Autophon user guide
English – North America

Engine: Montreal Forced Aligner 2.0
Model: English language model v2.0.0a

1 Introducing Autophon and Forced Alignment

Autophon is a free, user-friendly tool for phoneticians that performs *forced alignment* (FA) — the automated process of converting speech recordings and their transcriptions into phonetically time-stamped annotations.

Autophon leverages widely used alignment engines developed by the phonetics community, including:

- *FAVE*[1]
- *faseAlign*[2]
- *Montreal Forced Aligner, version 1.0*[3]
- *Montreal Forced Aligner, version 2.0*[3]

The tool produces time-aligned phonetic annotations compatible with Praat[4], based on two user inputs: (1) a speech recording and (2) its orthographic transcript.

This user guide is specifically for ***English***, using the ***Montreal Forced Aligner 2.0*** engine with the ***English language model v2.0.0a*** model. Autophon may support additional engine-model combinations for this language; therefore, ensure you are using the best option for your needs.

While many forced aligners exist, they often require command-line usage and are tied to outdated or incompatible operating systems. **Autophon offers a platform-independent, intuitive alternative for phoneticians worldwide**.

2 Using the app

2.1 Aligning files without registering To align smaller files, go to the main page and click *Add files* at the bottom. A box titled *Transcription Mode: change transcription mode* will appear. Click the heading to choose one of four *Transcription Modes* (see below), then select your files.

2.2 Registering and logging in To align larger files or access the full suite, click *Sign up* to create a free account. This helps us monitor usage for funders and guard against bots. After signing up, check your email for a verification link. If it doesn't arrive, check your spambox and wait 15 minutes before contacting tech support.

2.3 Cost Autophon is free of charge.

2.4 Aligning files in a registered account Once registered and verified, go to the *Aligner* tab and click *Add files*. A box titled *Transcription Mode: change transcription mode* will appear. Click the heading to choose one of four *Transcription Modes*, then select your files.

2.5 Transcription modes Autophon supports four *Transcription Modes*, named for the fields they're commonly used in: *Experimental Linguistics A*, *Experimental Linguistics B*, *Computational Linguistics*, and *Variationist Linguistics*. Each mode can be selected via the corresponding box in Figure 1, which illustrates expected file structures and links to instructional videos.

Video instructions for each transcription mode can be viewed. In addition, sample templates for each mode are available for **download here**.

[1] FAVE was built by Rosenfelder, Fruehwald, Brickhouse, Evanini, Seyfarth, Gorman, Prichard, and Yuan (2022). It relies on the Hidden Markov Toolkit (Young, Woodland, and Byrne 1993).

[2] faseAlign was built by Wilbanks (2022). Like FAVE, it relies on the Hidden Markov Toolkit (Young, Woodland, and Byrne 1993).

[3] The Montreal Forced Aligner was developed by McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger (2017). It uses the Kaldi toolkit (Povey, Ghoshal, Boulianne, Burget, Glembek, Goel, Hannemann, Motlicek, Qian, Schwarz, et al. 2011).

[4] Praat is a speech analysis tool developed by Boersma and Weenink (2017).

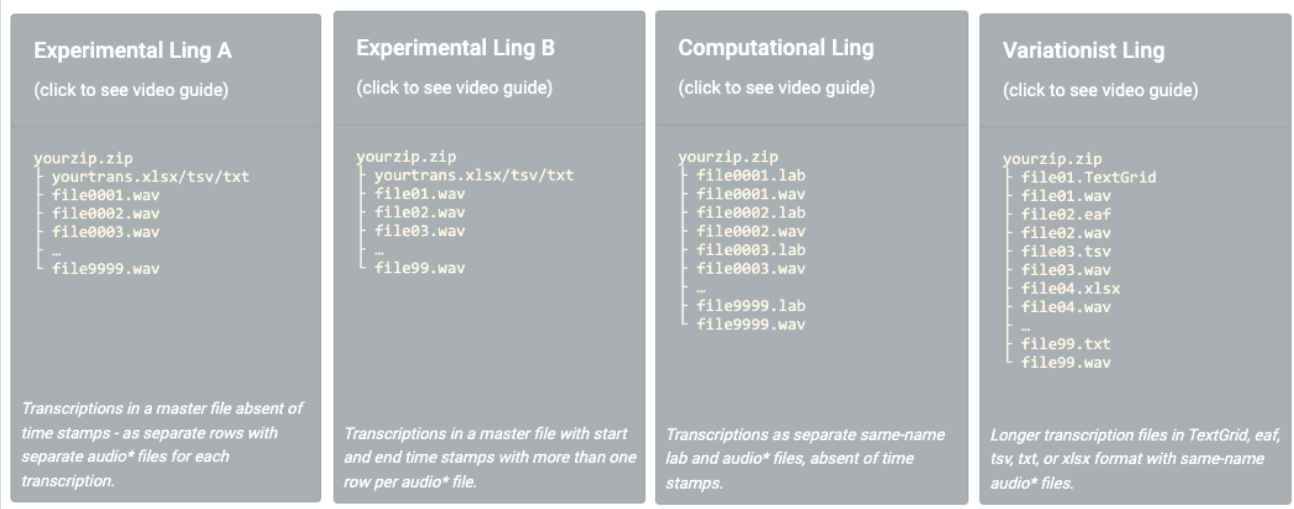


Figure 1: The Transcription Mode selection menu for Autophon.

Figure 2: Autophon outputs finished TextGrids using the same subfolder structure as the uploaded files.

Experimental linguistics A: Upload a two-column spreadsheet (Excel `xlsx`, or tab-delimited `txt`/`tsv`) with audio filenames in column 1 and transcriptions in column 2. No time stamps allowed. This format suits short clips and resembles CommonVoice[5]. Use zip or individual file upload.

Experimental linguistics B: Same structure as A, but with four columns: audio file name, start time, end time, and transcription. Designed for longer recordings requiring segmentation. Time stamps must be in real-number seconds (e.g., `1.23` or `1,23`); no colons or hour-minute markers are permitted (e.g., you may not use something like `00:00:01.23`).

Computational Linguistics: Upload matching audio and `lab` files (containing only the corresponding transcription). Files may be zipped with nested folders—Autophon preserves the hierarchy (Figure 2). No time stamps permitted.

Variationist Linguistics:[6] Upload paired transcription and audio files (individually or zipped). Transcriptions may be in Praat `TextGrid`, ELAN `eaf`, or tabular format (`xlsx`, `txt`, `tsv`). Use either three or four columns:

- Four-column: speaker, start time, end time, transcription
- Three-column: start time, end time, transcription

Time stamps must be real-number seconds (comma or period decimal separators); formats with colons (e.g., `00:00:01.23`) are not supported.

