Proceedings of SIGUL2022 @LREC2022, pages 23–32
Marseille, 24-25 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

ReadAlong Studio: Practical Zero-Shot Text-Speech Alignment for Indigenous Language Audiobooks

Patrick Littell[1], Eric Joanis[1], Aidan Pine[1], Marc Tessier[1], David Huggins-Daines[2], Delasie Torkornoo[3]
[1] National Research Council Canada
1200 Montreal Road, Ottawa, ON K1A 0R6
{Patrick.Littell, Eric.Joanis, Aidan.Pine, Marc.Tessier}@nrc-cnrc.gc.ca

[2] dhdaines@gmail.com

[3] Carleton University
1125 Colonel By Dr, Ottawa, ON K1S 5B6
delasie.torkornoo@carleton.ca

Abstract

While the alignment of audio recordings and text (often termed “forced alignment”) is sometimes treated as a solved problem, in practice the process of adapting an alignment system to a new, under-resourced language comes with significant challenges, requiring experience and expertise that many outside of the speech community lack. This puts otherwise “solvable” problems, like the alignment of Indigenous language audiobooks, out of reach for many real-world Indigenous language organizations. In this paper, we describe ReadAlong Studio, a suite of tools for creating and visualizing aligned audiobooks, including educational features like time-aligned highlighting, playing single words in isolation, and variable-speed playback. It is intended to be accessible to creators without an extensive background in speech or NLP, by automating or making optional many of the specialist steps in an alignment pipeline. It is well documented at a beginner-technologist level, has already been adapted to 30 languages, and can work out-of-the-box on many more languages without adaptation.

Keywords: forced alignment, text-speech alignment, Indigenous languages

1. Introduction

Despite recent advances in speech and natural language processing, many practical technologies remain out of reach for languages with few digitized resources, such as the vast majority of the roughly seventy Indigenous languages spoken in Canada (Littell et al., 2018).

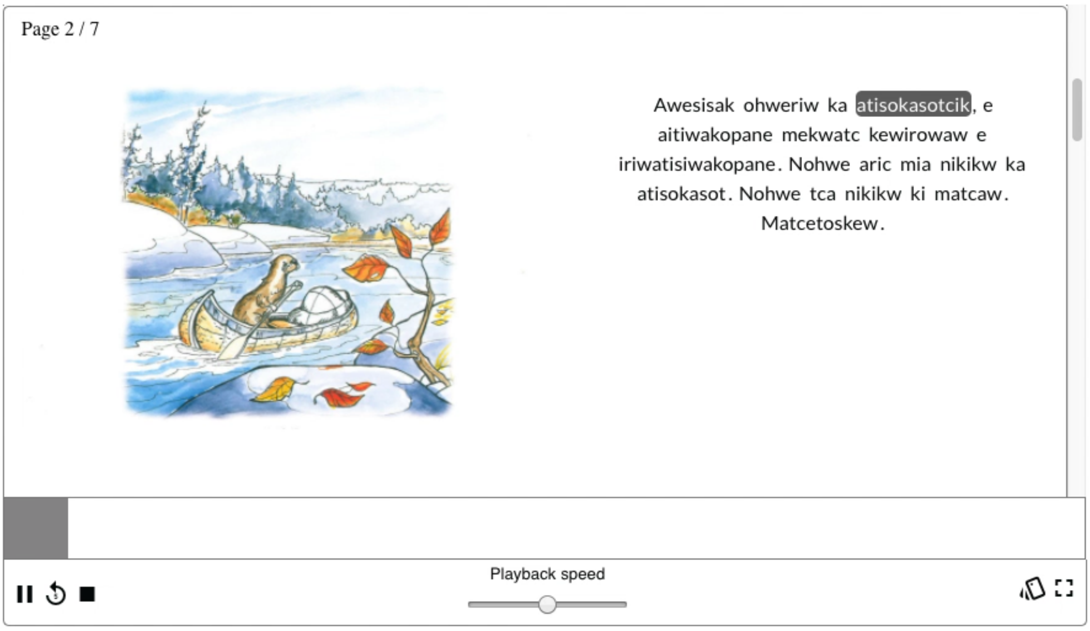
Text-speech alignment, the alignment of timestamps in a speech recording with sentences, words, or subword elements in its transcription (Robert-Ribes and Mukhtar, 1997; Moreno et al., 1998; Schiel, 1999; Yuan and Liberman, 2008; Gorman et al., 2011; McAuliffe et al., 2017), is a potential exception to this; such systems can be bootstrapped with little-to-no pre-existing data required. For example, a typical cross-linguistic alignment workflow in the Festival family of speech tools (Black et al., 1998) is to transliterate the input document into another language’s phoneme inventory (often English), and then use an off-the-shelf aligner for that language to align the transliterated document to the recording. This allows the approximate alignment of documents in a new language, even without any pre-existing training data in that language.

