Bangla-Align: A forced-aligning toolkit for annotating Bangla speech

Md. Jahurul Islam[1]
Lecturer
Department of Linguistics, University of British Columbia

Manuscript Received: 05/08/2023
Accepted: 16/12//2023
Published: 08/02/2024

Abstract

This paper presents Bangla-Align, a toolkit for force-aligning Bangla speech. Forced-aligners are widely used tools for annotating speech at the phone level which greatly aids in increasing the processing power. While forced-aligning tools are freely available for a number of languages including, English, French, Spanish, etc., there is no aligner to process Bangla speech. Bangla-Align is built on top of the Montreal-forced-aligner (McAuliffe et al., 2017) with some Python scripts, which takes the audio recordings (.wav format) and their transcriptions (Text Grid format) as input and then returns Text Grids with phoneme-level annotations for phonetic/acoustic analyses of the audio data. The aligner uses a rule-based and continually developing phoneme dictionary. The aligner currently runs on Linux operating system only via command line interface. Bangla-Align will facilitate phonetic/acoustic research involving Bangla data.

Key words: Bangla-align, Bangla speech, forced aligner, align performance

Introduction

Speech processing and analysis have become integral components of linguistics research, facilitating the study of spoken language and its underlying structures. One essential tool in this domain is a forced-aligner, a powerful computational tool designed to synchronize speech signals with their corresponding linguistic

[1]jahurul.islam@ubc.ca

transcriptions. As Wu et al. (2023) define it, "Forced alignment is a speech technique that can automatically align audio files with transcripts." A forced-aligner aligns phonetic units, such as phonemes or words, in the speech signal to their respective time points in the transcription, thereby providing a valuable time-aligned representation of the spoken utterance. Forced-aligners are used by numerous researchers to reduce the amount of manual workload and increase data processing power (Mahr, et al., 2021) and the benefits offered by this method have attracted the development of many different types of aligners including the traditional statistical methods (McAuliffe et al., 2017) as well as deep neural networks (Kelly & Tucker, 2018; Zhu et al., 2022). This temporal correspondence enables linguists to conduct various phonetic, phonological, and prosodic analyses, ultimately shedding light on the intricate aspects of language and speech.

The primary purpose of a forced-aligner is to streamline the time-consuming process of aligning speech data with their transcriptions manually. Manual alignment is a laborious task that demands meticulous precision, and its feasibility largely depends on the scale of the data being analyzed. A forced-aligner, on the other hand, employs sophisticated algorithms and acoustic models to automatically identify the correspondence between speech and transcription, significantly reducing the effort required. Besides saving time, the automated alignment process ensures greater consistency and reproducibility in linguistic studies, making it an invaluable asset for researchers and practitioners in the field of linguistics.

Linguists extensively utilize forced-aligners to investigate various linguistic phenomena. By employing these tools, researchers can analyze speech corpora (Milne, 2011), study phonetic variations (Young & McGarrah, 2023) and language documentation (Ćavar et al., 2016), investigate speech disorders (Punnoose, 2022), and investigate prosodic patterns (Wu et al., 2023), among other applications. Furthermore, forced-aligners play a crucial role in developing speech recognition systems, facilitating language learning and teaching processes. With the ability to process large

volumes of speech data efficiently and accurately, forced-aligners have become indispensable tools in linguistic research and beyond.

While forced-aligners exist for several languages, including English, French, Chinese, and Spanish, etc., the same cannot be said for Bangla speech. This paper presents "Bangla-Align," a novel toolkit based on a set of Python scripts and the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017), designed to address this gap. The Bangla-Align toolkit offers a solution for force-aligning Bangla speech, bringing the benefits of automatic alignment to linguistics researchers working with this language. With Bangla-Align, researchers can now align large-scale Bangla speech data with corresponding transcriptions with relatively less effort, opening up new possibilities for exploring the rich linguistic landscape of this important language.

