Autophon user guide
English – North America
Model: Montreal Forced Aligner 1.0 (English)

1 Introducing Autophon and forced alignment

Autophon is a **free** online forced aligner. *Forced alignment* (FA) refers to the automatic process by which speech recordings are phonetically time-stamped with the help of Hidden Markov models or Deep Neural Networks. Autophon uses the latter by means of the *Montreal Forced Aligner*[1], which is built on the Kaldi toolkit[2]. The app outputs a time-stamped phonetic annotation, readable in Praat (Boersma and Weenink 2017), that is based on an optimization of two user inputs: (1) the speech recording and (2) a corresponding orthographic transcription.

Forced alignment is important because it automates something that is resource-intensive when done manually. A typical phonetic annotation can take between 250 and 400 minutes per recorded minute. In a place like Scandinavia – where labor costs are high – this cost has presented a barrier for linguists.

For a forced alignment tool to work, an acoustic model must be trained on the specific language, and an accompanying pronunciation lexicon must be built that covers every word in the language (See section 5).

Numerous forced aligners are in circulation and available to download and use. However, they often are command-line based and rely on operating systems (OS) that may be outdated and/or incompatible with your OS. Therefore, *Autophon aims to offer an **OS-agnostic** and **user-friendly** option for phoneticians around the world.*

2 Using the app

Logging in You can create an account by following the relatively intuitive guidelines on our website. We require an account because we wish to keep track of usage in order to make a case for funders. Furthermore, an open system makes us vulnerable to bot attacks. After registering, a verification email will be sent to you with a link that you must click on to verify your account. If you do not receive the email, first check your spambox and then wait at least 15 minutes before contacting tech support.

Cost Autophon is free of charge.

Adding files First go to the *Aligner* tab and click *Add files*. A box will appear with the heading *Transcription Mode: change transcription mode*. Click on the heading to select one of four *Transcription Modes*. Once your transcription mode has been selected, use the file browser underneath to select your files.

Transcription modes Four different *transcription modes* are available, each named according to the field in which the format is most common: *Experimental Linguistics A*, *Experimental Linguistics B*, *Computational Linguistics*, and *Variationist Linguistics*. Each can be selected by clicking on one of the boxes illustrated in Figure 1. The boxes illustrate a typical file structure for each mode and provide a link to a video that offers detailed formatting instructions.

> **Experimental linguistics A:** In this mode, you upload a master transcription spreadsheet along with corresponding audio files — one by one or within a zip file. The master sheet should have two columns: column 1 holds the audio file names in your folder; column 2 holds the corresponding transcriptions. This format is similar to that used by, e.g., CommonVoice[3] and assumes that each audio file contains a short snippet of speech, which means that time stamps are *not* permitted. If you have a master transcription spreadsheet with time stamps, you are in the wrong transcription setting and need to select *Experimental Linguistics B*, described below. The master transcription sheet can be in a two-column Excel `xlsx` or tab-delimited file with either the extensions `txt` or `tsv`.
>
> **Experimental linguistics B:** In this mode, you also upload a master transcription spreadsheet with corresponding audio files— one by one or within a zip file. Unlike in mode A, it should have four columns: column 1 holds the names of the audio files in your folder; column 2 — start times; column 3 — end times; column 4 — transcription. This mode is designed for longer audio files that warrant multiple

[1] McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger (2017)
[2] Povey, Ghoshal, Boulianne, Burget, Glembek, Goel, Hannemann, Motlicek, Qian, Schwarz, et al. (2011)
[3] https://commonvoice.mozilla.org