However, in practice, non-specialists often have trouble adapting forced-alignment workflows to new languages and speech varieties (MacKenzie and Turton, 2020). Even accomplishing the zero-data workflow described above typically requires: having access to (and installation permissions on) a UNIX workstation, understanding Unicode and handling potentially noisy user-generated inputs, coping with out-of-vocabulary tokens and code mixing, mapping phonetic near-neighbours between languages, knowing speech-specific protocols like ARPABET, setting reasonable values for beam search, etc. While these may seem minor individually, there are many potential snags to navigate, and together these skills add up to a relatively rare expertise. So while the *data* requirements of alignment are potentially quite low, the corresponding bar for *expertise* is still set rather high.

The *ReadAlongs* collaboration seeks to lower this bar to entry, so that more organizations can adapt text-speech alignment technology to their languages. ReadAlong Studio[1] is a suite of software tools for UNIX, MacOS, and Windows that automates or makes optional some specialist steps that stymie non-expert users. To give just one example here, the system uses `PanPhon` (Mortensen et al., 2016) to automate cross-linguistic approximate phone matching that, otherwise, would have required specialist intervention.

Some technological background is still recommended

[1] `https://github.com/ReadAlongs/Studio`



Page 2 / 7

Awesisak ohweriw ka atisokasotcik, e aitiwakopane mekwatc kewirowaw e iriwatisiwakopane. Nohwe aric mia nikikw ka atisokasot. Nohwe tca nikikw ki matcaw. Matcetoskew.

Figure 1: A screenshot of a web component ReadAlong published for Atikamekw. Other ReadAlongs published for Atikamekw can be found at `https://atikamekw.atlas-ling.ca/lecture-audio/`. Highlighting guides the reader to the word currently being spoken in the recording, and the reader can play single words by clicking on them.

(complete, fluent use of the tools requires some familiarity with the command line and XML), but a speech/NLP background is not.

It should be emphasized that this system, and this paper, do not present a novel *model* of forced alignment (we use a lightweight, off-the-shelf English acoustic model); we do not feel that inadequate modeling is where the main barrier lies. Rather, our approach is about automating aspects of the larger *workflow*, and this larger approach could mix and match with other approaches to the modeling problem proper.

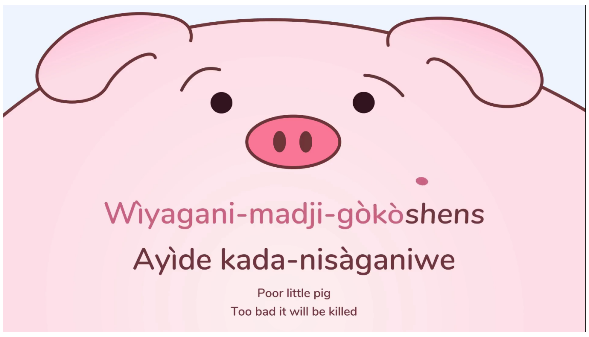
1.1. Motivation

The world's languages have vastly different amounts of digitized resources available. Among Indigenous languages spoken in Canada, for example, there are a few "medium-resourced" languages like Inuktitut, one of the official languages of the Nunavut territory, with a 1.3 million-line parallel corpus with English (Joanis et al., 2020). However, many have very limited digital resources: word lists of a few thousand words, a few hours of transcribed recordings, etc.

In light of these constraints, Littell et al. (2018) surveyed different language technologies in terms of the feasibility of developing and deploying them for *any* Indigenous languages spoken in Canada. Among these technologies, text-speech alignment stood out as a low-hanging fruit, since it can feasibly be done with no training data in the target language.

Meanwhile, the ability to align text and audio dovetailed with a real educational need. Many Indigenous language organizations (schools, publishers, etc.) already have books and other literacy materials that have been recorded by fluent speakers: often, as a printed book with an accompanying CD. However, we have heard from teachers and librarians that modern students are not necessarily using them: what kid uses a CD player these days?

Teachers need these resources to be converted into online content, which requires some level of time alignment to coordinate the different sections of the text and audio. This can be (and usually is) done manually at the page, paragraph, or sentence level, but alignment to a finer granularity can provide richer added value, like word-level highlighting and the ability to play single words by clicking them (Figure 1), or syllable-level highlighting for a sing-along karaoke video (Figure 2).



Wìyagani-madji-gòkòshens
Ayìde kada-nisàganiwe

Poor little pig
Too bad it will be killed

Figure 2: A screenshot of a bouncing-ball sing-along video in Kitigan Zibi Anishinàbemowin, made with ReadAlong Studio by aligning syllables rather than words.

In particular, we were inspired by online read-along/sing-along activities for East Cree[2] (Luchian and Junker, 2004). However, fine-grained manual alignment of text is very time-consuming, and requires a skilled annotator. Realizing that this process could be automated was the genesis of the ReadAlongs collaboration.

Upon seeing initial prototypes, the response from Indigenous language teachers and organizations has been highly enthusiastic. Teachers have mentioned to us on several occasions that their languages are traditionally oral, and that they are trying to train *speakers* and not just readers/writers, so they are always looking for ways to incorporate real speech into the curriculum. Another teacher noted that many language technologies are geared more towards advanced learners in a university-like setting, as opposed to younger students; read-along/sing-along activities are a rare language technology that even toddlers can use.

1.2. Special Considerations

Most speech/NLP libraries assume workflows where the input is being extracted and transformed, and only the transformed representations are of interest. Existing forced alignment libraries are typically conceptualized as a step in this kind of workflow, especially for the preparation of training data for speech processing or synthesis systems, or the isolation of speech segments for phonetic analysis.