2.6 File formats and codecs

If you encounter errors during upload, it's often due to unsupported file formats or *codecs*. The simplest fix is to re-save your files in a common format using tools like Praat or ELAN.

Transcription file formats: Autophon accepts transcription files in most standard encodings, including `UTF-8` and `UTF-16 (Windows CRLF)`. If you encounter issues, try re-saving the file or email a sample to tech support.

Audio file formats: Autophon supports a wide range of audio formats, including: `WAV`, `FLAC`, `MP3`, and more. Stereo files are not currently accepted. Therefore, convert all audio to mono first.

2.7 Transcription preparation

Regardless of the transcription mode, each entry should contain between one and 20 words. Boundary demarcations must include at least 0.01 seconds of silence before and after the speech. Figure 3 shows a five-word phrase with a 0.03-second pre-boundary and a 0.25-second post-boundary. This sort of variability is expected and handled well by Autophon.[7]

2.8 Select a language

After uploading files into the aligner, Autophon will suggest a language and language model. You may override this suggestion using the dropdown menu.

2.9 Task list

The task list displays all uploads along with file name, upload date, language, tier count, file size, word count, and an inventory of missing words. You can delete the task and start over, add words to your *custom pronunciations* box (described below), or proceed by clicking *Align*.

2.10 Missing words

To understand this feature, it helps to know that forced alignment maps phonemic pronunciations – defined in language-specific dictionaries – onto the speech stream using statistical models. These dictionaries contain a finite set of words. The *missing words* feature lists items not found in Autophon's dictionary and provides suggested pronunciations. Autophon will use these suggestions by default, but you can reject them and enter your own. The next section explains how.

2.11 Your custom pronunciations

If you disagree with either (a) Autophon's pronunciation suggestions for missing words or (b) the default dictionary entries, you can override them here. Enter your own phonemic transcriptions in this box, which will take precedence over both.

Pronunciations must be entered using the alphanumeric string specific to the language model at hand – in this case, the ***ARPAbet***. Section 4 provides a key that maps the ARPAbet to its IPA[8] equivalents.

[5] https://commonvoice.mozilla.org

[6] This field originally drove the development of forced alignment in the early 2000s.

[7] If your transcriptions are segmented with *exact* start and end times, performance may degrade and boundary shifts may occur. If you're working with such data, contact tech support—we are interested in designing a fifth transcription mode for these cases.

[8] International Phonetic Alphabet

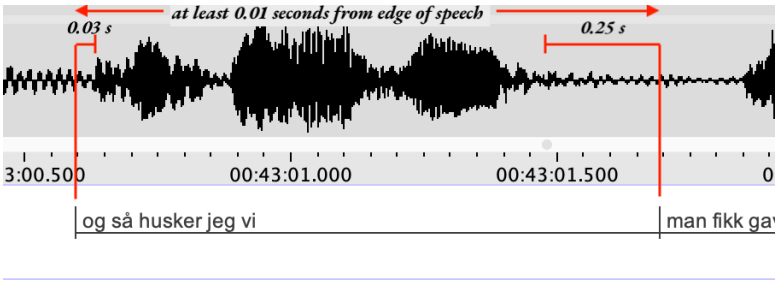


Figure 3: A sample transcription with at least 0.01 seconds of buffer on either end of the speech stream.

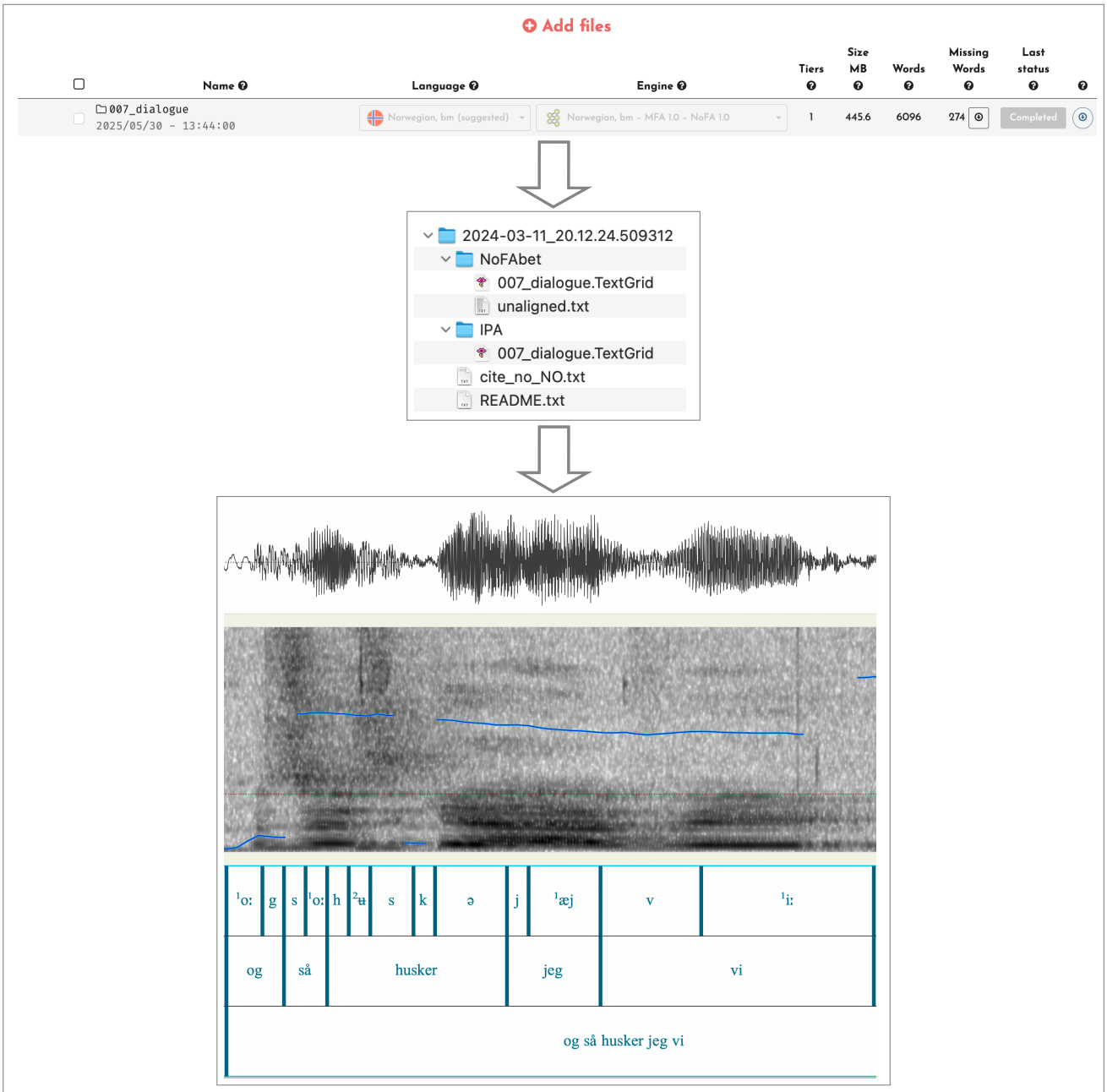


Figure 4: The alignment process, including task list, folder structure, and Praat TextGrid.