The remainder of this paper presents a comprehensive review of forced-aligners that are currently in use for languages including, English, Spanish, French, and many others, along with their strengths and weaknesses (Section 2). This is followed by a detailed introduction to Bangla-Align (Section 3), and performance evaluation (Section 4). Finally, implications for research into first and second language studies are discussed (Section 5), followed by a conclusion section (Section 6).

Common forced-aligners

This section provides a quick overview of a selection of currently available forced-aligned that are used by linguistics researchers. While there are other aligners outside the ones reviewed here, this section focuses on the ones that have been in widespread use by researchers.

Penn Phonetics Lab Forced Aligner

The Penn Phonetics Lab Forced Aligner (Yuan, J. & Liberman, 2008), known as P2FA, is an automatic speech alignment tool designed specifically for English language alignment. P2FA is based on the Hidden Markov Model Toolkit (HTK) (Mor et al., 2021; Gales & Young, 2008), a widely used platform for speech

recognition. The acoustic models used in P2FA were trained on audio of US Supreme Court oral arguments. Like MFA and the ProsodyLab aligner, P2FA operates through a command-line interface, requiring users to be familiar with terminal commands. The interface allows for batch alignment of files, facilitating the processing of multiple audio and transcript pairs concurrently. P2FA previously had a web interface front end named FAVE-Align (Rosenfelder et al., 2014), which is no longer available, making command-line usage the standard method for alignment tasks.

P2FA is primarily focused on English language alignment; therefore, researchers working exclusively with English speech data can benefit from P2FA's alignment capabilities. However, it is important to note that the installation process and training of new acoustic models for P2FA can be complex, requiring users to allocate sufficient time and resources for setup. Additionally, P2FA is limited to Mac and Linux operating systems, with no support for Windows platforms.

Prosody lab Aligner

The ProsodyLab aligner (Gorman et al., 2011) is an automatic speech alignment tool that employs a combination of the Hidden Markov Model Toolkit (HTK), SoX, and a set of Python scripts to facilitate the alignment of audio recordings and their corresponding transcripts. ProsodyLab aligner is primarily designed to work Mac and Linux operating systems; however, the documentation[2] claims that it may be used on Windows systems as well. Like P2FA, the primary focus of the ProsodyLab aligner is English language alignment. However, it is important to note that its performance excels when dealing with laboratory-style recordings, as highlighted by Gorman et al. (2011). This specialization sets it apart from the P2FA aligner (FAVE-align), which was more tailored to address dialectal variations in speech.

One notable capability of the ProsodyLab aligner is its adaptability to new languages. To perform alignment for a new language, researchers are required to provide several hours of high-

[2]https://github.com/prosodylab/Prosodylab-Aligner, Aug. 03, 2023.

quality speech data, accompanied by word-level transcripts. Leveraging this data, the ProsodyLab aligner induces a new acoustic model specific to the target language and subsequently computes the best alignments based on the generated model. Another distinctive aspect of the ProsodyLab aligner is its capacity to allow users to train their own acoustic models. This functionality offers researchers the flexibility to create custom models tailored to their specific alignment needs. By facilitating acoustic model training, the ProsodyLab aligner empowers users with greater control over the alignment process.

Montreal Forced Aligner

The Montreal Forced Aligner (MFA) (McAuliffe et al. 2017) is a state-of-the-art automatic speech alignment tool that stands apart from its predecessors, such as P2FA and ProsodyLab aligner, by being based on the Kaldi speech recognition toolkit. Unlike its predecessors that relied on the Hidden Markov Model Toolkit (HTK), MFA incorporates the advancements offered by Kaldi, providing enhanced accuracy and efficiency in speech alignment.

One significant characteristic of the Montreal Forced Aligner (and P2FA and ProsodyLab aligner too) is its exclusive reliance on a command-line interface. This interface requires users to have familiarity with terminal commands and grants them complete control over the alignment process. The command-line approach, though less user-friendly than a graphical user interface, offers the distinct advantage of enabling batch alignment of files. This means that researchers can process multiple audio and transcript pairs simultaneously, significantly expediting the alignment process and making it particularly suitable for large-scale projects. Unlike the DARLA aligner (introduced below), which lacks batch processing capabilities, the Montreal Forced Aligner thus proves to be a more efficient option for researchers dealing with considerable amounts of data.