It is worth highlighting some of the unspoken assumptions inherent in conventional speech pipelines:

- Documents are plain text to begin with, or structured documents have had the relevant textual material extracted.
- Formatting, capitalization, and non-phonetic material like punctuation can often be discarded as irrelevant to the downstream task.
- If a document fails to align, we can ignore it, discard the results, and move on to the next document: we do not, after all, want to train our systems or make measurements using text/audio pairs where the contents might not actually correspond.

On the other hand, for a read-along audiobook or other digital publishing product, the document in question is generally the whole point, and must be fully preserved:

- Documents have structure (pages or chapters, paragraphs, sometimes lines), formatting, capitalization, and punctuation that must be retained in the end product.
- A document that fails to align cannot be ignored or discarded; whatever is wrong with it has to be detected and fixed, whether by human or automated means.

There are also special considerations that arise due to the specific nature of our users' documents:

- English or French words (loanwords, personal and place names, etc.) occur fairly frequently. We cannot assume the document is monolingual; the software should be able to respect language annotations at any structural level (document, sentence, word), and have reasonable fallback behaviors when language tags are not used.
- Many documents for second-language learners are bilingual (e.g., where each line is accompanied by a translation), but with one of the languages not spoken in the recording.
- Conversely, the recording often has intro/outro speech that is untranscribed. In both this and the previous case, there must be some "do not align" annotation that the aligner respects, while still retaining the content in the final document.

This is not to say that existing libraries cannot be used in this context; our early versions used the Montreal Forced Aligner (McAuliffe et al., 2017) internally, although we later happened to swap it out for a more lightweight acoustic library (detailed in §2.4.6) for speed of alignment and ease of installation. However, these libraries cannot easily be used *alone* for this task, since their plain-text focus means that the original document must somehow be re-associated with the outputs or re-constructed.

Not all considerations related to the target languages introduce *greater* challenge. Most Indigenous languages, having had a shorter tradition of writing, have orthographies that are relatively transparent and organized on a phonemic basis. Grapheme-to-phoneme (G2P) transduction in these languages is often straightforward, and even rough ad-hoc G2P can suffice for many languages.

2. ReadAlong Studio

2.1. Internal Formats

In light of the above considerations, ReadAlong Studio (RAS) takes a philosophy of "non-destructive NLP": only *adding* information to a document, never transforming the document in a way where information is lost or the transformation cannot be undone.

To achieve this, RAS assumes XML-structured text internally; each step proceeds by adding elements or attributes, but leaves the text and previously-added information alone. If a more technically-advanced user has already added (say) tokenization or G2P, the system will respect it rather than overwriting it. The pipeline can be stopped at any step for advanced users to add markup by hand or by script, and restarted taking into account this markup.

RAS is usually intended for use with the ReadAlong Web Component display interface, which has a particular XML format it expects, but the aligner itself does not require this format; it could be used with a variety of XML document formats.

[2] `https://eastcree.org`

2.2. Text Standards: TEI

The intermediate XML formats, as well as the final output intended for visualization by the ReadAlong Web Component (§3.1), conform to the TEI P5 conventions for the digital humanities (TEI Consortium, 2021).

However, while the aligner should at least be able to align most TEI documents, the TEI standard is not so much a format as a collection of practices for defining a new format, specific to the sort of document one is dealing with. (That is, it is intended to allow a certain amount of interoperability and predictability whether one is working on Shakespeare folios or children's books, without requiring the scholar to coerce one sort of document into a format intended for the other.) It is *not* the case that an arbitrary TEI document will be able to be viewed in ReadAlong Web Component. We use a subset of the TEI conventions appropriate for the kinds of books our collaborators have needed to align: often children's books, but sometimes longer-form narratives for adults as well.

2.3. Alignment Standards: EPUB3/SMIL

For alignment outputs, we follow the EPUB3 e-book accessibility guidelines (Garrish et al., 2022), formerly part of the DAISY Consortium guidelines for audiobooks for the visually impaired. Rather than maintaining separate standards for plain-text books and audio-aligned accessible books, the EPUB3 standards keep the text document intact and treat aligned audio as a "media overlay" that publishers, manufacturers, and software developers can choose to support.

In the EPUB3 media overlay standards, a SMIL file (Bulterman et al., 2008) is used to express time-aligned parallelism between document elements in different kinds of media. In this case, it associates IDs within an XML document with start and end timestamps in one or more audio files. This association allows visualization software to (in one direction) drive the highlighting of text in time with accompanying media or (in the other) play snippets of media in response to the reader clicking/tapping text elements.

While the RAS library does not currently automate the creation of EPUB e-books with accessibility overlays, our compatibility with this standard means that it is fairly straightforward to convert/compile our outputs into an accessible EPUB and view it in software that supports them (e.g. Apple iBooks).

2.4. The Alignment Pipeline

2.4.1. Initial Document Generation

Although RAS uses TEI XML internally, it does not require the user to input the document in this format, and most users do not. The user can simply provide a plain-text document, and a minimal TEI document will be created from it with an appropriate structure for further processing. Additional metadata can be provided to, for example, associate images with particular pages in a picture book or mark some audio span as "do-not-align" to exclude it from the alignment process.