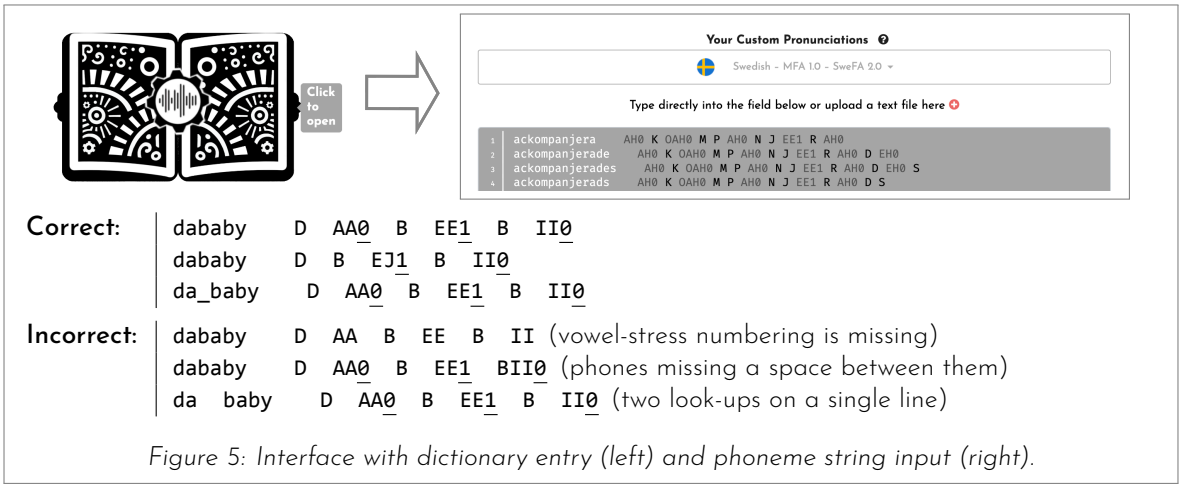
You can enter pronunciations directly in the dictionary box or upload them from a `txt` file. The maximum input length is 50 000 characters.

Entries must follow the format:

- `word[space]phoneme[space]phoneme` OR `word[tab]phoneme[space]phoneme`

Each phoneme must be separated by a space, and the lookup may not include two or more words — Autophon will interpret the second word as a phone and produce an error. You may submit multiple pronunciations for the same word by repeating the word on separate lines with different phoneme strings. Autophon will evaluate the best match for each speech instance. Refer to Figure 5 and the examples below.

Stress and/or accent ***must*** accompany every vowel and diphthong with a number. Consult Figure 1 for the specific digits used in this model; refer to Figure 5 to see how these are operationalized.



Correct:

```
dababy    D  AA0  B  EE1  B  II0
dababy    D  B  EJ1  B  II0
da_baby    D  AA0  B  EE1  B  II0
```

Incorrect:

```
dababy    D  AA  B  EE  B  II (vowel-stress numbering is missing)
dababy    D  AA0  B  EE1  BII0 (phones missing a space between them)
da  baby    D  AA0  B  EE1  B  II0 (two look-ups on a single line)
```

Figure 5: Interface with dictionary entry (left) and phoneme string input (right).

2.12 Aligning files To begin alignment, click *Align* to the far right of the upload list. Alignment typically takes a few minutes, depending on server load.

2.13 Downloading the annotations When alignment is complete, you can download the annotations as Praat TextGrids by clicking the downward arrow beside the task list. See Figure 4 for an illustration.

3 How to cite

Any dissemination or publication that makes use of this Autophon package for ***English – North America***, which uses ***the English language model v2.0.0a*** within ***The Montreal Forced Aligner 2.0*** for its engine, should cite the relevant references listed below. Proper citation is essential: not only to acknowledge the "daisy chain" of technical and academic work underpinning Autophon, but also to reinforce the incentives for sharing tools with the broader community.

While space constraints may tempt you to remove references to software, we strongly encourage prioritizing these citations. If trimming is necessary, consider reducing peripheral citations in the literature review instead.

Boersma, P., & Weenink, D. (2017). Praat: Doing phonetics by computer [softw.], ver.6.0.36. www.praat.org

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. *Proceedings of Interspeech*, 498–502.

McAuliffe, M., & Sonderegger, M. (2022, May). *English language model v2.0.0a* (tech. rep.). https://mfa-models.readthedocs.io/en/latest/language_model/English/English%20language%20model%20v2_0_0a.html#English%20language%20model%20v2_0_0a

Young, N. J., & Anikwe, K. H. (2025). Autophon – Automatic phonetic annotation and online forced aligner. www.autophon.org

4 Phoneme key

Autophon will output two versions of the same TextGrid for every file you align: (1) a TextGrid in the ARPAbet specific to the English language model v2.0.0a for The Montreal Forced Aligner 2.0 and (2) a TextGrid in the International Phonetic Alphabet (IPA). The phoneme key is located in Table 1.

ARPAbet	IPA	example	ARPAbet	IPA	example	ARPAbet	IPA	example	ARPAbet	IPA	example
Vowels			***Diphthongs***			HH	h	*high*	V	v	*vie*
AA	ɑ	*father*	AW	aʊ	*bout*	JH	dʒ	*jive*	W	w	*wise*
AE	æ	*bat*	AY	aɪ	*bite*	K	k	*kite*	Y	j	*yacht*
AH	ʌ	*butt*	EY	eɪ	*bait*	L	l	*lie*	Z	z	*zoo*
AO	ɔ	*caught*	OY	ɔɪ	*boy*	M	m	*my*	ZH	ʒ	*pleasure*
EH	ɛ	*bet*	***Consonants***			N	n	*nigh*	***Syllabic consonants***		
IH	ɪ	*bit*	B	b	*buy*	NG	ŋ	*sing*	ER	ɝ	*bird*
IY	i	*beat*	CH	tʃ	*cheers*	P	p	*pie*	***Lexical stress***		
OW	oʊ	*boat*	D	d	*die*	R	ɹ	*rye*	○0	○	*banana*
UH	ʊ	*book*	DH	ð	*then*	S	s	*sigh*	○1	ˈ○	*banana*
UW	u	*boot*	F	f	*fight*	SH	ʃ	*shy*	○2	ˌ○	*barnyard*
			G	g	*guy*	T	t	*tie*			
						TH	θ	*thigh*			

Table 1: ARPAbet, IPA, and lexical examples. The prosodic denotation means that any ARPAbet vowel, diphthong or syllabic consonant (eg., ER*) must* ***always*** *be followed by the numbers* 1 *(primary stress),* 2 *(secondary stress), or* 0 *(unstressed).*

This phoneme inventory was taken from The CMU Pronouncing Dictionary[9]. Every ARPAbet vowel, diphthong and syllabic consonant is followed by a numerical code that denotes suprasegmental information. ○0 refers to lexically unstressed vowels; ○1 – primary lexical stress; ○2 – secondary lexical stress.

