A semi-automatic workflow for orthographic transcription of a novel speech corpus: A case study of AusKidTalk

Tünde Szalay[1,2], Louise Ratko[3], Mostafa Shahin[2], Tharmakulasingam Sirojan[2],
Kirrie Ballard[1], Felicity Cox[3], Beena Ahmed[2]

[1]The University of Sydney, [2]The University of New South Wales, [3] Macquarie University
tuende.szalay@sydney.edu.au

Abstract

Developing automatic speech recognition (ASR) tools for AusKidTalk, the novel Australian English (AusE) children's corpus, presents a circular problem: AusKidTalk is designed to develop adequate ASR for AusE-speaking children; however, orthographic transcription of AusKidTalk requires ASR tools not yet developed. Our semi-automatic workflow augments existing (but inadequate) automatic tools with manual transcription. IBM-Watson diarisation and UNSW ASR orthographic transcription automatically generate Praat textgrids with time-aligned orthographic transcriptions. A webtool distributes the textgrids, collects manual corrections, and implements consistency checks. Manual correction is conducted with a custom Praat interface. The output is a searchable, orthographically transcribed, and time-aligned corpus.

Index Terms: audio corpus, orthographic transcription, automatic speech recognition for novel populations

1. Introduction

AusKidTalk is an audio-visual corpus of Australian English (AusE) speaking children [1, 2]. Orthographic transcription and annotation of AusKidTalk are the essential first steps towards obtaining phoneme-level annotation [2, 3]. Due to its size, cost-efficient transcription and annotation is only possible with automatic speech recognition (ASR) tools.

Current ASR systems have been developed for adult speech and their performance drops considerably on children's data due to developmental differences [4, 5]. As ASR systems are trained on vast amounts of domain-specific annotated speech data [6], developing ASR for children has been thwarted by the lack of available children's corpora. Currently only 15 children's speech corpora are publicly available worldwide [7]. All were collected using problem-specific protocols with limited tasks, none is fully annotated, and only three of them are sufficiently sized for ASR development [7]. Developing new ASR tools for a novel large corpus presents a circular problem: one of the aims of AusKidTalk is to develop new and accurate ASR for AusE-speaking children [2]; but to efficiently annotate AusKidTalk, accurate ASR tools not yet developed are required.

To overcome the lack of suitable ASR tools, we developed a multi-step workflow combining existing, but suboptimal ASR tools designed for other populations, augmented with manual correction, to provide orthographic annotation for parts of AusKidTalk. Our aim was to balance the efficiency of existing ASR tools with the accuracy of manual annotation. This paper is a case study in corpus building, describing the challenges of orthographically transcribing AusKidTalk and presenting our solutions through a step-by-step guide of our workflow.

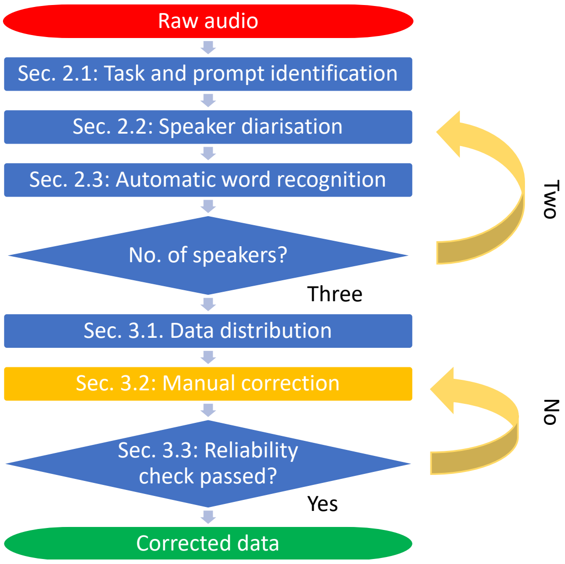
Raw audio
Sec. 2.1: Task and prompt identification
Sec. 2.2: Speaker diarisation
Sec. 2.3: Automatic word recognition
No. of speakers?
Two
Three
Sec. 3.1. Data distribution
Sec. 3.2: Manual correction
Sec. 3.3: Reliability check passed?
No
Yes
Corrected data

Figure 1: *Workflow outline. Red: start. Green: end. Blue: automated process. Yellow: manual process. Diamond: decision.*

1.1. The AusKidTalk corpus

To create the AusKidTalk corpus, we are collecting data from 750 native speakers of AusE, aged 3–12, with and without speech disorders and autism spectrum disorder, who contribute 90-120 minutes of audio. 475 children have participated; data collection is ongoing. Children complete five tasks with a range of linguistic complexity: three are prompted (word elicitation, pseudoword- and sentence repetition) and two are spontaneous (story telling and emotion elicitation); for details, see [2].