An advanced user can skip this step and write the XML by hand, or output it from another program, but most users let the system generate the initial XML, and (if they need more advanced features like word-level language tags or custom tokenization) modify the generated document before proceeding to subsequent steps.

2.4.2. Tokenization

If the input is not already tokenized, the system will attempt to tokenize the document at the word level.

For the purposes of RAS, "word" refers to the unit that the user wishes to align: the unit that will be highlighted in the ReadAlong Web Component, that readers can click on to hear in isolation, etc. If users have special needs with respect to this unit, they can provide these units themselves; RAS considers any material between <w> tags to be "words". For example, the singalong karaoke video in Figure 2 was made by wrapping <w> tags around syllables rather than words.

In the absence of these tags in the input, RAS will assume that word-level alignment is desired and attempt to find these units. This can be difficult given that some languages use punctuation characters phonetically (e.g., comma represents a glottal stop in SENĆOŦEN, and colon represents vowel length in Kanyen'kéha). When the character inventory of the language is known by virtue of being included in our G_i2P_i library (Pine et al., 2022), this will be taken into account, and words will not be split when the punctuation inside them can be parsed as a part of a known character.

This step will also ignore any elements tagged with an XML attribute `do-not-align`, and any elements under that element. As mentioned in §1.2, books for second-language learners often have line-by-line translations, but these are rarely spoken in the audio version; `do-not-align` attributes allow their presence in the text without the system attempting to align them.

2.4.3. ID Assignment

RAS then adds a unique XML ID attribute to each word unit. IDs are necessary because, when the document has finally been aligned, the visualizer does not just need to know that the word "the" was spoken between timestamps 32.41s and 32.65s; it needs to know *which* instance of "the" was said at that time, so it can highlight the appropriate one. In further steps (like constructing the pronunciation dictionary and finite state grammar in §2.4.6), the "words" will actually be these IDs rather than their orthographic forms.

2.4.4. Cross-Linguistic G2P

The system then performs a cross-linguistic G2P step between the target language's orthography and the phone vocabulary of the acoustic model, using the G_i2P_i library (Pine et al., 2022). In our case, the acoustic model is trained on English and thus has an English phone vocabulary, but other languages, or a multilingual model, could be used instead.

The transduction between orthographic form and model vocabulary is achieved by the composition of three transductions. First, the system performs an initial G2P from the orthographic form to the International Phonetic Alphabet (IPA). If the language is already supported in G_i2P_i, this G2P is used. At the time of writing, 30 language-specific mappings have been written: Anishinàbemowin (alq), Atikamekw (atj), Michif (crg), Southern & Northern East Cree (crj), Plains Cree (crk), Moose Cree (crm), Swampy Cree (csw), Western Highland Chatino (ctp), Danish (dan), French (fra), Gitksan (git), Scottish Gaelic (gla), Gwich'in (gwi), Hän (haa), Inuinnaqtun (ikt), Inuktitut (iku), Kaska (kkz), Kwak'wala (kwk), Raga (lml), Mi'kmaq (mic), Kanyen'kéha (moh), Anishinaabemowin (oji), Seneca (see), Tsuut'ina (srs), SENĆOŦEN (str), Upper Tanana (tau), Southern Tutchone (tce), Northern Tutchone (ttm), Tagish (tgx), and Tlingit (tli). English is also supported via the CMU Pronouncing Dictionary (Weide, 1998).

As mentioned in §1.2, there is no requirement that a document be monolingual; the G2P subsystem respects `xml:lang` attributes at any structural level. Also, if G2P fails on a word—for example, if a sentence was marked as being in the target language but it contained an unmarked English loanword with characters not in the target language—the system can fall back to a list of alternative languages provided as an XML attribute or a command-line parameter.

If no language attributes are present, the specified language is ISO 639-3 `und` (undetermined), or G2P happens to fail for the specified language and all fallback languages, the system performs a very rough automatic G2P, which we label `und`. First, the system runs the word through the `text-unidecode` library[3], which assigns each character an ASCII representation that (in most cases) roughly corresponds to its name in the Unicode table. (For example, U+12A8 ETHIOPIC SYLLABLE KA receives the ASCII representation "`ka`".) These ASCII characters are then converted to rough IPA equivalents representing cross-linguistically common usages of these characters.

While the "transcription" resulting from this would probably be inadequate for, say, text-to-speech, and would be entirely inappropriate for difficult cases like Japanese, for many of our target languages this level of rough G2P is adequate for alignment purposes. The kinds of errors that this tends to introduce are often featural (e.g., incorrect voice, glottalization, or velar vs. uvular), and would not necessarily result in different alignment outputs anyway, after the more radical transformation in the following step.

Next, the resulting IPA characters are mapped to their closest equivalents in English (or whatever language(s) the acoustic model has been trained on). This is performed automatically by `PanPhon` (Mortensen et al., 2016), a phonological knowledge base containing feature-level information about any possible human speech sound, and distance metrics between any two speech sounds. During evaluation (§4), we compare two of `PanPhon`'s distance metrics, a weighted feature edit distance and Hamming distance. It is also possible to specify a handwritten mapping, or to hand-edit the automatically generated mapping; from the point of view of the G_i2P_i library this is just another mapping to be composed with others. Finally, the resulting English IPA phones are mapped to the ARPABET vocabulary that the acoustic model expects.