5 Acoustic model and pronunciation dictionary

This specific Autophon package for ***English – North America*** uses the English language model v2.0.0a within The Montreal Forced Aligner 2.0, which was trained on American English. The pronunciation dictionary is the The CMU Pronouncing Dictionary[10].

6 Performance metrics

We currently have no performance metrics for the English language model v2.0.0a within The Montreal Forced Aligner 2.0. However, they are available for Montreal Forced Aligner 1.0 and can be accessed on page 501 in McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger (2017) (attached as an Appendix here). If you are willing to provide us with a set of manually-corrected TextGrids in this language, we would be eager to validate our model with them (and update this document accordingly).

7 Data security and GDPR compliance

Files uploaded to Autophon are encrypted and transmitted to a secure server hosted by *Digital Ocean* within the European Union (Frankfurt and Amsterdam). Transcriptions and audio files are automatically deleted immediately after alignment. This approach enhances privacy while also reducing storage costs. By contrast, finished TextGrids remain available in your account until you choose to delete them. Once deleted, they are permanently removed from our servers.

[9] http://www.speech.cs.cmu.edu/cgi-bin/cmudict
[10] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

If you upload files but do not initiate alignment by clicking *Align*, the files will be automatically purged at 3:00 AM GMT[11].

Autophon adheres to the principles of the European Union's General Data Protection Regulation (GDPR). We collect only four pieces of user information: name, title, affiliation, and email address. Once a file is aligned, the corresponding audio is permanently deleted. Deleting a file from your task list also permanently removes the transcription and filename metadata. You may delete your account at any time, which will erase all associated personal data. However, we **do** retain anonymized alignment metadata — such as a randomly assigned alphanumeric user ID and summary usage statistics — to demonstrate the platform's utility to funders.

8 Features and limitations

What Autophon is: Autophon is a web-based application designed to simplify forced alignment workflows and expand access for users with minimal technical background. It is particularly useful for research on under-resourced languages and non-standard varieties, and emphasizes ease of use, format flexibility, and language model diversity. The backend relies on existing forced alignment technologies developed over the past decades, wrapped in a modern frontend that facilitates fast, OS-independent processing.

Key features include:

1. Fully web-based and platform-independent (OS-agnostic).
2. No programming or installation required.
3. Accepts a wide range of transcription and audio formats.
4. Capable of processing low-resource and non-standard language varieties.
5. Supports user-defined pronunciation dictionaries and multiple transcription modes.

What Autophon is not: Important caveats to bear in mind:

1. Alignment quality depends on transcription accuracy, recording quality, and language characteristics.
2. Performance may vary across languages, dialects, and speaking styles.
3. Benchmarking accuracy is ongoing and not available for all models.
4. Core updates to underlying alignment engines may not be immediately reflected, due to the complexity of the Autophon backend.

9 Budget and funding

Autophon costs approximately 25 000 SEK (2 300 EUR) per year to run. Founded by Dr. Nate Young (who is the sole copyright holder), the project has since received support from the University of Helsinki, Linnaeus University, The Swedish Academy, the Department of Linguistics and Scandinavian Studies at the University of Oslo, and the European Union's *Horizon 2020* research and innovation programme under the *Marie Skłodowska-Curie* grant agreement No. 892963. Additional funding for language model development has come from The National Library of Norway[12].

We continue to seek funding and welcome collaboration. If you are experienced in grant writing or interested in supporting the project, please reach out via the support page.

Acknowledgements

Numerous individuals have contributed to Autophon. We especially thank Michael McGarrah for strategic guidance and Kaosi Anikwe for extensive backend and frontend development. Ismail Raji Damilola helped implement a bootstrapping function to expand phoneme inventories. Additional contributions in the early stages came from Nabil Al Nazi, Zamanat Abbas Naqvi, and Santiago Recoba.

[11] Users working near this cutoff time—e.g., at 2:55 AM GMT—should be aware that their files may disappear if alignment is not initiated in time.

[12] https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-59/

References

Boersma, P., & Weenink, D. (2017). Praat: Doing phonetics by computer [softw.], ver.6.0.36. www.praat.org

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. *Proceedings of Interspeech*, 498–502.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). *The Kaldi speech recognition toolkit* (tech. rep.). IEEE Signal Processing Society. Piscataway.

Rosenfelder, I., Fruehwald, J., Brickhouse, C., Evanini, K., Seyfarth, S., Gorman, K., Prichard, H., & Yuan, J. (2022). FAVE (Forced Alignment and Vowel Extraction) Program Suite v2.0.0 [Zenodo].

Wilbanks, E. (2022). faseAlign (Version 1.1.14). https://github.com/EricWilbanks/faseAlign

Young, S. J., Woodland, P. C., & Byrne, W. J. (1993). *HTK: Hidden Markov Model Toolkit V1. 5*. Cambridge Univ. Eng. Dept. Speech Group; Entropic Research Lab. Inc.

INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Montreal Forced Aligner: trainable text-speech alignment using Kaldi

Michael McAuliffe[1], *Michaela Socolof*[2], *Sarah Mihuc*[1], *Michael Wagner*[1,3], *Morgan Sonderegger*[1,3]

[1]Department of Linguistics, McGill University, Canada
[2]Department of Linguistics, University of Maryland, USA
[3]Centre for Research on Brain, Language, and Music, McGill University, Canada

michael.mcauliffe@mail.mcgill.ca, msocolof@umd.edu, sarah.mihuc@mail.mcgill.ca,
chael@mcgill.ca, morgan.sonderegger@mcgill.ca

Abstract

We present the Montreal Forced Aligner (MFA), a new open-source system for speech-text alignment. MFA is an update to the Prosodylab-Aligner, and maintains its key functionality of trainability on new data, as well as incorporating improved architecture (triphone acoustic models and speaker adaptation), and other features. MFA uses Kaldi instead of HTK, allowing MFA to be distributed as a stand-alone package, and to exploit parallel processing for computationally-intensive training and scaling to larger datasets. We evaluate MFA's performance on aligning word and phone boundaries in English conversational and laboratory speech, relative to human-annotated boundaries, focusing on the effects of aligner architecture and training on the data to be aligned. MFA performs well relative to two existing open-source aligners with simpler architecture (Prosodylab-Aligner and FAVE), and both its improved architecture and training on data to be aligned generally result in more accurate boundaries.