Tasks are presented via an Android app on a tablet while speech is recorded onto a PC. There is no direct synchronisation between the tasks, the prompts that appear on the tablet, and the recorded audio file. To align speech, the Android app plays a 1s high-frequency tone at the start of each task and records timestamps at the start and end of each task and prompt.

1.2. Challenges in annotating children's data

The audio recordings contain varied, spontaneous, and unexpected speech, inherent to children's data. There are unexpected responses to the prompts, such as responding to a picture of a cucumber with "zucchini", with a giggle to a picture of a belly-button, or only repeating parts of a sentence. In all tasks, there are non-task-related conversations between the child and the interviewer leading the recording session.

As the entire conversation was recorded due to the child's headset microphone picking up all speakers, the audio contains three speakers: the child, the pre-recorded model speaker who produced verbal prompts for the word and sentence level tasks, and the interviewer, instructing and aiding the child (e.g., "Can you speak up a bit?", "Very good!"). The combination of unexpected responses, spontaneous conversations, and three distinct speakers results in a high volume of non-target speech which further increases the difficulty of automated annotation.

1.3. Scope and goal of the annotation workflow

The goal of the workflow protocol described here (Fig. 1) is to annotate the prompted picture naming task (Task 1) for each child by orthographically transcribing all 130 target words and locating their start and end times in the audio file. Our output is a time-aligned Praat textgrid for each child that contains intervals for the target words only and an easily searchable csv file listing the target words with their start and end times.

2. Automated tools

2.1. Time-aligning with tone detection

To determine the start and end of the word elicitation task in the audio file, we developed automated tone detection using a nonlinear binary Support Vector Machine (SVM) with the radial basis function kernel to identify the location of the 1s tone and match it with the timestamps. To train the SVM, feature vectors were extracted from 3700 not-tone and 3700 tone frames selected randomly from 10 recordings and spliced with the feature vector of two preceding and two succeeding frames. On a test set of 10 recordings, our classifier achieved 0% False Acceptance Rate by never identifying not-tone as tone and approximately 10% False Rejection Rate by missing 4 out of 45 tones.

The SVM classifies each 10ms frame of the recording as tone or not-tone. The moving average of the number of detected tones is calculated using a one-second sliding window. Peak points with a moving average above 0.9, i.e., with at least 90% of frames classified as tone within a 1s window, are considered to be tone positions. The duration between every two tone sounds is calculated and compared to the duration between every two timestamps marking the start and the end of a task. Reference tone-timestamp pairs are identified when the duration between any two tone sounds is equal to the duration between any two timestamps. Task 1 is separated in the audio file using reference tone-timestamps.

A Praat textgrid is generated with intervals indicating the start and end of each prompt using the prompt timestamps. The prompt interval is the time between the presentation of the prompt picture to the child, and the pressing of the assessment button by the interviewer which indicates that the child completed the attempt of the current prompt.

2.2. Diarisation with IBM-Watson

To separate the child's voice from the interviewer and the model speaker, the IBM-Watson speech-to-text web service is used to map the three speakers onto three different tiers on a Praat textgrid. IBM-Watson uses deep learning AI with language specific models of grammar, vocabulary and acoustics to diarise and transcribe speech. The "AU-Narrow Band" model has been tested, as it is suitable for audio sampling rates above 8 kHz and can diarise and transcribe the speech of up to three speakers. The "AU- Narrow Band" model will be phased out in 2023, and we will replace it with the "AU Multimedia model"; the current paper reports data using the former.

Task 1 audio files are resampled from 44.1 kHz to 12 kHz to reduce uploading time and processed with IBM-Watson. The resulting JSON files contain utterance labels, confidence scores, and start and end times. Transcription and diarisation information is converted into Praat textgrids, with one speaker per tier.