2.4.5. Audio Preparation

Prior to alignment, we convert the audio file into 16-bit signed PCM (if it is not already). Also, if any timespans are marked as `do-not-align` in the user-provided metadata file, these are replaced by silences. These silences are only used for the following step; they do not affect the audio in the final read-along audiobook.

2.4.6. Alignment

For alignment, RAS uses the `SoundSwallower`[4] library, a refactored version of PocketSphinx (Huggins-Daines et al., 2006) with minimal requirements for easy installation across platforms.

It has been previously found that forced alignment at the *sentence* level does not require phonetically precise models, and in fact can be made more robust by the use of universal models estimated over broad categories of phonemes (Hoffmann and Pfister, 2013). Likewise, the context-dependent phone models typically used in large-vocabulary continuous speech recognition are equally counterproductive for alignment even at the phone level (Huggins-Daines and Rudnicky, 2006). We thus hypothesize that to produce a word-level alignment sufficient for the ReadAlongs application, the cross-linguistic G2P should be more than sufficient, and even the automatic `und` fallback should produce acceptable results in many cases.

In theory, forced alignment is quadratic in the length of the input, since every HMM state must be evaluated against every input frame in order to allow any possible alignment. This can, of course, be accelerated using beam search, at the risk of failure to align when the forced phone sequence is too divergent from the acoustic observations. However, there is another option, when state- or phone-level alignments are not needed, which is to treat alignment as a *speech recognition* task with a highly constrained grammar, accepting only the sequence of words in the input text. This allows us to perform alignment many times faster than real-time even on modest hardware, and dramatically faster than full-fledged phone-level alignment such as

[3] `https://github.com/kmike/text-unidecode/`

[4] `https://github.com/ReadAlongs/SoundSwallower`

done by the Montreal Forced Aligner. It is also possible to run the alignment code on the client side by cross-compiling it to JavaScript.

`SoundSwallower` requires (other than the input audio), two documents: a dictionary file with ARPABET pronunciations of each word (as created in §2.4.4) and a finite-state grammar representing the grammar to be recognized (in this case a trivial grammar, in which each word in the document transitions only to the following word, with 1.0 probability). Both of these (as noted in §2.4.3) use XML ID attributes as the word identifiers, so that outputs can unambiguously be re-associated with particular elements in the document.

3. Output Formats and Visualization

While the primary intended use case for RAS is the development of interactive read-along audiobooks that can be embedded in any website (§3.1), RAS's output files follow existing standards in publishing and the digital humanities that can be visualized in other ways (§3.2). It can also export to other text-audio alignment formats for a variety of use cases (§3.3).

3.1. Web Component

The primary intended downstream application for RAS is a web component[5], written in Stencil[6], that highlights words as they are spoken. Web components can be embedded in any web application for use in any browser, allowing for maximum interoperability and easy embedding in any project.

The structured XML output from RAS is interpreted by the web component such that each page element in the XML has a horizontal scrolling visual metaphor in the web component; paragraph and sentence elements have a vertical scrolling visual metaphor. Each word element becomes clickable and plays the audio for that word, allowing the reader to listen back to specific words in the document.

Deploying a ReadAlong web component involves taking the exported XML text, SMIL and audio, importing the library either with `npm` or by including the package in the HTML file in which the ReadAlong exists.

While such deployment will work for users who already have a website that they can access and edit, it requires an HTTP server to serve the assets and a developer comfortable with web hosting; it also requires that users have a stable internet connection to view the ReadAlong. To circumvent both of these problems, we also allow RAS to export to a single-file format we label "HTML", which Base64 encodes all of the fonts and assets required by the ReadAlong, and embeds them in a single HTML file that can then be used to view and share the activity offline. This allows readers without an internet connection to view it (provided they have some other means of transferring the HTML file to their computer), and removes the need for a web server, since this file is viewable in any browser without the use of an HTTP server.

3.2. Other Visualizations

Although the ReadAlong Web Component is the default visualizer assumed in our documentation, we target standard output formats (wav, XML, SMIL) that could be visualized and used in other ways. For example, as mentioned in §2.3, the formats are close enough to the EPUB3 accessibility specification that compilation into an accessible e-book is fairly straightforward. For another collaboration, we took output files aligned at the syllable level, rendered them frame-by-frame into PNG images, and then rendered those into MP4 format to make karaoke videos (Figure 2). However, video rendering is a fairly complex process, the details of which are outside of the scope of this paper.

3.3. Formats for Other Downstream Uses

A common request from academic collaborators has been support for ELAN (Brugman and Russel, 2004) and Praat TextGrid (Boersma and van Heuven, 2001) formats. RAS can produce output in these formats, so that the aligner can be used within labs' existing transcription and annotation workflows.

We also can export alignments directly to WebVTT and SRT subtitle formats to provide automatic subtitling for video content in a format compatible with YouTube.