Index Terms: forced alignment, automatic segmentation, acoustic analysis

1. Introduction

In *forced alignment*, speech and its corresponding orthographic transcription are automatically aligned at the word and phone level, given a way to map graphemes to phonemes (typically a pronunciation lexicon) and a statistical model of how phones are realized. Forced alignment has become widely used in scientific research on language over the past ∼10 years, including in sociolinguistics, phonetics, language documentation, and psycholinguistics (e.g. [1, 2, 3, 4, 5]). This use has been driven by the availability of accurate, pre-built, and easily usable aligners, such as FAVE/P2FA, (Web)MAUS, and Prosodylab-Aligner [6, 7, 8]. We focus on this broad use case: forced alignment for language sciences using publicly-available software, when at least an orthographic transcript is available.[1]

Many such forced aligners now exist (e.g. [6, 7, 8, 12, 13, 14, 15, 16, 17]), which differ in two key ways. First, in *architecture*, including the acoustic model used to model the realization of phones, and whether the acoustic features are transformed to account for speaker variability. Second, in *trainability*: most aligners ship with pre-trained acoustic models only, while others can be retrained on new data [8, 17].

We describe the Montreal Forced Aligner (MFA), new open-source forced alignment software which is a successor to the Prosodylab-Aligner. MFA maintains Prosodylab-Aligner's trainability and updates its architecture. MFA uses triphone acoustic models to capture contextual variability in phone realization, in contrast to monophone acoustic models used in Prosodylab-Aligner and other current aligners (e.g. FAVE). MFA also includes speaker adaptation of acoustic features to model interspeaker differences. MFA uses the Kaldi speech recognition toolkit [18], which offers advantages over the HTK toolkit underlying most existing aligners.

We evaluate MFA's performance on detecting word and phone boundaries in laboratory and conversational speech. Our experiments test whether the more complex architecture and trainability of MFA affect performance, by comparing to two existing monophone acoustic model aligners and varying the training data.

2. Montreal Forced Aligner

MFA is an open-source command line utility, with prebuilt executables for Windows and Mac OSX, and online documentation.[2] MFA is built on top of Kaldi, an actively maintained, open-source automatic speech recognition toolkit [18], and has three key usability features: it builds on the *trainability* of Prosodylab-Aligner, and improves *portability* and *scalability*. The use of Kaldi as the ASR toolkit rather than HTK allows for easier distribution due to Kaldi's more permissive license, so no compilation from source is required by the user. MFA's use of Kaldi is highly parallel, which mitigates run time when using larger corpora and more computationally-intensive training.

The ASR pipeline that MFA implements uses a standard GMM/HMM architecture, adapted from existing Kaldi recipes. To train a model, monophone GMMs are first iteratively trained and used to generate a basic alignment. Triphone GMMs are then trained to take surrounding phonetic context into account, along with clustering of triphones to combat sparsity. The triphone models are used to generate alignments, which are then used for learning acoustic feature transforms on a per-speaker basis, in order to make the models more applicable to speakers in other datasets [19]. MFA has been successfully applied to 29 languages from GlobalPhone [20], the NCHLT corpora of South African languages [21], and other corpora.

MFA uses mel-frequency cepstral coefficients (MFCCs) as acoustic features. Thirteen MFCCs are calculated with a 25 ms window size and 10 ms frame shift. The feature calculation has a frequency ceiling of 8 kHz, allowing for acoustic models to be built and used regardless of sampling rate (i.e., models trained on 16 kHz sampled files can be applied to 44.1 kHz sampled files without manual resampling). Delta and delta-delta features from surrounding MFCC frames are also included, giving

[1]We do not address related work, such as on linguistic analysis of untranscribed speech [9], or phoneme boundary detection [10], or text-speech alignment for TTS [11].

[2]https://montrealcorpustools.github.io/Montreal-Forced-Aligner/

http://dx.doi.org/10.21437/Interspeech.2017-1386

39 features per frame. Following MFCC generation, CMVN is applied to the features on a per-speaker basis to increase robustness to speaker variability. In the final round of training, feature transforms for each speaker are estimated using feature space Maximum Likelihood Linear Regression (fMLLR) [19]. Speaker adaptation is also done when aligning using pre-trained models, but can be disabled for faster alignment.

During training, MFA does 40 iterations of monophone GMM training, with realignment done during 20 of the iterations. Following monophone training, 35 iterations of triphone training are done, with 15 iterations that perform realignment. Speaker-adapted triphone training includes another 35 iterations with 15 realignment iterations, as well as 5 iterations that include fMLLR estimation. Multiprocessing is used extensively during feature calculation and training, allowing MFA to handle training and alignment of large corpora. For instance, the 1000-hour LibriSpeech corpus was aligned in 80 hours (on a desktop using 12 3.4-Ghz processors, 32 GB memory), and training from scratch on the 20-hour Buckeye corpus (Sec. 3) took 2 hours (on a laptop using 4 2.5-GHz processors, 8GB memory).

MFA ships with a pre-trained model for English that has been trained on the LibriSpeech corpus [22] ($\sim$1000 hours of audiobooks), and pre-trained acoustic models (mostly from GlobalPhone corpora [20]) and grapheme-phoneme models for generating pronunciation dictionaries are publicly available in the online documentation for 20+ languages. A key feature of MFA is trainability of acoustic models on new data, as in the Prosodylab-Aligner [8]. Thus, a user can align their dataset either using pre-trained models, or by training from scratch on the dataset. Alignment can be significantly better when using acoustic models trained from scratch—especially when the dataset to be aligned is sufficiently large and varied. We recommend experimenting with pre-trained models and retraining, as it is an empirical question which method gives better alignments.[3] The experiments in Section 3 address this question.

There are two primary transcription formats used in current forced aligners, exemplified by Prosodylab-Aligner and FAVE. Prosodylab-Aligner aligns short wav files, each with an associated text file specifying the transcription. This format is common to lab speech where individual trials keep speech segments naturally short. FAVE aligns long files containing time-aligned periods of transcribed speech, a format more common to sociolinguistic data and spontaneous speech. MFA supports both formats, building on the Prosodylab-Aligner format and adding support for Praat [23] TextGrids as a way to specify transcriptions in longer sound files. The TextGrid format allows for the user to specify transcriptions for multiple speakers in the same file. The output of alignment is then a TextGrid for each input file, with separate word and phone tiers for each speaker.