IBM-Watson may diarise the audio correctly as containing three speakers or incorrectly as containing two speakers. To ensure that the child's speech is identified, tiers are counted automatically. Textgrids containing fewer than three tiers per recording are visually inspected by a trained phonetician to identify which tier does not contain the child's speech (top yellow arrow in Fig. 1). When the child's tier is separate from the other two speakers, the audio and the textgrid is passed onto the UNSW ASR tool (Sec. 2.3). When the child's speech is not separate from either the interviewer or the model speaker, speech on the non-child tier is silenced using a Praat script that identifies the intervals on the non-child tier and reduces their amplitude to zero. IBM-Watson is redeployed to differentiate the child's speech from the remaining speaker. A second JSON file is returned and converted into a Praat textgrid, with two tiers, one for the child, and one for the other speaker.

In a test sample of five randomly selected recordings, IBM-Watson correctly identified the three different voices in the file in two recordings. The remaining three were diarised as two speakers; having silenced the non-child audio and redeploying IBM-Watson resulted in a total of three speakers. That is, IBM-Watson successfully identified three distinct speakers in five out of five recordings. Diarisation and transcription accuracy of the tier identified as child only was evaluated in test session recordings from four children (age range = 4 – 10 years, mean = 6.75). 82%–95% (mean = 89%) of all intervals mapped onto the child's tier contained the child's speech. 85%–91% (mean = 87.93%) of all targets were identified on the child's tier, while the remaining targets were mapped onto another speaker's tier. That is, the child's speech was separated from the other speakers' with high accuracy. Orthographic transcription accuracy however was so low as to being practically unusable with a word error rate ranging from 94% to 57% per child (mean = 78.23). Therefore, we only used IBM-Watson for diarisation.

2.3. Automatic word recognition with UNSW ASR system

Word recognition is conducted using the UNSW ASR engine [10]. The UNSW ASR engine's acoustic model is based on deep-learning and trained on $\sim$400 hours of children's speech from four different American speech corpora using the Kaldi toolkit [9, 10]. The UNSW ASR engine uses a large-vocabulary language model trained on transcriptions of adult and child speech to cope with the spontaneous conversations occurring in the recording, and prioritises the 130 target words of Task 1.

Audio files with IBM-Watson textgrids containing diarised and time-aligned transcription are fed to the UNSW ASR. The UNSW ASR returns word-level transcriptions for all intervals identified by IBM-Watson, replacing IBM-Watson's transcriptions with its own. Word-level boundaries are compared to the prompt interval (Sec. 2.1). Prompt intervals that do not overlap with any word intervals are processed by the UNSW ASR tool and the intervals of the recognised words are added to all speaker tiers. On the test set of the same four children, UNSW ASR achieved a word error rate of 18%-45% (mean = 30%), less than half of the word error rate of IBM-Watson.

To minimise manual annotation time, IBM-Watson is used to isolate child only segments and the UNSW ASR tool to orthographically transcribe audio. The IBM-Watson web service and the UNSW ASR are linked with a Python script taking several sound files and returning diarised and transcribed textgrids.

3. Manual correction

Manual correction is required to improve the accuracy of the transcription and placement of target words on the automatically generated textgrids as well as to remove annotation for non-target words. Given the large size of the collected corpus, a team of annotators based at multiple sites is involved in the manual correction process.

3.1. Distributing data across multiple sites

A web tool was developed to distribute the audio recordings and the automatically generated textgrids across the annotation team, collect the manually corrected textgrids, and ensure consistency across annotators. The front-end of the web tool provides two interfaces: one for annotators and one for their supervisors. Users are directed to their corresponding interface based on their roles and authentication credentials.

The annotator interface provides options to download a new, automatically allocated audio file with the associated prompt and uncorrected speaker textgrids, as well as to upload the corrected annotation files. To test correction consistency, benchmark files are inserted at regular intervals. Benchmark files are files with ground truth annotations (i.e. previously corrected by expert phoneticians; Sec. 3.3). Annotators are blinded as to which files are benchmark files. When corrections to benchmark files are uploaded, the corrections are automatically scored against ground truth annotation (Sec. 3.3). The annotator cannot proceed to the next file unless a passing score is achieved. Feedback and additional training will be provided.

The backend of the web tool contains a database that keeps track of file allocation statistics, annotator scores, and account information. The backend logic is implemented in PHP scripting language with the MySQL database while the front end is developed with HTML5 and JavaScript. The web tool is hosted in a university web server which provides accessibility to the annotators from different locations.