4. Evaluation

While this is not primarily intended as a modeling paper, we performed a small evaluation to show that RAS does indeed produce reasonable outputs, and to illustrate the circumstances in which a handwritten G2P might be necessary.

4.1. Data

We manually annotated three recordings in Kanyen'kéha (Mohawk), SENĆOŦEN, and South Qikiqtaaluk Inuktut in Praat, annotating boundaries at the start and end of each word. The Kanyen'kéha recording is 5m 7s long and has 249 words, the SENĆOŦEN recording is 5m 46s long and has 419 words, and the Inuktut recording is 5m 35s long and has 282 words. Given the small size of this evaluation set, care should be taken in interpreting the results, and small differences are probably insignificant.

While both Kanyen'kéha and SENĆOŦEN use orthographies based on the Roman alphabet, they use the glyphs in very different ways, making an illustrative contrast. The Kanyen'kéha orthography is similar to a phonemic transcription of the language, using letters in much the same way as the IPA does, whereas the SENĆOŦEN orthography is entirely unique. For example, underlined W̱ represents IPA [x^w], and strikethrough Ŧ represents IPA [θ]. A pronunciation "guesser" like our `und` (see §2.4.4) would

[5] https://github.com/ReadAlongs/Web-Component

[6] https://stenciljs.com/

Language	Mapping type	Distance metric	Accuracy within tolerance (ms)				Span overlap		
			<10	<25	<50	<100	P	R	F1
SENĆOŦEN	Handmade	Weighted	0.23	0.47	0.67	0.87	0.90	0.84	0.87
		Hamming	**0.24**	**0.49**	**0.69**	**0.88**	**0.91**	**0.86**	**0.88**
	Und	Weighted	0.15	0.34	0.49	0.62	0.57	0.66	0.61
		Hamming	0.17	0.37	0.53	0.68	0.61	0.69	0.65
Kanyen'kéha	Handmade	Weighted	0.19	0.37	0.63	0.81	0.94	0.90	0.92
		Hamming	0.19	0.38	0.64	0.81	0.96	0.90	0.93
	Und	Weighted	**0.20**	**0.42**	**0.67**	**0.85**	**0.97**	**0.93**	**0.95**
		Hamming	0.19	0.39	0.64	0.82	0.97	0.91	0.94
Inuktut (Syllabics)	Handmade	Weighted	0.21	0.54	0.74	0.92	0.98	0.94	0.96
		Hamming	0.19	0.46	0.69	0.88	0.98	0.92	0.95
	Und	Weighted	0.22	0.53	0.73	0.91	0.98	0.94	0.96
		Hamming	0.20	0.48	0.71	0.89	0.98	0.94	0.96
Inuktut (Romanized)	Handmade	Weighted	0.22	0.54	0.75	0.92	0.98	0.94	0.96
		Hamming	0.19	0.49	0.70	0.89	0.98	0.92	0.95
	Und	Weighted	0.23	0.54	0.76	0.93	0.98	0.95	0.97
		Hamming	0.20	0.48	0.71	0.90	0.98	0.94	0.96

Table 1: Evaluation of SENĆOŦEN, Kanyen'kéha, and Inuktut forced alignments showing alignment accuracy of word boundaries with varying amounts of tolerance, and an F1 measurement of span overlap. Results are shown for alignments created from handmade G$_i$2P$_i$ mappings, and mappings from text-unidecode ('Und'), measured against hand-labelled alignments. The results of the best SENĆOŦEN and Kanyen'kéha systems are in bold (statistical significance is not implied), while the Inuktut results are too close to meaningfully label a best system.

not be able to guess this usage from the typical cross-linguistic usage of W and T, so SENĆOŦEN is a case where we expect a human-written G2P mapping to outperform a guessed one.

Meanwhile, the Inuktut dataset evaluates how well RAS handles a non-Roman orthography; the *qaniujaaqpait* orthography uses the Canadian Aboriginal Syllabics abugida. This same text is also available in the *qaliujaaqpait* (Romanized) orthography, letting us observe the relative performance of G2P and `und` in two different orthographies on the same recording.

4.2. Evaluation Procedure

We test two conditions for the G2P mapping from orthographic forms to language-specific IPA phones:

- **Handmade**, a hand-written mapping provided in the G$_i$2P$_i$ library.
- **Und**, the `und` fallback mapping based on the `text-unidecode` library, described in §2.4.4.

We also test two possibilities for the `PanPhon` edit distance metric, which determines which English phonemes are considered nearest neighbours to the target-language phonemes.

- **Hamming**, in which all articulatory features of each phone are weighted equally.
- **Weighted**, in which some features are weighted more highly than others, according to a phonologist's judgment of their perceptual importance.

We follow the evaluation procedure in McAuliffe et al. (2017), in which system outputs are compared for accuracy at a variety of tolerance thresholds. For example, an accuracy of 0.24 with a threshold of <10ms means that 24% of word boundaries detected were within 10ms of the human-annotated boundaries.[7]

By itself, accuracy within a fixed threshold is not clearly illustrative of whether RAS outputs are appropriate for their intended downstream task: guiding a reader through a text. This can be especially misleading when comparing languages with different word durations, or when comparing different speech styles. SENĆOŦEN typically has shorter words than Kanyen'kéha or Inuktut (in these recordings, 370ms on average compared to 769ms and 834ms, respectively); a 100ms error in SENĆOŦEN is more likely to highlight the wrong word entirely.