An additional noteworthy feature of the Montreal Forced Aligner is its ability to align data even in the absence of acoustic models. This flexibility is especially beneficial for users who may

not possess the resources or expertise to create custom acoustic models. However, it is important to note that aligning without acoustic models can lead to longer alignment times, as the system needs to rely on built-in, generic models. Researchers should consider this trade-off while choosing their alignment strategy.

Language diversity is another strength of the Montreal Forced Aligner. The tool comes equipped with a wide array of pre-trained models, making it suitable for aligning speech recordings in various languages. This multilingual support expands the applicability of MFA, facilitating alignment tasks for researchers working with diverse linguistic datasets.

DARLA

One of the most convenient options for force-aligning speech data is the Dartmouth Linguistic Automation or DARLA (Reddy & Stanford 2015) service. With the MFA aligner (McAuliffe et al. 2017) as the backend, DARLA provides a GUI web interface where it is possible to simply upload audio and transcription files with simple mouse clicks and then receive the aligned TextGrids via email; that is, there is no need to possess command line skills to be able to force-align speech using DARLA. As per DARLA's website, the toolkit is tailored to the needs of sociophonetic research questions; however, the tool can be immensely useful to anyone working with speech data. On top of the semi-automatic aligning, DARLA also offers fully-automatic alignment, named Bed Word (Ma & Glass 2022), where aligned TextGrids can be generated solely from audio files (i.e., no transcriptions required); the accuracy, however, will be low for consonants and unstressed vowels while offering decent results for English stressed vowels (Reddy & Stanford, 2015). DARLA's functionality is limited to the English language only and no other languages. Another limitation of DARLA is that the user can align only one audio file at a time; therefore, if there are, for example, 30 files to align, it would involve repeating the process 30 times.

Easy Align

EasyAlign (Goldman, 2011) is a specialized plug-in for the widely used speech analysis software Praat, offering a semi-automatic and multi-step approach to speech alignment. This tool utilizes the Hidden Markov Model Toolkit (HTK) to find phone boundaries, enabling precise alignment of audio recordings and their corresponding orthographic transcriptions. To initiate the alignment process, users are required to provide the sound files and their respective orthographic transcriptions as plain text files. EasyAlign then proceeds to analyze the data through a series of steps, resulting in a TextGrid file with five tiers, namely phones, syllables, words, phonological transcriptions, and orthographic transcriptions. Notably, EasyAlign employs phone symbols following the Speech Assessment Methods Phonetic Alphabet (SAMPA) conventions, distinguishing it from other reviewed aligners that use the ARPAbet symbols. EasyAlign has been primarily designed to cater to the alignment needs of French, Spanish, and Taiwan Min languages. However, it does offer the possibility of training the tool for new languages. Nonetheless, this process is more involved, demanding additional effort and expertise. One notable aspect of EasyAlign is its exclusive compatibility with the Windows operating system. A key advantage of EasyAlign is its user-friendly interface, which eliminates the need for users to possess command-line familiarity. This feature simplifies the alignment process, making it more accessible to researchers without extensive technical expertise. Table 1 provides an overview of the aligners reviewed above:

Table 1: An overview of the currently available force-aligners to align speech data

Aligner	**Based on**	**languages**	**Required Transcription type**	**OS**	**Command line required?**
P2FA	HTK; Python	English	TextGrid, txt	Mac, Linux	Yes

Prosodylab	HTK; Python	English	TextGrid, txt	Mac, Linux	Yes
MFA	Kaldi; Python	Multiple languages	TextGrid, txt	Windows, Mac, Linux	Yes
DARLA	MFA	English only	TextGrid	Web-based	No
EasyAlign	HTK; Praat	English	txt	Windows	No

Introducing Bangla-Align

Bangla-Align is a toolkit designed specifically to force-align Bangla speech, offering researchers a solution for aligning audio and transcriptions in the Bangla language. The toolkit uses the MFA (McAuliffe et al., 2017) aligner in the back end along with a set of additional Python scripts dedicated to handling transcriptions in Bangla orthography, streamlining the alignment process. A distinguishing feature of Bangla-Align is its utilization of a newly developed phone symbol set based on the ARPAbet convention. This choice enhances alignment accuracy and consistency, ensuring reliable results for various speech analysis tasks. The acoustic models employed in Bangla-Align were trained on a small corpus of Standard Colloquial Bangla (Bangladeshi variety), with a specific emphasis on speech from news presenters.