3.2. Praat interface

A Praat [11] tool was designed to enable efficient manual correction of word-level annotation. The frontend streamlines the correction procedure with a user-friendly interface. The backend automates low-level and time-consuming tasks (e.g., opening and saving textgrids) and loads those portions of the sound file that are likely to contain a target word and skips the rest.

3.2.1. Steps of manual correction

Tasks that could not be fully automated are streamlined by creating a series of five tasks with simple instructions presented in Praat pop-up windows. As the IBM-Watson diarisation tool only identifies the speech segments belonging to the different speakers (interviewer, model speaker, child) and not who the actual speakers are (Sec. 2.2), the first manual task is identifying which tier belongs to the child. When the annotator starts a new file, the waveform and the spectrogram with a textgrid showing the prompts on one tier and the three speakers on three separate tiers are displayed. The annotator is asked to identify which tier is the best match for the child's speech. The annotator is instructed to scroll through the recording and to listen to as much of the audio as needed before making their decision.

After selecting the child's tier, the script proceeds to the correction phase of the selected tier, while the other two speaker tiers are discarded. The prompt tier is displayed together with the child's tier. Correction of an interval contains four steps: interval evaluation, label and boundary evaluation, noise evaluation, and phoneme-level discrepancy evaluation. When an interval is presented, the annotator has the options to Accept, Delete (not child), or Delete (not target) the interval. The annotator is instructed to only accept an interval if the interval contains the child's voice and the child's speech matches the prompt.

Once the annotator accepts an interval, the annotator is instructed to either accept or edit the interval's label (i.e., the automatic transcription) and/or its boundaries. Annotators are trained to transcribe the child's speech using standard English spelling and grammar, e.g., spell the target word *rhinoceros* with the letter "r", regardless of whether it was produced with the rhotic or an approximant, such as [w] or [ʋ]; and *two eggs* with the plural marker -s, even if the child produced *two egg*. Annotators are trained to ensure that the boundaries contain the entire word produced by the child and no other speech. In particular, they are warned to be careful to include final stop bursts which may be cut off by automatic annotation tools due to children producing a longer closure phase in stops than adults [12].

Once all the necessary corrections to the intervals are complete, the annotator is asked to decide whether the recording contains any noise (e.g., overlap between the interviewer and the child, background noise) and whether the token contains any phoneme-level insertions (e.g., *skirts* for *skirt*), deletions (e.g., *four egg* instead of *four eggs*), or substitutions (e.g., *tooth* produced with a final /f/ instead of /θ/). Annotators are told to flag any phoneme-level discrepancy from adult production irrespective of it being age-appropriate (e.g., [w] for /ɹ/ flagged at each instance, regardless of age). When a word is flagged as non-adult like, annotators must identify the differing phoneme(s). To prevent the annotators from spending too much time on noise and discrepancy evaluations, the script only allows the target word to be played twice for noise and twice for discrepancy evaluation (four times in total) and automatically closes the sound and the textgrid while waiting for the noise and discrepancy decisions. The annotator is able to move to the next interval once all four steps for a given interval are completed.

To improve annotation quality, the script checks that the annotator follows the instructions as closely as possible. A notification is triggered when the interval label does not match the prompt and/or when the annotator's evaluation does not match what they have done (e.g., the annotator has evaluated the automatically placed interval label as correct yet they have changed it). The annotator is instructed to correct their error(s). Erroneous edits are not saved and the annotator is not allowed to move onto the next interval until all checks are passed. When all checks are passed, final edits and progress are saved and the interval is marked as corrected, allowing the annotator to exit Praat any time. At the next launch, the script loads an unfinished sound file - textgrid pair at the first uncorrected interval.

Once the annotator corrects the textgrid for the entire sound file, a clean and corrected textgrid free from unnecessary annotations is created by removing all remaining child labels in which the label does not match the prompt. To create an easily searchable output, a csv file is generated that lists the label, and the start and end time of every on-prompt target and is saved automatically with the child's identifier. The clean and corrected textgrid, together with the csv file are uploaded to the web tool.

3.2.2. Focus on intervals of interest

The Praat tool automatically identifies those intervals that are the most likely to contain target words by comparing the prompt tier to the automatic transcription of the child's speech in an iterative process. Comparison of the child's production and the prompt is required as target words produced without being prompted are excluded from the data. For instance, frequent words (e.g., *yes, no, that*) produced without a prompt, or easily confusable words produced for the incorrect prompt (e.g., the target *boat* produced for the picture of a canoe) are excluded.