MFA contains other upgrades to the Prosodylab-Aligner. Instead of requiring every word in the transcripts to be in the pronunciation dictionary, MFA includes an explicit model for unknown words as having a unique phone, which allows them to be modeled while maintaining alignment of surrounding words. The unknown word's phone is constructed similarly to the silence phone, and can match any amount of vocal noise or speech (e.g. words of different lengths). Before performing alignment, MFA prompts the user if unknown words are found, including their location, to deal with simple typos for existing words. Anecdotally, MFA's alignment quality remains very good when up to 5–10% of word types are unknown.

[3]Similarly, disabling speaker adaptation may lead to better alignments if there is little enough data per speaker.

A common source of alignment errors in read speech like audio books or laboratory experiments is deviations from the prompt, such as filled pauses, restarts, or speech errors. Transcriptions of spontaneous speech often contains analogous transcription errors, since listeners are prone to filtering out such deviations. Rather than manual inspection of each audio file for deviations from the transcription, MFA offers a feature from Kaldi to facilitate finding and correcting them. A limited lexicon per utterance is generated, supplemented with frequent words, and a simple speech recognition pass is run on the file to generate a transcript. This generated transcript is compared to the original transcript and deviations are saved to facilitate manual inspection.

3. Evaluation

Our evaluation of MFA addresses three questions: (1) how good is the aligner's performance relative to manual annotation, and what is the effect on performance of the two key aspects of MFA: (2) architecture (acoustic model and speaker adaptation) and (3) trainability? We evaluate MFA's performance by examining its accuracy on detecting phone and word boundaries in two datasets, representing types of speech commonly used in language research: isolated-word lab speech and conversational interview speech. We compare MFA to two existing widely-used aligners with simpler architectures—FAVE and Prosodylab-Aligner—and vary the training data for aligners where possible.

3.1. Datasets

The first dataset used in our evaluations was the Buckeye Corpus [24], which contains 20.7 hours of conversational speech from 40 speakers. Buckeye comes with manual transcription and boundaries at the phone and word level, which were produced by forced alignment followed by manual correction. The Buckeye phone set represents more subphonemic detail (e.g. flapping) than needed for our evaluations; we thus mapped it to the phone set used in our pronunciation dictionary (see below).

HTK-based aligners, such as FAVE and Prosodylab-Aligner, require relatively short speech chunks. We thus broke up Buckeye into chunks bounded by non-speech (pauses, noise, interviewer speech) of $>$150 msec marked in the transcription files, using PolyglotDB.[4] Each of these chunks consists of an orthographic transcription and speech, as well as corresponding word and phone-level manual alignments. In our evaluation, the transcription and speech are force-aligned, and the manual alignments used as the gold standard.

Utterances were excluded if they contained words not in the pronunciation dictionary used in evaluation, for comparability between FAVE/Prosodylab-Aligner (which require all words to be in the dictionary) and MFA (which does not).

The second dataset, Phonsay, consists of 48 minutes of lab speech from 45 participants from two experiments. Participants said words in the frame "Please say ___ again". The target words all contained vowels followed by a consonant: a voiced obstruent, unvoiced obstruent, or sonorant (e.g. *buzz*, *bus*, *bun*). The boundaries of the vowel and the following consonant were hand-annotated, and these manual annotations are the gold standard in our evaluation.

In the evaluation, we examine two kinds of boundaries. First, left and right *word boundaries*, across all words, for Buckeye only. (Most word boundaries in Phonsay were not anno-

[4]https://github.com/MontrealCorpusTools/PolyglotDB

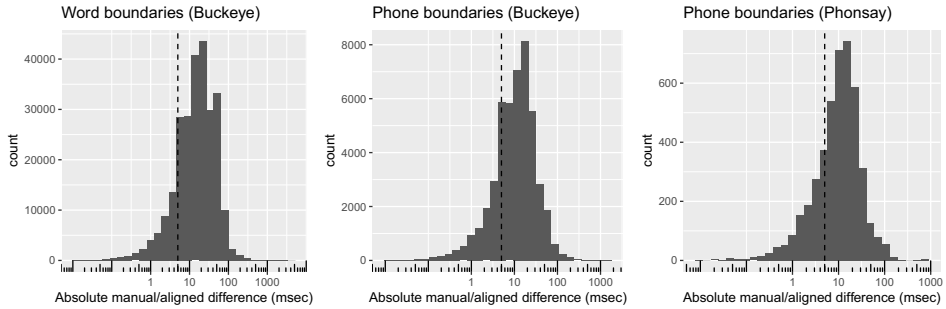


Figure 1: *Histograms of absolute differences (on log scale) between force-aligned word and phone boundaries using MFA-LS aligner and gold-standard annotations. Dashed line is at 1/2 frame rate (5 msec), which is a lower bound on average absolute difference.*

tated.) Second, *phone boundaries*, for each phone boundary of CVC words in either dataset, that corresponds to a manually-annotated boundary. For Buckeye, this is all four boundaries (denoted .CVC, C.VC, CV.C, CVC.). The CVC words in Buckeye were those from the list of [25], with the additional criterion of having all three segments realized in some way according to the manual transcription. For Phonsay, the boundaries were C.VC, CV.C, and CVC. for the target word in every sentence.

3.2. Aligners and training

Our evaluation uses MFA and two HTK-based aligners which are currently used in language research: FAVE, the most widely-used aligner in recent work, and Prosodylab-Aligner (PLA). PLA and FAVE are used as representative of aligners using GMM-HMM monophone acoustic models[5] without speaker adaptation, which are and are not trainable, respectively. Many existing aligners fall into these two categories (e.g. [6, 7, 8, 15]).

In order to minimize out-of-vocabulary words for PLA and FAVE, the pronunciation dictionaries which ship with each of the three aligners were combined into one Arpabet-based dictionary, which was used across all three aligners for training (MFA, PLA) and alignment (MFA, PLA, FAVE).

Both MFA and PLA were trained in two ways: on the LibriSpeech corpus, and on the corpus to be aligned: Buckeye (the subset without unknown words) or Phonsay. For training on LibriSpeech, MFA was trained on the full corpus (∼1000 hours), while PLA was trained on the 'clean' subset (∼450 hours), due to technical difficulties in HTK training on large datasets. For training on Buckeye, we treated the corpus as if only utterance boundaries and the orthographic transcription were known, to simulate the most common case in aligning speech in linguistic research. We refer to the resulting trained aligners as *MFA-LS*, *MFA-Retrained*, *PLA-LS*, and *PLA-Retrained*, where the "retrained" aligners refer to the version trained on Buckeye or the version trained on Phonsay, when discussing each corpus. We also used the existing version of *FAVE*, which uses acoustic models trained on the SCOTUS corpus (25 hours) [26]. Thus, our experiments compare five types of aligner (*MFA-*{*LS, Retrained*}, *PLA-*{*LS, Retrained*}, *FAVE*).