Bangla-Align requires audio files and breath-group level transcriptions as input and produces outputs in the form of TextGrids with two tiers: word and phone. Currently, Bangla-Align is compatible with Linux systems and successfully tested on Linux Mint 21.2 (Victoria), which is based on Ubuntu 22.04. The software should function smoothly on Mac OS X as well, while developing a Windows version to expand its accessibility is planned to be future work.

The setup process for Bangla-Align is user-friendly and relatively straightforward. No separate software installation is necessary; as long as the Montreal Forced Aligner and Python (version 3.5 or later) are installed on the system, Bangla-Align should run seamlessly without issues.

To set up Bangla-Align, users can follow these steps:

A. Download the zip file from the following link: https://github.com/

B. Unzip the downloaded .zip file.

C. Place the unzipped folder named "Bangla-Align-linux_v1" inside the Documents folder or any other convenient location.

The following sections provide a detailed description of Bangla-Align's operational methodology, offering insights into its features and capabilities. With its specialized focus on Bangla speech alignment and user-friendly setup process, Bangla-Align stands as a valuable tool for researchers working with Bangla language datasets.

Preparing data

The aligner needs two types of files as input: the audio files and their transcription TextGrid files.

AUDIO data

The audio files to be used must be in .wav format. It is important to note that formants in formats other than .wav will not function properly. If the audio files are currently stored in different formats, they must be converted to .wav format. However, it is crucial to avoid storing the recordings in a lossy format (e.g., MP3) at any step of the process, as this may result in the loss of crucial phonetic details. The audio can be recorded at any sampling frequency, but for optimal performance, it is recommended not to use a sampling frequency lower than 16,000 Hz. It is also advised to use single-channel (mono) audio, as it is preferred for processing efficiency. While multiple channels (e.g., stereo) in the recordings

should still work, they may increase processing time without providing additional benefits. Additionally, the aligner may not perform well if the audio contains a substantial amount of background noise. Therefore, it is strongly recommended to record the audio in an environment with minimal background noise. For a comprehensive guide on good practices for audio recording in phonetic analyses, including microphone selection and placement, choice of environment, storage formats, and other relevant accessories, please refer to (Zsiga, 2014).

Bangla-Align offers the convenience of batch processing, allowing multiple files to be aligned in a single step. To ensure the smooth functioning of the aligner and avoid potential crashes, it is essential that all audio files within a batch have the same sampling frequency. Mixing files with different sampling frequencies in a batch may result in errors during alignment. In such situations, Praat (Boersma & Weenink, 2023) can serve as a helpful tool to downsample the audio (never upsample) as needed. Additionally, prior to transcription, it is recommended to remove any long unnecessary chunks in the audio files. This practice not only ensures the audio files are appropriately sized but also helps reduce processing time, improving overall efficiency. However, cleaning the audio is not a requirement; the file can still contain intervals that are not expected to be force-aligned; they can simply be kept blank.