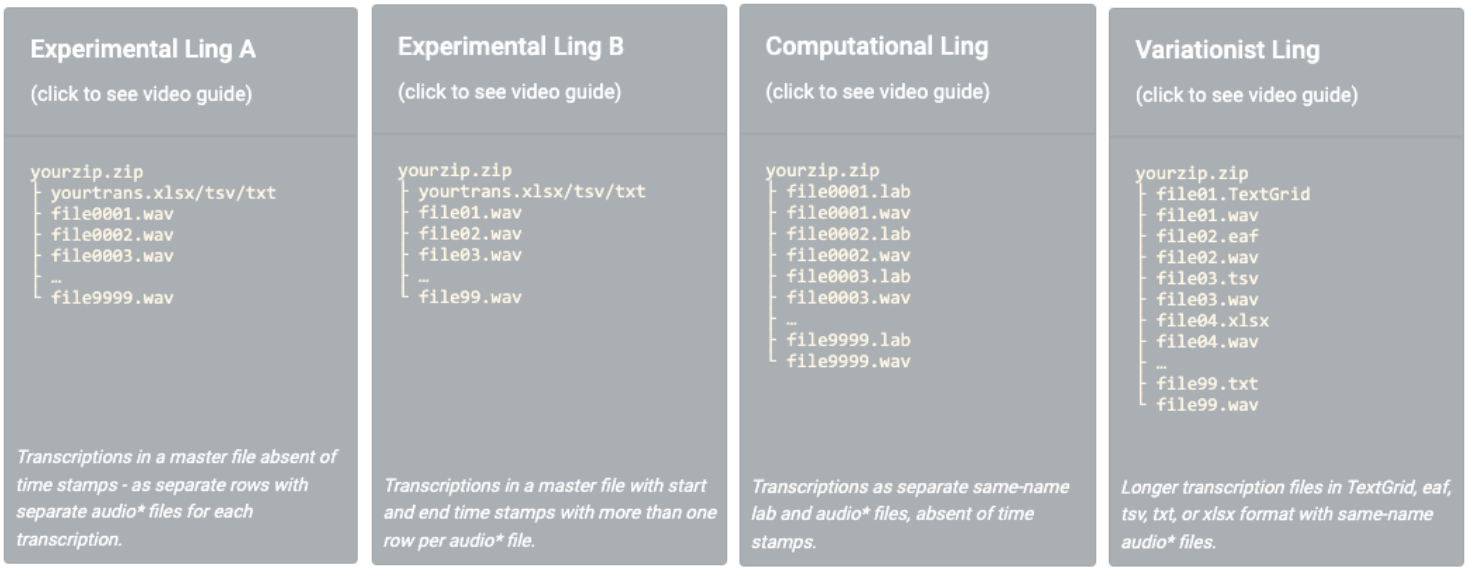

Figure 1: The Transcription Mode selection menu for Autophon.

Figure 2: Autophon will output the finished TextGrids using an identical subfolder structure as the uploaded file.

lines of transcription. The master transcription sheet should either be a four-column Excel `xlsx` or a tab-delimited file with either the extensions `txt` or `tsv`. Time stamps must be in *seconds formatted as real numbers*. European comma decimals are accepted (e.g., `1,23`) as well as Anglo-American period decimals (e.g., `1.23`). What will not work, however, are time stamps with colons, minutes, or hours (e.g., `00:00:01.23`).

Computational linguistics: In this mode, you upload pairs of `lab` and audio files by the same name – one by one or within a zip file. These so-called `lab` files are simply text files that contain a single transcription phrase that matches the speech within the same-named audio file. Importantly, transcriptions should contain no time stamps. If you wish to have a complex set of subfolders within the zip file, as is common for comp-ling corpora like, e.g., NST[4], Autophon will output the finished TextGrids using the same folder structure. An example of one such folder hierarchy is shown in Figure 2

Variationist linguistics:[5] In this mode, you upload pairs of transcription and audio files by the same – one by one or within a zip file. In contrast to the previous mode, transcriptions are longer and include time stamps that delineate the speech at the phrase level. Transcription files may be in Praat `TextGrid` or in ELAN `eaf` and may have multiple speaker tiers. Alternatively, transcription files may be in Excel `xlsx` or a tab-delimited `txt` or `tsv` file.[6]. You have the option of uploading a three-column or four-column file, depending on your needs. If the recording has multiple speakers, upload a four-column transcription file whereby column 1 holds the speaker name, column 2 – start time, column 3 – end time, and column 4 – transcription. If the recording has just one speaker, a four-column file is of course fine, but you may also upload a three-column file. Column 1 should hold the start time, column 2 – end time, and column 3 – transcription. Time stamps must be in *seconds formatted as real numbers*. European comma decimals are accepted (e.g., `1,23`) as well as Anglo-American period decimals (e.g., `1.23`). What will not work, however, are time stamps with colons, minutes, or hours (e.g., `00:00:01.23`).