In the first iteration, the script loads the intervals in which the automatic transcription (Sec. 2.3) matches the prompt, as these are likely to contain a target word. For instance, the interval labelled with the target word *key* is loaded only if the prompt interval is also labelled with *key*. When more than one interval labels match the prompt, all are loaded and corrected. The tool tracks which prompts are found and corrected in the first iteration. All other intervals, i.e., intervals transcribed as non-target words or as target words not matching the prompt are skipped.

In the second iteration, the script loads all the word-level intervals that map onto prompts not found in the first iteration. For instance, if the target *key* was found in the first iteration, then all other word-level intervals that map onto the prompt *key* are assumed to be non-target and skipped. If, however, the target *key* was not found in the first iteration, all intervals identified as the child's speech produced during the prompt *key* are displayed, irrespective of their transcription. If *key* was produced twice, and transcribed as *he* and *e* respectively, both are loaded and corrected in the second iteration.

In the third iteration, the script loads an entire prompt interval if the prompt was not found in either of the previous iterations. The annotator is asked to listen to the entire interval during which a prompt was displayed to identify and add the target word. More than one target interval can be added. We assume that target words that are not found in any of the iterations were not produced or were only produced when not prompted.

Repeated targets are identified when they are corrected in the same iteration. Repetitions are skipped when they would be corrected in different iterations, e.g., if one instance of *key* has a correct automatic transcription and the other does not, the one with the correct automatic transcription is identified in the first iteration, and the other is skipped in the second iteration.

The length of audio the annotator must listen to is reduced. For instance, a prompt might be displayed for 6-10 seconds; however, the word-level interval(s) might be only 0.6-1 second long. The annotator only listens to the automatically generated, shorter word-level interval(s). In a sample of four children, the length of audio the annotator listened to was reduced from a total of 107 minutes to 22 minutes. We identified 125–127 targets out of a total of 130 (mean = 126). 75–99 targets (mean = 85.75) were found in the first iteration, 15–30 (mean = 23) in the second, and 12–23 (mean = 17.25) in the third.

3.3. Reliability checks

Interrater reliability is typically done by all annotators correcting the same set of files (customarily 20% of the data). Due to the large number of files (approximately 750 sound files for 750 children), a 20% cross-correction is not possible as it would have required 140 speakers being marked by all annotators. Similarly, 20% rescoring for intrarater reliability is not feasible due to the large number of files.

Therefore, a ground-truth approach is chosen to achieve consistency. Eight benchmark sound files are identified consisting of four older (10-12 years, M = 2, F = 2) and four younger children (3-5 years, M = 2, F = 2). Four of the children are typically developing, and four have current speaking disorders (one older male, one younger male, one older female, one younger female). Four expert phoneticians manually correct the benchmark files independently from each other and compare their correction. In case of a disagreement, consensus is reached through discussion; disagreement typically arises regarding flagging non-adult like productions and almost never regarding identifying target words.

In every benchmark file, the annotator's work is compared to the ground truth by comparing the number of targets identified and calculating overlap rate for matching targets between the benchmark and the current annotation. The number of identified targets tests whether the annotator found all the target words in the audio data. Repetitions of targets are not counted towards the pass rate, as the task was not designed to capture repetitions for targets and the Praat interface does not require identifying all repetitions. Overlap rate is calculated relative to the ground truth annotation, using the time shared between ground truth and current annotation (*Dur Shared*), the duration of the ground truth annotation (*Dur GT*), and the duration of the current annotation (*Dur Current*) (Equation 1), [13, 14].

$$Overlap = \frac{DurShared}{DurGT + DurCurrent - DurShared} \quad (1)$$

Overlap rate ranges from 1 (complete agreement, 100% overlap) to 0 (no agreement, 0% overlap) and penalises too long and too short intervals equally. If the ground truth annotation for a target is 0.6s long, a current annotation with 1.2s duration and 0.6s shared duration and a current annotation with 0.3s duration and 0.3 shared duration both yield an overlap rate of 0.5.

Passing rate for annotators is calculated automatically from the number of target words identified and from the overlap rate (Sec. 3.1). If a passing rate is not achieved, the corrected files after the last successful checkpoint (if any) will be reviewed (Bottom yellow arrow in Fig. 1) and re-corrected if needed.