TEXTGRID data

The aligner requires a set of TextGrid files, which are in the file format used by Praat (Boersma & Weenink, 2023). Each audio file must be transcribed using the TextGrid format. For transcription, the TextGrids should contain exactly one tier, which can have any name, and the tier must provide transcriptions at the breath-group (pause-to-pause) level. To mark speech chunks of interest, two boundaries need to be demarcated, and the corresponding text for each speech chunk should be entered between these boundaries. Chunks that are not relevant for analysis can be excluded by leaving the corresponding TextGrid intervals without any labels or text. All text inputs should be in Bangla orthography, using a Unicode font. The aligner has undergone testing with both Avro (OmicronLab,

n.d.) and Bijoy Unicode (Bijoy Ekushe, n.d.)) keyboards to ensure compatibility. It is necessary to make sure that the TextGrids are saved in Unicode text; in Praat, this can be done by checking the UTF-8 option the menu:: "Praat >> Preferences >> Text writing preferences..." >> UTF-8. If copying and pasting text from somewhere else, it is important to make sure there is no "newline" character at the end of any transcription interval. A sample chunk of transcription TextGrid, along with the waveform of the audio, is presented in Figure 1.

`

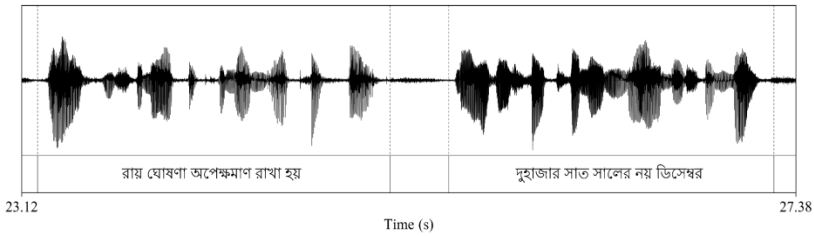

Figure 1: Examples of breath-group-level transcription chunks inside TextGrid

Running the aligner

Getting the setup ready

All the audio files and their corresponding transcription TextGrids (in Bangla orthography) need to be placed inside the folder "input_audio_and_transcription_tgs"; the names of this folder must not be changed since the scripts will expect a folder on this name. (Other folders not to touch include the “Bangla-Aligner” and “files_required_for_scripts” folders. Since Bangla-Align requires Python version 3 to run, it is time to check the python installation on the system. Here are the steps to follow:

- Open a new instance of the terminal application
- Navigate to the folder "bangla-align-linux*" (e.g., using ‘cd path_to_folder’ command)

    - For example, if we are currently in the home directory and we placed the alinger folder inside the “Documents” folder, running "cd Documents/bangla-align-linux-v1" should take us there
    - After this, the console prompt should show something like this: my_username@computer_name: ~/Documents/Bangla-Align-linu-v1"$
- Test the Python3 installation
    - Type “python3 --version” in the console and hit Enter
    - If python3 is properly installed, it should show the version number 3.5.0 or later; otherwise, check/(re)install python3 on your system before proceeding to the next steps.

Now we start the actual alignment works. The alignment is done in the major steps:

1. Generating a data-specific phoneme dictionary
2. Manually inspecting and correcting the dictionary file
3. Running the MFA and post-process the output data

Generating data-specific phoneme dictionary

The first step is to generate a phoneme dictionary tailored to the input data (particularly, for the words available in the transcription TextGrids). To do this, type the following line of command in the terminal and press Enter:

$ bash Script_1_generate_dictionary.sh

Running the command will create a phoneme dictionary named "temp_dict_bangla_ortho.dict" in the same directory (“Bangla-Align-linux-v1”). There is a high chance that the master

phoneme dictionary that comes with the aligner does not have entries for all the words in the data being aligned; in such case, running "Script_1…" will also generate a second dictionary file named "oov_with_phoneme_suggestions.txt".

Dealing with erroneous phoneme sequences

The next step is to manually inspect all the entries in the temporary dictionary and the OOV files for any erroneous entries and hand-correct them. Technically speaking, the aligner should be able to run without this step but the accuracy may suffer. Therefore, it is crucial to check each entry for errors and hand-correct them. While correcting them, the phone symbols must be chosen from the following list (because the aligner currently has acoustic models for these sounds only): AA, AAI, AAY, AE, AO, B, BS, CH, CHS, D, DD, DDS, DS, EA, EH, EHU, EHY, G, GS, HH, IH, JH, JHS, K, KS, L, M, N, NG, OI, OO, OU, P, PS, R, RH, S, SH, T, TS, TT, TTS, UH, W, Y, Z. Figure 2 provides some example entries in the phoneme dictionary.

```
33 অকার্যকর  AO K AA R JH OO K AO R
34 অকালমৃত্যু  AO K AA L M R IH TT TT UH
35 অকালে  AO K AA L EH
36 অকালেই  AO K AA L EHI
37 অক্টোবর  AO K T OO B AO R
```

Figure 2: Sample entries in phoneme dictionaries. Phones must be separated by single white spaces.