Transcription codecs We have built Autophon so that it accepts transcription files in ***most*** codecs, and this is a vital feature for its OS-agnostic goal. Accepted codecs include, but are not limited to, UTF-8 Unix, UTF-16 Windows CRLF, Windows ISO Latin 1, and Windows ISO Latin 9. If you encounter errors, please email a sample file to tech support so that we can update our code with that format[7]

Audio codecs We have built Autophon so that it accepts audio files in ***most*** codecs, and this is a vital feature for its OS-agnostic goal. Accepted codecs include AAC(M4A), AC-3, AIFF, AIFF/24bit, AIFF/32bit, ALAC, FLAC, M4R, MP3, OGG, OPUS, WAV/8bit, WAV/24bit, WAV/32bit, WAV/A-law, WAV/mu-law, and WMA. Autophon will automatically consolidate stereo files to mono, *which may compromise quality due to phase cancellation*[8]. Therefore, you may wish to explicitly select either the left or right channel of your stereo file before aligning. If you encounter errors, please email a sample file to tech support so that we can update our code with that format[9]

Transcription preparation Regardless of what transcription mode you use, transcriptions should contain between one and 20 words. Boundary demarcations should have at least 0.01 seconds of buffer before and after the speech stream. This is illustrated in Figure 3, which shows a five-word phrase with a start boundary 0.03 seconds from the speech and an end boundary 0.25 seconds from the speech. Varying the boundary demarcation in this way is expected, and Autophon handles it well[10]

Select a language Once you upload your files into the aligner, it will suggest a language and language model. You are welcome to change the selection using the dropdown menu.

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

[4] https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-16/
[5] This also happens to be the field that originally kickstarted forced alignment back in the early 2000s.
[6] The tab-delimited format is similar to the input format that was used for the legacy Penn Forced Aligner and FAVE Align.
[7] In the meantime, a quick fix is to open and resave them in a current version of Praat or ELAN.
[8] For more on phase cancellation, check out https://youtu.be/wY9QokRPJts
[9] In the meantime, a quick fix is to convert the file to WAV using software like FFmpeg or MediaHuman audio converter.
[10] If you have transcriptions of single words or phrases that are segmented at the *exact* start and finish times. Autophon will perform poorly and move those boundaries. This, however, is something we would be interesting in remedying by means of a fifth transcription mode, so kindly reach out to tech support if you have such a project.

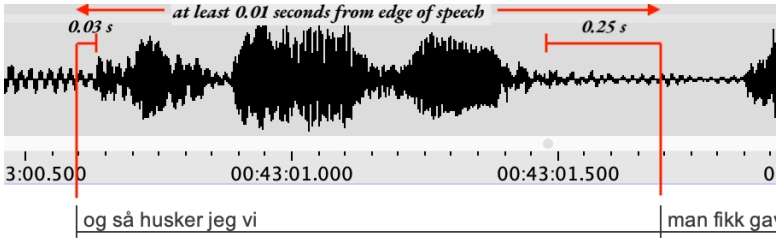

Figure 3: A sample transcription with at least 0.01 seconds of buffer on either end of the speech stream.