Each type of aligner was applied to align the Buckeye and Phonsay datasets, resulting in predicted word and phone boundaries. Note that we did not split the datasets into training and test sets, as the common use case for a trainable aligner is to simultaneously train on and align the entire dataset of interest.

[5]While it is possible to use triphone models in HTK, all distributed software packages for alignment use monophone models.

Table 1: *Accuracies at different tolerances (percentage below a cutoff) for absolute differences between force-aligned boundaries using MFA-LS aligner, and gold-standard annotations.*

	Tolerance (ms)			
	<10	<25	<50	<100
Word boundaries (Buckeye)	0.33	0.68	0.88	0.97
Phone boundaries (Buckeye)	0.41	0.77	0.93	0.98
Phone boundaries (Phonsay)	0.36	0.72	0.88	0.95

Our evaluation considers two subsets of the predicted boundaries, described above: word boundaries (Buckeye only), and phone boundaries (Buckeye and Phonsay). The metric we use for accuracy of a force-aligned boundary is the absolute difference (in msec) from the manually-annotated boundary.

3.3. Results

Our results address questions (1)–(3): how good are MFA's alignments 'out of the box' compared to hand annotation, and do the more complex architecture and trainability of MFA lead to more accurate alignments?

3.3.1. Alignment quality

We first consider the performance of MFA-LS, which is the version distributed with the current version of MFA. Performance on the two datasets approximates the performance a user can expect if MFA-LS is applied to lab (Phonsay) or conversational (Buckeye) English data, without retraining.

Figure 1 and Table 1 show the distribution of manual/force-aligned differences, for each kind of boundary, for the two datasets. The distributions of differences are highly right-skewed, as for other forced aligners [8, 26]: 2–5% of tokens have differences of at least 100 msec, while about 90% have differences of less than 50 msec. Table 2 (row 1) gives the mean and median of manual/aligned boundary differences for each case. These measures can be compared for the Buckeye corpus to differences between human transcribers reported by [27]—bearing in mind that the set of word and phone boundaries used there differs from the set used in our evaluation.

For word boundaries, the mean manual/aligned difference is 24 msec, which is comparable to 26 msec intertranscriber

Table 2: *Comparison of aligners in detecting word boundaries (Buckeye only) and phone boundaries (Buckeye and Phonsay). Means and medians are over differences between aligned and gold-standard boundaries.*

	Word bound.		Phone boundaries			
	Buckeye		Buckeye		Phonsay	
Aligner	mean (ms)	med (ms)	mean (ms)	med (ms)	mean (ms)	med (ms)
MFA-LS	24.1	15.8	**17.0**	**11.2**	25.2	11.3
MFA-Retrained	**22.6**	**15.0**	17.3	11.8	**16.6**	**10.8**
PLA-LS	30.5	15.6	24.0	13.9	40.1	21.5
PLA-Retrained	27.2	15.6	24.7	15.8	25.9	16.5
FAVE	24.7	16.6	19.3	12.0	21.8	13.0

reliability [27]. 68% of manual/aligned differences are under 25 msec, which is significantly lower than the 90% intertranscriber agreement reported at 26 msec tolerance.

For phone boundaries, the mean difference is 17 msec for Buckeye and 25 msec for Phonsay. For Buckeye, an identical figure (17 msec) is reported for intertranscriber agreement [27]. The median difference is comparable (11 msec) for Phonsay and Buckeye, suggesting that the main difference between them is more gross misalignments for Phonsay (visible in Fig. 1 right).

In sum, MFA performs well across both datasets and boundary types. While phone and word-level alignment is comparable to human annotators on average, the force-aligned boundaries do contain more medium-to-large alignment errors (>25 msec).

3.3.2. Architecture

To examine the effect of MFA's more complex architecture—triphone acoustic models and speaker-adapted features, compared to monophone acoustic models without speaker adaptation—we compare MFA-LS to PLA-LS and FAVE. The comparison with PLA-LS is most important, since MFA is essentially the same as PLA except for this modified architecture.

Rows 1, 3, 5 of Table 2 show, for these three aligners, the mean and median differences between manual and force-aligned boundaries for each condition. In most cases (columns of Table 2), the ordering is MFA-LS < FAVE < PLA-LS. However, MFA-LS and PLA-LS have roughly the same median for word boundaries for Buckeye (below FAVE), and FAVE has the lowest mean for phone boundaries for Phonsay.[6] Still, MFA-LS has the best overall performance of the three aligners. The difference between MFA-LS and PLA-LS suggests that MFA's different architecture led to better alignments.

To what extent is MFA's performance in this comparison due to the updated acoustic model versus speaker adaptation? Experiments with a version of MFA with speaker adaptation disabled suggest that it is the triphone acoustic model that primarily accounts for MFA's performance relative to PLA, with 88%/95% of the performance difference for word/phone boundaries (as measured by mean absolute manual/aligned difference) between PLA-LS and MFA-LS on Buckeye coming from just changing the acoustic model.[7]

3.3.3. Experiment 3: Training

To examine the effect of retraining on the dataset to be aligned, we compare MFA-Retrained to MFA-LS and PLA-Retrained to PLA-LS. This comparison represents a common use case: a researcher has a medium-to-large dataset (say 5–50 hours) of speech from speakers of a given type (e.g. Buckeye: Columbus-dialect adults). She can either re-train the aligner's acoustic models on this data, or use acoustic models which have been pre-trained on a much larger dataset that contains significant interspeaker variation (e.g. LibriSpeech: 1000 hours). Will training on a smaller amount of more similar data or a larger amount of more variable data give better alignments?

The effect of retraining can be evaluated by comparing rows 1 and 2 of Table 2 for MFA, and rows 3 and 4 for PLA, again examining the mean and median of absolute differences between manual and aligned boundaries. In five cases (Buckeye word boundary mean for MFA/PLA, Phonsay phone boundary mean for MFA and mean/median for PLA), retraining leads to better performance, decreasing the mean or median difference by at least 1 msec. In six of the remaining seven cases, retraining makes little difference (< 1 msec mean or median). In only one case (Buckeye phone boundary median for PLA) does retraining lead to clearly worse performance (> 1 msec difference).

On balance, retraining on the dataset to be aligned often improves alignment accuracy relative to using acoustic models pretrained on a larger dataset—and rarely hurts. However, the discrepancy between mean and median values in some conditions suggests that a more thorough evaluation should examine the effect of retraining on gross alignment errors.