Generating final force-aligned TextGrids

At the final step, run the following command in the terminal:

$ bash Script_1_generate_dictionary.sh

The aligner will now run the Montreal Forced Aligner to generate the phone-level annotations followed by some post-processing. At this step, the MFA aligner uses the acoustic models stored in the file "bangla.zip" (inside - it will force-align the input files using the pre-trained acoustic models provided in bangla.zip (inside folder "Bangla-Aligner"). If successful, the aligner will now generate a new folder named "Z_final_aligned_TGs" which contains

all the final force-aligned TextGrids. Figure 3 shows an example chunk of a force-aligned TextGrid. The command will also save all the entries in the corrected temporary dictionary as well as the OOV file into the master phoneme dictionary so that the user does not have to correct the same errors multiple times.

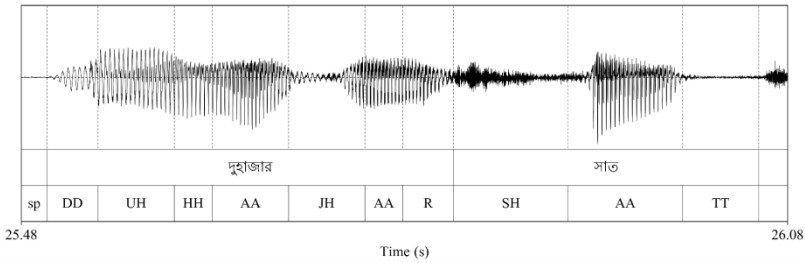

Figure 3: An example of output TextGrid tiers

Bangla-Align performance

The evaluation of accuracy for forced-aligners like Bangla-Align poses challenges as it involves assessing multiple levels of information, especially since there is no available gold standard database for comparison. Qualitatively speaking, the toolkit has shown reasonably good performance. Previous studies, including those by Author (2019) and Author (2020) have employed a previous version of the aligner, reporting reasonably acceptable results; in most cases, especially for vowels, the aligned TextGrids provided segment boundaries within 20 milliseconds of the expected positions.

In the absence of any previous gold standard database, a quantitative evaluation of the performance of Bangla-Align was performed with the data used in Author (2023). A total of 41,865 speech segments were force-aligned. Following the forced-alignment process, all the segments in each of the output TextGrids were manually checked to verify whether 1) the sounds were correctly labelled, and 2) the start and end timepoints for the sounds were accurate. Incorrect labels were manually corrected; also start and end

time points were manually adjusted if they were found to be misaligned. For the performance evaluation, the uncorrected force-aligned TextGrids were compared with the manually corrected TextGrids.

As the measures of performance, two metrics are presented here: 1) the percentage of tokens that were correctly labelled by aligner, and 2) percentage of tokens that required manual intervention in the form of time adjustment either in segment start time or end time. Table-2 and Table-3 provide the results. As Table-2 shows, 97.84% of the labels were correctly placed by the aligner and did not require manual intervention. Given the amount of time it requires to align speech manually; this level of accuracy is substantially high. Table-3 reports the cases that required manual adjustment of timepoints at segment boundaries (among the correctly labeled ones). As the results show, the percentage of tokens that required this adjustment was very low (up to 2%).