Therefore, we also report an F1 metric intended to capture what proportion of the time the highlighting is correctly guiding the reader, as opposed to misleading them.[8] In this metric, recall (R) represents the proportion of timespans in the reference that correctly overlap with their corresponding timespans in the system output. For example, if we were evaluating a one-word document, with a word "hello" spoken from 2.6s to 3.0s, and the system output said that word occurred from 2.8s to 3.1s, the recall would be 0.2s/0.4s = 0.50. In the other direction, precision (P) represents the pro-

[7] It should be noted that human annotations of segment boundaries vary; Schiel et al. (2004) suggest that inter-annotator agreement for phoneme-level segmentation is typically around 85–95% given a tolerance of 20ms.

[8] Many thanks to an anonymous reviewer for inspiring this line of inquiry.

portion of timespans in the system output that overlap with their corresponding timespans in the reference. Because having the highlight linger on a word during periods of silence is not misleading (indeed, it is helpful to keep the highlight on the screen even during silence), we do not penalize system timespans that extend into silences; instead, silences adjacent to the word being evaluated are ignored when calculating the precision of its alignment.

4.3. Results

Results are given in Table 1.[9] We can see that, as expected, the handwritten G2P mapping for SENĆOŦEN substantially outperformed the automatic one. On the other hand, a handwritten mapping did not outperform the automatic mapping for Kanyen'kéha; here, the automatic mapping was slightly better for all tolerances. Small differences on a small dataset should not be over-interpreted, but these results do illustrate that it is probably not necessary, in languages with cross-linguistically typical orthographies like Kanyen'kéha, to write a language-specific G2P mapping just for the purpose of approximate forced alignment.

For Inuktut, G2P and `und` performed very similarly for both orthographies, confirming that the `und` fallback can work even for non-Roman characters.

Comparison between weighted and Hamming distances did not reveal a clear winner. For SENĆOŦEN, Hamming distance performed somewhat better (especially in the poorly-performing `und` condition), but in Kanyen'kéha and Inuktut, the best systems used the weighted distance. Again, however, we should not over-interpret small differences on a small dataset.

For comparison, the Montreal Forced Aligner achieved a top score of 0.97 in the 100ms tolerance condition, in English, but this is after having been trained on approximately 1000 hours of English training data (McAuliffe et al., 2017). Our aligner has not seen *any* target-language data prior to evaluation.[10]

5. Issues and Future Work

Our early users largely agree on a central problem with the RAS workflow. When everything goes correctly and the document aligns adequately, the system seems "magical", replacing hours of human labour with a process taking seconds. However, when the document does *not* align properly, or at all, it is difficult for a novice user to know where the problem occurred (e.g., is there untranscribed text in the audio, or unspoken speech in the text?), and to fix this problem.

In early user tests, we noticed that users took a "divide-and-conquer" approach when alignment failed: dividing both the audio and text into smaller files based on obvious landmarks (like page/chapter breaks and obvious loanwords), aligning those segments separately, and then reassembling the original document. This is effective but tedious, especially when the landmark is deep within an XML structure and splitting the document means introducing matching element tags; while it may have been less labour than manual alignment, it is very frustrating labour, especially when the result of that labour still does not align!

We therefore introduced the idea of "anchors". The user can drop a custom `<anchor/>` element anywhere in the XML document, with a timestamp indicating where in the audio that anchor must be aligned, and the software will perform the division, alignment, and reassembly automatically. Anchors have made error recovery much easier; when an alignment fails or is of poor quality, the user can progressively search for landmarks and drop anchors until the alignment succeeds to their satisfaction.

This still, however, requires a basic knowledge of audio software like Audacity or Praat (to find the timestamp) and XML and text editing (to insert the anchor tag). Our next major milestone in development is a simple graphical user interface for this operation, where a user can "drag" alignments between the waveform and the text, attempt to align again, make further adjustments, etc. This sort of *human-in-the-loop* forced-alignment system, where a human and automated system negotiate the alignment of complex documents until the human is satisfied, will be a focus of future development for ReadAlong Studio.

6. Conclusion

Given the vastly different scales of available resources between languages, we are particularly interested in the "language zero-shot" frontier: what tasks can be achieved at a reasonable accuracy when a system has seen *no* data from the target language before inference?

Text-speech alignment, at least for the relatively-forgiving purpose of helping beginner readers follow along in audiobooks, is among these tasks. However, given the complexity of the pipelines and the special needs of Indigenous language audiobook alignment, it is difficult for more novice users to adapt existing forced alignment workflows to this end.

In this paper, we describe a robust text-speech alignment library that should work out-of-the-box on a variety of languages, and can be adapted via handwritten mappings for languages with more atypical orthographies. This library is open-source, comes with extensive documentation and will, we hope, help more language organizations benefit from automatic text-speech alignment.

[9] Due to fixing some bugs and addressing an issue in the reference data, our SENĆOŦEN and Kanyen'kéha results here are slightly different from those reported in Pine et al. (2022), but not in a way that affects system rankings.