4. Conclusion

We have presented a new open-source trainable forced aligner for language research, the Montreal Forced Aligner, which updates the Prosodylab-Aligner. MFA uses more complex acoustic models (triphones), and is built using the Kaldi toolkit instead of HTK. MFA showed good performance in aligning word and phone boundaries in one lab speech dataset and one spontaneous speech dataset. Notably, in each test case (columns of Table 2), it is one of the MFA aligners which gives the most accurate alignment relative to the gold standard.

Our evaluation suggests that both MFA's more complex architecture and the ability to retrain on new data generally improve performance. Using triphone acoustic models in particular seems to improve accuracy, compared to the monophone models commonly used in current aligners. More complex architectures, such as using artificial neural network models implemented in Kaldi (as in [14]), could improve accuracy further and are planned in future development. Retraining on the data to be aligned generally improved alignment accuracy, though it often had little effect—perhaps reflecting the similarity of training data for all aligners tested (North American English).

The mixed results of our evaluations point to the need for more thorough evaluations of forced aligners, to establish best practices for deploying forced alignment in language research [2, 28, 29]. Future work could examine the conditions under which adding speaker adaptation, or adapting an existing forced aligner versus retraining, improves alignment [30].

5. Acknowledgements

We acknowledge funding from SSHRC #430-2014-00018, FRQSC #183356 and CFI #32451 to MS, and SSHRC #435-2014-1504 and the SSHRC CRC program to MW.

[6] All comparisons are significant (paired Wilcoxson rank-sum test).

[7] Disabling speaker adaptation gives *better* performance as measured by the median, suggesting that enabling speaker adaptation may induce more gross errors, while increasing mean alignment accuracy.

6. References

[1] M. Adda-Decker and N. D. Snoeren, "Quantifying temporal speech reduction in French using forced speech alignment," *Journal of Phonetics*, vol. 39, no. 3, pp. 261–270, 2011.

[2] C. DiCanio, H. Nam, D. H. Whalen, H. Timothy Bunnell, J. D. Amith, and R. C. García, "Using automatic alignment to analyze endangered language data: Testing the viability of untrained alignment," *The Journal of the Acoustical Society of America*, vol. 134, no. 3, pp. 2235–2246, 2013.

[3] W. Labov, I. Rosenfelder, and J. Fruehwald, "One hundred years of sound change in Philadelphia: Linear incrementation, reversal, and reanalysis," *Language*, vol. 89, no. 1, pp. 30–65, 2013.

[4] B. Schuppler, M. Ernestus, O. Scharenborg, and L. Boves, "Acoustic reduction in conversational Dutch: A quantitative analysis based on automatically generated segmental transcriptions," *Journal of Phonetics*, vol. 39, no. 1, pp. 96–109, 2011.

[5] J. Yuan, M. Liberman, and C. Cieri, "Towards an integrated understanding of speaking rate in conversation," in *Proceedings of Interspeech*, 2006, pp. 541–544.

[6] I. Rosenfelder, J. Fruehwald, K. Evanini, and J. Yuan, "FAVE (Forced Alignment and Vowel Extraction) Program Suite [Computer program]," 2011, available at http://fave.ling.upenn.edu.

[7] T. Kisler, F. Schiel, and H. Sloetjes, "Signal processing via web services: the use case WebMAUS," in *Digital Humanities Conference 2012*, 2012.

[8] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics*, vol. 39, no. 3, pp. 192–193, 2011.

[9] S. Reddy and J. Stanford, "A web application for automated dialect analysis," in *Proceedings of HLT-NAACL*, 2015, pp. 71–75.

[10] J. Yuan, N. Ryant, M. Liberman, A. Stolcke, V. Mitra, and W. Wang, "Automatic phonetic segmentation using boundary models," in *Proceedings of Interspeech*, 2013, pp. 2306–2310.

[11] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 286–290.

[12] A. Pettarin, "Aeneas [computer program]," 2017, available at https://www.readbeyond.it/aeneas/.

[13] R. Fromont and J. Hay, "LaBB-CAT: An annotation store," in *Australasian Language Technology Association Workshop 2012*, vol. 113, 2012, pp. 113–117.

[14] R. M. Ochshorn and M. Hawkins, "Gentle forced aligner [computer program]," 2017, available at https://github.com/lowerquality/gentle.

[15] J.-P. Goldman, "EasyAlign: an automatic phonetic alignment tool under Praat," in *Proceedings of Interspeech*, 2011, pp. 3233–3236.

[16] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan, "SailAlign: Robust long speech-text alignment," in *Proceedings of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[17] B. Bigi, "SPPAS: a tool for the phonetic segmentations of speech," in *Proceedings of LREC*, 2012, pp. 1748–1755.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 1–4.

[19] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance Gaussians," in *Proceedings of Interspeech*, 2006, pp. 1145–1148.

[20] T. Schultz, N. T. Vu, and T. Schlippe, "GlobalPhone: A multilingual text & speech database in 20 languages," in *Proceedings of ICASSP*, 2013, pp. 8126–8130.

[21] E. Barnard, M. H. Davel, C. J. van Heerden, F. De Wet, and J. Badenhorst, "The NCHLT speech corpus of the South African languages," in *Proceedings of SLTU*, 2014, pp. 194–200.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proceedings of ICASSP 2015*, 2015, pp. 5206–5210.

[23] P. P. G. Boersma *et al.*, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, 2002.

[24] M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, *Buckeye Corpus of Conversational Speech (2nd release)*. Department of Psychology, Ohio State University, 2007.

[25] S. Gahl, Y. Yao, and K. Johnson, "Why reduce? phonological neighborhood density and phonetic reduction in spontaneous speech," *Journal of Memory and Language*, vol. 66, no. 4, pp. 789–806, 2012.

[26] J. Yuan and M. Liberman, "Speaker identification on the SCOTUS corpus," in *Proceedings of Acoustics '08*, 2008, pp. 5687–5790.

[27] W. D. Raymond, M. A. Pitt, K. Johnson, E. Hume, M. J. Makashay, R. Dautricourt, and C. Hilts, "An analysis of transcription consistency in spontaneous speech from the Buckeye corpus," in *Proceedings of Interspeech*, 2002.

[28] P. Milne, "The variable pronunciations of word-final consonant clusters in a force aligned corpus of spoken French," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2014.

[29] T. Knowles, M. Clayards, M. Sonderegger, M. Wagner, A. Nadig, and K. Onishi, "Automatic forced alignment on child speech: Directions for improvement," *Proceedings of Meetings on Acoustics*, vol. 25, p. 060001, 2015.

[30] L. MacKenzie and D. Turton, "Crossing the pond: Extending automatic alignment techniques to British English dialect data," 2013, talk given at *New Ways of Analyzing Variation 42*.