Table 2: Performance metric-1: Percentage of correctly labelled phones

Criteria	**Count (%)**
Total tokens measured	41865
Token count correctly labelled by aligner	40961 (**97.84%**)

Table 3: Performance metric-2: Percentage of tokens that needed manual time adjustments

Criteria	**Count (%)**
Token count where start time needed manual adjustment	447 (**1.09%**)
Token count where end time needed manual adjustment	414 (**1.01%**)

Token count where start time OR end time needed manual adjustment	742 (**1.80%**)

Implications

Within the framework of multilingual education, particularly in bilingual environments where native Bangla speakers engage in learning English, the development and use of a Bangla speech force-aligner toolkit can hold significant pedagogical implications. L1 transfer, a well-recognized factor influencing second language (L2) pronunciation acquisition, underscores the necessity of comprehending the phonetic nuances inherent in learners' first language (L1) (Hui, 2010). Proficiency in L1 phonetics becomes indispensable for accurately identifying and assessing the L1 characteristics being transferred during English language learning. This knowledge not only informs educators about potential challenges arising from L1 transfer but also becomes a foundation for crafting targeted teaching strategies that are specifically attuned to the multilingual education context. By leveraging the Bangla speech force-aligner toolkit, which offers a systematic and data-driven approach, educators can move beyond impressionistic observations. This shift facilitates a more evidence-driven exploration of L1 transfer in the Bangla-English education context, thereby contributing valuable insights to the field of TESOL.

Limitations and future work

The aligner has a number of limitations right now. First, the toolkit works only on Linux at the moment. The author has plans to expand it for Windows OS in the future. Second, even though the aligner has a decent number of trained phones in acoustic models, it lacks most of the Bangla diphthongs (which can often be tricky to distinguish from vowel hiatus in Bangla). As a workaround, the dictionary currently handles this by coding most diphthongs as a sequence of distinct vowels. This can, however, introduce new issues. Having the diphthongs split into two vowel phonemes makes it impossible to separate the real diphthings from vowel hiatus while

automating targeted vowel measurements, for example, using Praat (Boersma & Weenink, 2023) scripts. The steady part of the diphthong may fall into the first vowel while the glide part may be represented by the second vowel segment. Given that a researcher is really interested in studying Bangla diphthongs, there can be two ways to handle this. The first way is to take measurements from midpoints from both vowels and use them as the measurements for the steady and glide parts. Alternatively, they could curate the TextGrid itself to combine the two segments together and take necessary measurements. Future work on Bangla-Align will involve training new models for diphthongs. Finally, the master phoneme dictionary is very small; future work is planned to expand the dictionary (potentially by means of training Grapheme-to-Phoneme (G2P) models.

Conclusion

In conclusion, this research presents "Bangla-Align," a novel toolkit designed to address the pressing need for a force-aligner for Bangla speech. By streamlining the process of aligning speech data with their corresponding transcriptions, Bangla-Align offers linguistics researchers a valuable resource to explore and analyze the intricacies of spoken language. While the aligner showcases promising results and significantly reduces the manual effort required for alignment tasks, it also reveals some limitations that call for further development and refinement. Efforts are underway to expand platform compatibility, making the toolkit accessible to a wider range of users. The treatment of Bangla diphthongs, although handled with workarounds, presents a future avenue for research to enhance the alignment accuracy further. Additionally, the expansion of the master phoneme dictionary through Grapheme-to-Phoneme (G2P) models offers exciting prospects to enrich the toolkit's performance and versatility. As researchers continue to refine and expand "Bangla-Align," we anticipate that this toolkit will become an valuable tool for linguistic research, fostering new discoveries and insights into the complexities of the Bangla language.

References:

Bijoy Ekushe. (n.d.). Typing. Retrieved [date], from https://www.bijoyekushe.net/index.php?action=typing

Boersma, P., & Weenink, D. (2023). *Praat: Doing phonetics by computer* [Computer program].Version 6.3.09. Retrieved March 2, 2023, fromhttp://www.praat.org/

Ćavar, M., Ćavar, D., & Cruz, H. (2016). Endangered language documentation: Bootstrapping aChatino speech corpus, forced aligner, ASR. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (Vol. 10, pp. 4004-4011).