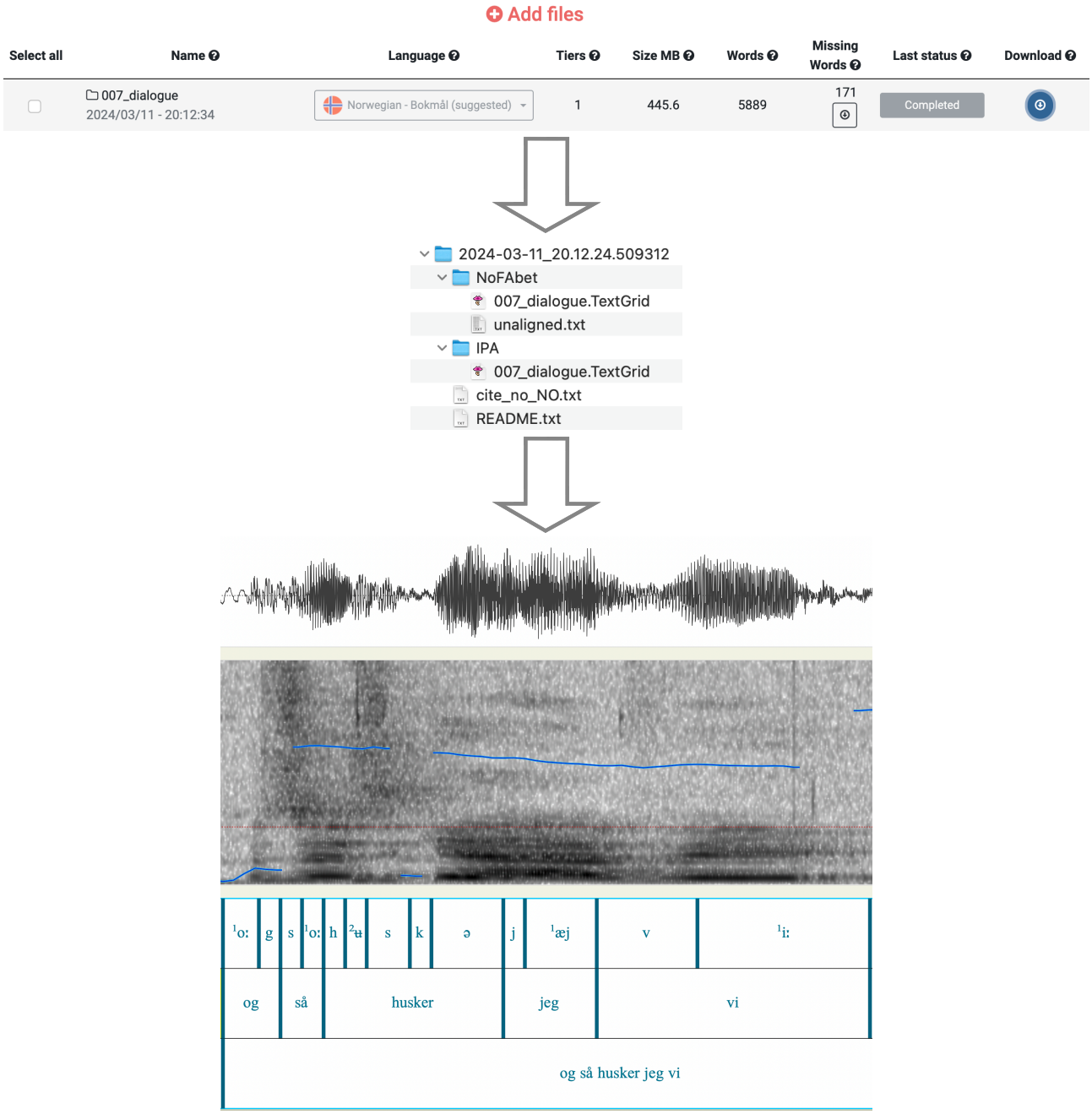

Figure 4: The alignment process, including task list, folder structure, and Praat TextGrid.

Missing words This feature can be understood if you have a basic understanding of how forced alignment works. Forced alignment maps a pre-defined phonemic pronunciation onto the speech stream by means of Deep Neural Networks. These pronunciations are defined by language-specific dictionaries that hold a finite list of words. The *missing words* feature provides a list of words not found in Autophon's dictionary and suggests a corresponding pronunciation. Autophon will simply default to using those suggestions for alignment, but you also can reject a suggestion and enter your own pronunciation. This process is described in the next section.

Your custom pronunciations As described above, forced alignment maps pre-defined phonemic pronunciations onto the speech stream by using language-specific dictionaries that hold a finite list of words. For missing words, Autophon suggests a pronunciation. You may decide that you either (a) do not agree with Autophon's missing words suggestions or that you (b) do not agree with the pronunciations within the language-specific dictionary. In this box you can enter your own pronunciations that will override both.