4. Conclusions and future work

Our goal was to provide time-aligned orthographic transcription to the prompted single word elicitation task (Task 1) in the AusKidTalk corpus. We overcame the lack of suitable ASR tools required for the task by developing a semi-automated workflow that concatenates IBM-Watson diarisation with the task-specific UNSW ASR orthographic transcription system to automatically generate textgrids with time-aligned transcription. A webtool distributes the automatically generated textgrids and collects manually corrected textgrids, and implements consistency checks against ground truth annotations. Correction is done in a custom Praat interface.

This workflow is essential to achieve an orthographic annotation of target items in an efficient manner and creates a corpus to be further processed using forced alignment to generate phonemic annotations. However, a lot of valuable data are necessarily disregarded, such as incidental conversations between the interviewer and the child. Therefore, raw data files will be made available as a corpus for researchers who are interested in more than just the target items. We aim to extend the workflow to include non-word repetition by adding the pseudowords to the UNSW ASR system's dictionary, and use the workflow as a starting point for annotating the sentence repetition task using a modified Praat interface.

5. Acknowledgements

This project was supported by the Australian Research Council LE190100187 and FT180100462 grants, as well as the University of New South Wales, The University of Sydney, Western Sydney University, Macquarie University and The University of Melbourne. We would like to thank our participants without whom this project would not have been possible.

6. References

[1] http://www.auskidtalk.edu.au/

[2] Ahmed, B., Ballard, K. J., Burnham, D., Tharmakulasingam, S., Mehmood, H., Estival, D., Baker, E., Cox, F., Arciuli, J., Benders, T., Demuth, K., Kelly, B., Diskin-Holdaway, C., Shahin, M., Sethu, V., Epps, J., Lee, C. B. and Ambikairajah, E., “AusKidTalk: An Auditory-Visual Corpus of 3-to 12-year-old Australian Children’s Speech”, ISCA, 2021.

[3] Schiel, F., Draxler, C., Baumann, A., Ellbogen, T. and Steffen A. “The production of speech corpora” Version 2.5, 2012, http://www.bas.uni-muenchen.de/Forschung/BITS/TP1/Cookbook

[4] Russell, M. and D’Arcy, S. “Challenges for computer recognition of children’s speech” , Workshop on Speech and Language Technology in Education, 2007.

[5] Elenius, D. and Blomberg, M. “Adaptation and normalization experiments in speech recognition for 4 to 8 year old children”, Interspeech, 2749–2752, 2005.

[6] Keshet, J., “Automatic speech recognition: A primer for speech-language pathology researchers”, International Journal of Speech-Language Pathology, 20(6):599–609, 2018.

[7] Chen, N. F., Tong, R., Wee, D., Lee, P. X., Ma, B. and Li, H., “SingaKids-Mandarin: Speech Corpus of Singaporean Children Speaking Mandarin Chinese”, Interspeech, 1545-1549, 2016.

[8] Transcribing speech with Watson Speech to Text. (2021, 2022). IBM Corporation. https://www.ibm.com/docs/en/cloud-paks/cp-data/4.0?topic=solutions-watson-speech-text Accessed: 5 September 2021

[9] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G. and Veselý, K. “The Kaldi speech recognition toolkit”, IEEE workshop on automatic speech recognition and understanding, 2011.

[10] Shahin, M. A., Lu, R., Epps, J. and Ahmed, B. “UNSW System Description for the Shared Task on Automatic Speech Recognition for Non-Native Children’s Speech”, Interspeech, 265-268, 2020.

[11] Boersma, P. and Weenink, D. (2022). Praat: doing phonetics by computer [Computer program]. Version 6.2.14, retrieved 24 May 2022 from http://www.praat.org/

[12] Millasseau, J., Ivan, Y., Bruggeman, L. and Demuth, K., “Acoustic cues to coda stop voicing contrasts in Australian English-speaking children”, Journal of Child Language, 48(6): 1262-1280, 2021.

[13] González, S., Grama, J. and Travis, C., “Comparing the accuracy of forced-aligners for sociolinguistic research”, Linguistics Vanguard, 6(1).

[14] Paulo, S. and Oliveira, L. C., “Automatic phonetic alignment and its confidence measures”, Proc. Advances in Natural Language Processing, 36–44, 2004.