[10] For an additional comparison, we performed forward-backward alignment using the Montreal Forced Aligner on these documents alone, but the systems failed to converge or produce useful alignments on such a small amount of data, so we did not report these.

7. Acknowledgements

This work would not have been possible without the many collaborators who shared their expertise and recordings, including but not limited to the Yukon Native Language Centre, the Kitigan Zibi Cultural Centre, W̱SÁNEĆ School Board, the Pirurvik Centre, Conseil de la Nation Atikamekw, Onwkawenna Kentyohkwa, Owennatekha Brian Maracle, Timothy Montler, Marie-Odile Junker, Hilaria Cruz, Nathan Thanyehténhas Brinklow, Francis Tyers, Fineen Davis, Eddie Antonio Santos, Mica Arseneau, Vasilisa Andriyanets, Christopher Cox, Bradley Ellert, Robbie Jimmerson, Shankhalika Srikanth, Sabrina Yu, Caroline Running Wolf, Michael Running Wolf, Fangyuan (Toby) Huang, Zachery Hindley, Darrel Schreiner, Luyi Xiao, Siqi Chen, Kwok Keung Chung, Koon Kit Kong, He Yang, Yuzhe Shen, Rui Wang, Zirui Wang, Xuehan Yi, and Zhenjie Zhou.

8. Bibliographical References

Black, A. W., Taylor, P., and Caley, R. (1998). The Festival speech synthesis system. http://www.festvox.org/festival.

Boersma, P. and van Heuven, V. (2001). Speak and unSpeak with PRAAT. *Glot International*, 5(9/10):341–347, December.

Brugman, H. and Russel, A. (2004). Annotating multi-media/multi-modal resources with ELAN. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Bulterman, D., Jansen, J., Cesar, P., Mullender, S., Hyche, E., DeMeglio, M., Quint, J., Kawamura, H., Weck, D., García Pañeda, X., Melendi, D., Cruz-Lara, S., Hanclik, M., Zucker, D. F., and Michel, T. (2008). *Synchronized Multimedia Integration Language (SMIL 3.0)*. W3C Recommendation.

Garrish, M., Kerscher, G., LaPierre, C., Pellegrino, G., and Singh, A. (2022). *EPUB Accessibility 1.1 Conformance and Discoverability Requirements for EPUB Publications*. W3C Working Draft.

Gorman, K., Howell, J., and Wagner, M. (2011). Prosodylab-aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

Hoffmann, S. and Pfister, B. (2013). Text-to-speech alignment of long recordings using universal phone models. In Frédéric Bimbot, et al., editors, *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, pages 1520–1524. ISCA.

Huggins-Daines, D. and Rudnicky, A. I. (2006). A constrained Baum-Welch algorithm for improved phoneme segmentation and efficient training. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*. ISCA.

Huggins-Daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., and Rudnicky, A. I. (2006). PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE.

Joanis, E., Knowles, R., Kuhn, R., Larkin, S., Littell, P., Lo, C.-k., Stewart, D., and Micher, J. (2020). The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France, May. European Language Resources Association.

Littell, P., Kazantseva, A., Kuhn, R., Pine, A., Arppe, A., Cox, C., and Junker, M.-O. (2018). Indigenous language technologies in Canada: Assessment, challenges, and successes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2620–2632, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Luchian, R. and Junker, M.-O. (2004). Developing an on-line Cree read-along with syllabics. *Carleton University Cognitive Science Technical Report*, 2006-01.

MacKenzie, L. and Turton, D. (2020). Assessing the accuracy of existing forced alignment software on varieties of British English. *Linguistics Vanguard*, 6(s1):20180061.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In *Interspeech 2017*, pages 498–502. ISCA, August.

Moreno, P. J., Joerg, C., Thong, J.-M. V., and Glickman, O. (1998). A recursive algorithm for the forced alignment of very long audio segments. In *International Conference on Spoken Language Processing, vol. 8*.

Mortensen, D. R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., and Levin, L. S. (2016). PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

Pine, A., Littell, P., Joanis, E., Huggins-Daines, D., Cox, C., Davis, F., Santos, E. A., Srikanth, S., Torkornoo, D., and Yu, S. (2022). G$_i$2P$_i$: Rule-based, index-preserving grapheme-to-phoneme transformations. In *Proceedings of The 5th Workshop on The Use of Computational Methods in the Study of Endangered Languages*.

Robert-Ribes, J. and Mukhtar, R. (1997). Automatic generation of hyperlinks between audio and transcript. In *Eurospeech*.

Schiel, F., Draxler, C., Baumann, A., Elbogen, T., and Steen, A. (2004). The production of speech corpora. `https://www.bas.uni-muenchen.de/Forschung/BITS/TP1/Cookbook/`.

Schiel, F. (1999). Automatic phonetic transcription of nonprompted speech. In *Proc. of the ICPhS*, pages 607–610.

TEI Consortium. (2021). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. TEI Consortium.

Weide, R. (1998). The Carnegie Mellon pronouncing dictionary. `www.speech.cs.cmu.edu/cgi-bin/cmudict`.

Yuan, J. and Liberman, M. (2008). Speaker identification on the SCOTUS corpus. *Proceedings of Acoustics 2008*, pages 5687–5690.