Pronunciations must be entered using the alphanumeric string specific to the language model at hand – in this case, ARPAbet. Table 1 holds a key for ARPAbet and its respective IPA[11] equivalents. You may type pronunciations directly into the dictionary box or upload them from a `txt` file. You are limited to 1 million characters. Entries must be formatted as `word-space-phoneme-space-phoneme` or `word-tab-phoneme-space-phoneme`. Note that each phoneme must be separated by a space and that every vowel and every diphthong must have a numerical stress attached to it (unstressed vowels take zero). Note also that the lookup cannot be two or more words because that will confuse Autophon and make it treat the second word as a phone.

You may enter more than one pronunciation for the same word by repeating the word on the next line and providing a different pronunciation. Autophon will respond by attempting to find the most suitable pronunciation for that specific speech event. See below for examples of correct versus incorrect entries.

§ Correct vs. incorrect entries in the "Your Custom Pronunciations" box.

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)
```

Aligning files Click *Align* to the far right of the upload list to initiate alignment. This will usually just take a few minutes, depending on how many people are using the aligner at that moment.

Downloading the annotations When alignment is finished, your annotations can be downloaded as Praat TextGrids via the downward arrow to the right of the task list. Figure 4 shows an example of this process.

3 How to cite

Any dissemination that makes use of Autophon *English – North America* should cite the below references. We understand that publishers often pressure researchers to slim down bibliographies; however we make our view plain: failure to cite constitutes plagiarism. Refer publishers to this document if you receive pushback.

Boersma, P., & Weenink, D. (2017). Praat: Doing phonetics by computer [Computer software], Version 6.0.36. http://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.

Young, N. J., & Anikwe, K. (2024). Autophon – Automatic phonetic annotation of Nordic languages (web application). www.autophon.se

[11] International Phonetic Alphabet

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 Montreal Forced Aligner 1.0 (English) and (2) a TextGrid in the International Phonetic Alphabet (IPA). The phoneme inventory for *English – North America* was taken from The CMU Pronouncing Dictionary[12]. The 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, foreword*
IY	i	*beat*	CH	tʃ	*China*	P	p	*pie*	***Lexical stress***		
OW	oʊ	*boat*	D	d	*die*	R	ɹ	*rye*	XX0	x	*banana*
UH	ʊ	*book*	DH	ð	*thy*	S	s	*sigh*	XX1	ˈx	*banana*
UW	u	*boot*	F	f	*fight*	SH	ʃ	*shy*	XX2	ˌx	*barnyard*
			G	g	*guy*	T	t	*tie*			
						TH	θ	*thigh*			

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

Every ARPAbet vowel is followed by a numerical code that denotes suprasegmental information. XX0 refers to lexically unstressed vowels; XX1 – primary lexical stress; XX2 – secondary lexical stress.

5 Acoustic model and pronunciation dictionary

English – North America uses the very same acoustic model developed for version 1.0 of the Montreal Forced Aligner, which was trained on American English (see McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger (2017) attached as an Appendix here). The pronunciation dictionary is the The CMU Pronouncing Dictionary[13].

6 Performance metrics

Metrics for this specific model can be accessed in (McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger 2017) (attached as an Appendix here).

7 Data security and G.D.P.R.

The files you upload to Autophon are encrypted and sent to a server in Frankfurt, Germany, that is run by *Digital Ocean*. Transcriptions and audio files are deleted immediately after alignment, which significantly reduces the chance of a data breach and keeps our costs low[14]. On the other hand, finished TextGrids are stored in your account for as long as you like. Once, however, you delete them, they will be removed from our server permanently.

If you upload any files and fail to click on *Align*, Autophon will delete them at 3AM Greenwich Mean Time[15].

We recognize our obligations to the European Union General Data Protection Regulation (GDPR), which is why we only collect four types of information from you: name, title, affiliation, and email address. Once you align a file, we permanently delete the audio. Once you delete the file from your task list, we also permanently remove

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