Gales, M., & Young, S. (2008). The application of hidden Markov models in speech recognition.*Foundations and Trends in Signal Processing*, *1*(3), 195–304.https://doi.org/10.1561/2000000004

Goldman, J. P. (2011). *EasyAlign: An automatic phonetic alignment tool under Praat*. In Twelfth Annual Conference of the International Speech Communication Association.

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

Hui, Y. (2010). The role of L1 transfer on L2 and pedagogical implications. *Canadian Social Science*, *6*(3), 97-103.

Islam, M. J. (2019). *Phonetics and Phonology of'Voiced-Aspirated'Stops: Evidence from Production, Perception, Alternation and Learnability* [Doctoral dissertation, Georgetown University].

Islam, M. J., & Ahmed, I. (2020). Mid-front and back vowel mergers in Mymensingh Bangla: An acoustic investigation. *Linguistics Journal*, *14*(1).

Islam, M. J., Al Masum, A. A. M. A., & Anwar, M. S. (2023). The Role of Temporal and Spectral Cues in Non-native Speech Production: Bangla Speakers' L2 English Tense and Lax Vowels. *Crossings: A Journal of English Studies*, *14*, 130-153.

Kelley, M. C., & Tucker, B. V. (2018). A comparison of input types to a deep neural network based forced aligner.

Ma, M., & Glass, L. (2022). *BedWord: A new automated pipeline for interview transcription for linguistic analysis and vowel extraction with DARLA*. Presented at New Ways of Analyzing Variation 50 (NWAV 50), October 15th, Stanford University.

Mahr, T. J., Berisha, V., Kawabata, K., Liss, J., & Hustad, K. C. (2021). *Performance of forced alignment algorithms on children's speech*. *Journal of Speech, Language, and Hearing Research*, *64*(6S), 2213-2222.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017). *Montreal Forced Aligner: Trainable text-speech alignment using Kaldi*. In Proceedings of the 18th Conference of the International Speech Communication Association.

Milne, P. M. (2011). Finding schwa: Comparing the results of an automatic aligner with human judgments when identifying schwa in a corpus of spoken French. *Canadian Acoustics*, *39*(3), 190-191.

Mor, B., Garhwal, S., & Kumar, A. (2021). *A systematic review of hidden Markov models and their applications*. *Archives of Computational Methods in Engineering*, *28*, 1429-1448.

OmicronLab. (n.d.). *Avro*. Retrieved [date], fromhttps://www.omicronlab.com/avrokeyboard.html

Punnoose, A. K. (2022). A study on forced alignment error patterns in Kaldi. In *2022 8th International Conference on Signal Processing and Communication (ICSC)* (Vol. 8, pp. 250-253). IEEE.

Reddy, S., & Stanford, J. (2015). A web application for automated dialect analysis. In *Proceedings of NAACL-HLT 2015*.

Reddy, S., & Stanford, J. N. (2015). *Toward completely automated vowel extraction: Introducing DARLA*. *Linguistics Vanguard*, *1*(1), 15-28.

Rosenfelder, I., Fruehwald, J., Evanini, K., Seyfarth, S., Gorman, K., Prichard, H., & Yuan, J. (2014). *FAVE (Forced Alignment and Vowel Extraction) Program Suite v1.2.2 10.5281/zenodo.22281*.

Wu, H., Yun, J., Li, X., Huang, H., & Liu, C. (2023). Using a forced aligner for prosody research. *Humanities and Social Sciences Communications*, *10*(1), 1-13.

Young, N. J., & McGarrah, M. (2023). *Forced alignment for Nordic languages: Rapidly constructing a high-quality prototype*. *Nordic Journal of Linguistics*, *46*(1), 105-131.

Yuan, J., & Liberman, M. (2008). *Speaker identification on the SCOTUS corpus*. *Journal of the Acoustical Society of America*, *123*(5).

Zhu, J., Zhang, C., & Jurgens, D. (2022). *Phone-to-audio alignment without text: A semi supervised approach*. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Zsiga, E. (2014). Sound recordings: Acoustic and articulatory data. In R. J. Podesva & D. Sharma (Eds.), *Research methods in linguistics*. Cambridge University Press.