, which isn’t always documented.
- Beam search parameters are set to reasonable defaults, named intuitively (e.g. “strict” rather than “1e-80”), and are progressively loosened if alignment fails.
- It actually installs and runs cross-platform, even on Windows, and it’s fast even on CPUs.

It also respects what's already there

All transformations of the data are non-destructive

- It preserves the capitalization, punctuation, formatting, structure, metadata, etc. of the original document.
- You don't have to transform the data to get it to work with the aligner, then re-associate it somehow with elements in the original document.

It respects xml:lang tags at any level (even sub-word).

- L2 education material is often multilingual!

If you know better, and do any particular step yourself, it respects what you've done

- For example, for sing-along karaoke, we don't want the default word-level tokenization, we want syllables. If I produce this markup myself, the system will respect it rather than re-do it.

How well does it work?

We tested four language/orthography combinations

- Kanyen'kéha (Mohawk)
- SENĆOŦEN (a variety of Straits Salish)
- South Qikiqtaaluk Inuktut, written in both syllabics (ᖃᑭᖅᑖᓗᒃ ᐃᓄᒃᑎᑐᑦ) and a Romanized orthography

Testing two G2P conditions to examine the question, "When it is necessary to write a language-specific G2P?"

- Using language-specific handwritten G2P systems for each language (using G_i2P_i, Pine et al., 2022)
- Using a rough language-independent "unidecode" fallback, which uses information from the Unicode table to take a guess at pronunciation.

Language	G2P	Accuracy, tolerance <N ms <10	<25	<50	<100	Span overlap F1
SENĆOŦEN	handwritten	**0.24**	**0.49**	**0.69**	**0.88**	**0.88**
	auto	0.17	0.37	0.53	0.68	0.65
Kanyen'kéha	handwritten	0.19	0.37	0.63	0.81	0.92
	auto	**0.20**	**0.42**	**0.67**	**0.85**	**0.95**
Inuktut (Syllabics)	handwritten	0.21	0.54	0.74	0.92	0.96
	auto	0.22	0.53	0.73	0.91	0.96
Inuktut (Roman)	handwritten	0.22	0.54	0.75	0.92	0.96
	auto	0.23	0.54	0.76	0.93	0.97

The gold standards aren't very large (~5 minutes each) so don't put too much stock in small differences.