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

[14] We pay Digital Ocean approximately 90 USD per month for 60 GB of space, which means we have thin margins and cannot store much data. This also happens to keep Autophon's carbon footprint relatively low.

[15] Note that this means that if you are working late at night at, for example, 2.55 AM GMT, your uploaded files may disappear before you manage to align them. Bear this in mind.

the transcription and documentation of its original name. You may delete your account at any time, at which point we permanently delete your name, title, affiliation, and email address from our server. What we do *permanently* keep, however, is your alphanumeric account ID and the alignment activity linked to that ID — absent of original file names. We keep these records to show funders that Autophon is worth funding.

8 Features and limitations

What Autophon is: Autophon is a frontend web application for the Nordic languages that uses the Montreal Forced Aligner (MFA) [16] as a core component of its backend. The language-specific models and pronunciation dictionaries were constructed by Dr. Nate Young. The most significant pieces of the app's backend were constructed by Kaosi Anikwe who joined the project in early 2023. The language-specific models are trained on various corpora, and the pronunciation dictionaries are usually adaptations of existing dictionaries available online.

The main advantages of using Autophon are:

1. Autophon is a web app, which means it is OS-agnostic.
2. As a web app, it requires no programming knowledge, which expands access to researchers and students.
3. Autophon accepts nearly all types of transcription and sound formats.
4. Autophon has a limitless repertoire of pronunciations by making use of grapheme-to-phoneme algorithms.
5. Autophon has models for Nordic languages, which have typically been neglected by forced alignment tech.

What Autophon is not: Important limitations are:

1. This is no magic bullet. Even with an accurate orthographic transcription, results may not satisfy.
2. Autophon varies in accuracy, and this accuracy depends on the language, speaker, and style.
3. Accuracy metrics are complex projects unto themselves, so they are unavailable for most languages.
4. Autophon will be slower to implement core MFA updates because it consists of layers and layers of code packed around MFA. For example, MFA 2.0 and 3.0 are not part of its backend yet.

9 Budget and funding

Autophon has cost ca. SEK 768 000 (ca. EUR 69 000) to develop and maintain since 2021. It was initially financed with private means by Dr. Nate Young but has since grown in scope with a grant from the Swedish Academy, a grant from the Department of Linguistics and Scandinavian Studies at The University of Oslo, and it has received funding from the European Union's *Horizon 2020 research and innovation programme* under the *Marie Sklodowska-Curie* grant agreement No 892963. Furthermore, The National Library of Norway funded development of the Norwegian Bokmål model[17].

We are actively looking for funders and collaborators who will support Autophon. We are also willing to share authorship with someone who can prepare grant applications and successfully procure funding. Contact us on the support page if you are interested.

Acknowledgements

Numerous individuals helped make Autophon possible. Michael McGarrah has offered important industry guidance, and Kaosi Anikwe has been an invaluable backend and frontend developer. Ismail Raji Damilola helped develop a bootstrapping function to adapt monophone inventories. The following programmers also worked on the app in its infancy: Nabil Al Nazi, Zamanat Abbas Naqvi, and Santiago Recoba.

References

Boersma, P., & Weenink, D. (2017). Praat: Doing phonetics by computer [Computer software], Version 6.0.36. http://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.

[16] McAuliffe, Socolof, Mihuc, Wagner, and Sonderegger (2017)

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

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.

Young, N. J., & Anikwe, K. (2024). Autophon – Automatic phonetic annotation of Nordic languages (web application). www.autophon.se

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

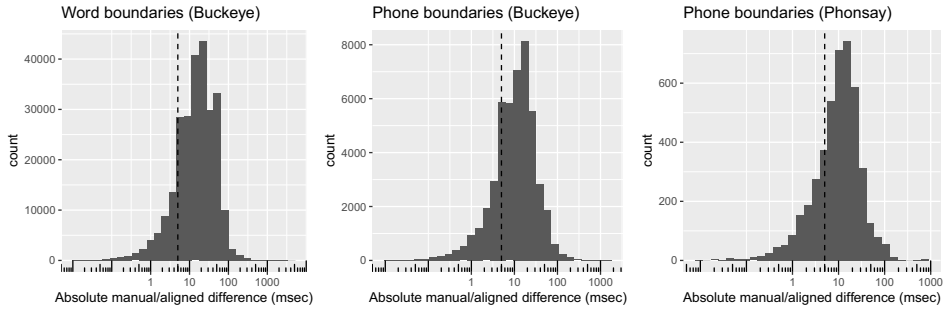

Figure 1: *Histograms of absolute differences (on log scale) between force-aligned word and phone boundaries using MFA-LS aligner and gold-standard annotations. Dashed line is at 1/2 frame rate (5 msec), which is a lower bound on average absolute difference.*

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*.